A
Project Report
on

A VERIFIABLE SEMANTIC SEARCHING SCHEME BY OPTIMAL MATCHING OVER ENCRYPTED DATA IN PUBLIC CLOUD

Submitted in partial fulfillment of the requirements
for the award of degree
of

Bachelor of Technology

by

K JASHWANTH
ROY (20EG105624)

M VAMSHI KRISHNA
(20EG105629)

M SUSHMITHA
(20EG105723)

Under the Guidance of
Dr. C. Dastagiraiah
Assistant Professor
Department of CSE

Department of Computer Science & Engineering
Anurag university
Venkatapur (V), Ghatkesar (M), Medchal (D), T.S-500088
(2023-2024)

DECLARATION

We, K JASHWANTH ROY (20EG105624), M VAMSHI KRISHNA (20EG105629), and M SUSHMITHA (20EG105723) here declare that the report entitled **“A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud”** submitted for the award of **Bachelor of Technology in Computer Science and Engineering** is our original work and report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma, to the best of our knowledge and faith.

Place: Anurag University, Hyderabad

K JASHWANTH ROY
(20EG105624)

M VAMSHI KRISHNA
(20EG105629)

M SUSHMITHA
(20EG105723)

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the report entitled **“A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud”** that is being submitted by K JASHWANTH ROY (20EG105624), M VAMSHI KRISHNA (20EG105629), M SUSHMITHA (20EG105723) in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering to the Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this report have not been submitted to any other university or Institute for the award of any degree or diploma.

Signature of the Supervisor **Dean, CSE**

Dr. C. Dastagiraiah
Assistant Professor
Department of CSE

External Examiner

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this major project work.

First, we take this opportunity to express our sincere gratitude to the **Faculty of Computer Science and Engineering** for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.

We want to thank our guide **Dr. C. Dastagiraiah, Dept. of Computer Science and Engineering** for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to thank **Dr. V. Vijaya Kumar, Dean School of Computer Science & Engineering** Anurag University**,** for their constant encouragement and timely support in our B. Tech Program.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy, Dean,** Dept. of. CSE Anurag University. We also extend our deep sense of gratitude to **Dr. VV SS Balaram**, Academic coordinator. **Dr. Pallam Ravi** project coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of our project work.

K JASHWANTH ROY
(20EG105624)

M VAMSHI KRISHNA
(20EG105629)

M SUSHMITHA
(20EG105723)

ABSTRACT

Semantic searching over encrypted data is a critical endeavor for ensuring secure information retrieval in the public cloud, offering flexibility in querying and search result generation. Current semantic searching schemes lack verifiability, as they depend on forecasted results from predefined keywords for verifying search outcomes, expanding queries on plaintext, and performing exact matching with semantically extended words and predefined keywords, thereby limiting accuracy. To address these challenges, this paper proposes a novel secure verifiable semantic searching scheme. It introduces a formulation of the Word Transportation (WT) problem to achieve semantic optimal matching on ciphertext, aiming to calculate the Minimum Word Transportation Cost (MWTC) as a measure of similarity between queries and documents. Additionally, a secure transformation method is proposed to convert WT problems into random Linear Programming (LP) problems, enabling the computation of encrypted MWTC. For ensuring verifiability, the duality theorem of LP is explored to design a verification mechanism leveraging intermediate data generated during the matching process, thereby verifying the correctness of search results. A comprehensive security analysis is conducted, demonstrating that the proposed scheme can guarantee both verifiability and confidentiality. Experimental evaluations are conducted on two datasets, showcasing the superior accuracy of the proposed scheme compared to existing approaches. By enabling flexible retrieval services for arbitrary words and addressing the limitations of existing schemes, the proposed semantic searching framework offers a robust solution for secure information retrieval in the cloud environment. Additionally, its ability to ensure verifiability enhances the trustworthiness of search outcomes, making it suitable for a wide range of applications where data security and accuracy are paramount concerns.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The project focuses on addressing the critical need for secure information retrieval in the public cloud through semantic searching over encrypted data. Existing semantic searching schemes lack verifiability, relying on predefined keywords and plaintext expansion for search accuracy. To overcome this limitation, a novel secure verifiable semantic searching scheme is proposed. The approach involves formulating a Word Transportation (WT) problem to calculate the Minimum Word Transportation Cost (MWTC) as a measure of similarity between queries and documents on ciphertext. A secure transformation method is introduced to convert WT problems into random Linear Programming (LP) problems, ensuring encryption of MWTC. Verifiability is achieved by leveraging the LP duality theorem to design a mechanism that utilizes intermediate data generated during the matching process to verify search results' correctness. Security analysis confirms the scheme's ability to guarantee both verifiability and confidentiality. Experimental validation on two datasets demonstrates the scheme's superior accuracy compared to existing approaches. Overall, the proposed scheme offers a flexible and secure solution for semantic searching over encrypted data, enhancing information retrieval capabilities in the public cloud while ensuring data privacy and integrity.

1.2 PURPOSE OF THE PROJECT

The purpose of the project is to address the critical need for secure information retrieval in public cloud environments through semantic searching over encrypted data. Current semantic searching schemes lack verifiable searching capabilities, relying on predefined keywords for result verification and performing exact matching on plaintext, leading to limited accuracy. To overcome these limitations, the proposed secure verifiable semantic searching scheme aims to provide retrieval services for arbitrary words, ensuring flexible queries and search results. By formulating the Word Transportation (WT) problem and calculating the Minimum Word Transportation Cost (MWTC) as the similarity metric between queries and documents on ciphertext, the scheme enhances semantic optimal matching. This is achieved through a secure transformation process that converts WT problems into random Linear Programming (LP) problems, thereby obtaining encrypted MWTC. Additionally, to ensure verifiability, the project leverages the duality theorem of LP to design a verification mechanism using intermediate data from the matching process. Through rigorous security analysis, the scheme guarantees both verifiability and confidentiality. Experimental evaluations conducted on two datasets demonstrate that the proposed scheme outperforms existing methods in terms of accuracy, thus offering a robust solution for secure semantic searching over encrypted data in public cloud environments.

1.3 MOTIVATION

The motivation behind this project stems from the pressing need for secure information retrieval in public cloud environments. Semantic searching over encrypted data is pivotal for maintaining confidentiality while still enabling flexible query and retrieval capabilities. Existing semantic searching schemes fall short in supporting verifiable searching due to their reliance on predefined keywords for result verification, leading to limited accuracy. By addressing this gap, our proposed secure verifiable semantic searching scheme aims to revolutionize information retrieval in the cloud. Through the formulation of a word transportation problem and its transformation into random Linear Programming (LP) problems, we achieve semantic optimal matching on ciphertext, enhancing search accuracy while maintaining confidentiality. Additionally, leveraging the LP duality theorem, we design a verification mechanism to ensure the correctness of search results. The security analysis affirms the verifiability and confidentiality of our scheme, while experimental results on two datasets demonstrate its superior accuracy compared to existing solutions, underlining the significance and potential impact of this project.

CHAPTER 2
LITERATURE SURVEY

A thorough literature review was conducted to understand existing solutions. This section compares various approaches, highlighting their strengths and weaknesses, and providing insights into the methods employed in similar applications.

2.1 COMPARISON LITERATURE

Over the past two decades, searchable encryption has garnered significant attention due to its practicality, facilitating secure information retrieval over encrypted cloud data. Numerous works have concentrated on enhancing both the security and functionality of searchable encryption schemes. Regarding security, scholars have formulated various definitions and attack patterns to evaluate the robustness of existing schemes. Goh et al. introduced a security model termed semantic security against adaptive Chosen Keyword Attack (IND-CKA), ensuring document indexes conceal document contents. Curtmola et al. extended security definitions, including chosen-keyword attacks and adaptive chosen-keyword attacks. Additionally, privacy concerns led to the introduction of access pattern disclosure and novel attacks, such as search pattern leakage.

In parallel, research efforts have focused on enhancing functionality to meet practical demands. This includes ranked search capabilities, where cloud servers evaluate relevance scores between queries and documents without disclosing sensitive information. Cao et al. proposed a privacy-preserving Multi Keyword Ranked Search Scheme (MRSS), utilizing binary vectors and secure kNN algorithms. Furthermore, advancements in homomorphic encryption facilitated multi keyword ranked search schemes, enabling relevance score encryption and calculation under the vector space model.

Furthermore, the incorporation of semantic information has been pivotal in enhancing search accuracy. Traditional schemes often fail to exploit semantic relationships between words, limiting evaluation of query-document relevance. Fu et al. pioneered synonym searchable encryption, extending keyword sets using synonym thesauri and integrating secure indexes. Xia et al. introduced semantic extension searching schemes based on concept hierarchies, improving accuracy by weighting query words based on grammatical relations and extending central words using hierarchy trees.

Finally, the quest for verifiable searching over encrypted data has led to innovative approaches ensuring the correctness of search results. Some schemes verify the presence of encrypted documents containing specific query words, while others focus on verifying ranked search results. Wang et al. proposed a single keyword ranked verification scheme based on hash chains, while Sun et al. introduced a multi-keyword ranked verifiable searching scheme using Merkle Hash trees and cryptographic signatures. However, challenges remain in supporting semantic searching and minimizing communication overhead, especially in multi-data owner scenarios.

2.2 EXISTING SYSTEM

Most of the existing secure semantic searching schemes consider the semantic relationship among words to perform query expansion on the plaintext, then still use the query words and extended semantically related words to perform exact matching with the specific keywords in outsourced documents. We can roughly divide these schemes into three categories: secure semantic searching based synonym secure semantic searching based mutual information model secure semantic searching based concept hierarchy. We can see that these schemes only use the elementary semantic information among words. For example, synonym schemes only use synonym attributes; mutual information models only use the co-occurrences information. Although Liu et al. introduce the Word2vec technique to utilize the semantic information of word embeddings, their approach damages the semantic information due to straightly aggregating all the word vectors. We think that secure semantic searching schemes should further utilize a wealth of semantic information among words and perform optimal matching on the ciphertext for high search accuracy.

2.3 DISADVANTAGES OF THE EXISTING SYSTEM:

The existing system faces several disadvantages:

- Limited Verifiability: Many current schemes lack robust mechanisms for verifying the correctness of search results. While some schemes focus on verifying the presence of queried keywords in returned documents, they fail to ensure the accuracy of the entire search result set. This limitation undermines the trustworthiness of the search outcomes, particularly in scenarios where data integrity is critical.

- Semantic Information Neglect: Traditional searchable encryption schemes often overlook semantic information, relying solely on keyword matching. This omission hampers the accuracy and relevance of search results, especially when users input queries with synonymous or semantically related terms. Consequently, the system may retrieve irrelevant or incomplete information, reducing user satisfaction and efficiency.
- Complexity and Overhead: Certain schemes introduce significant computational overhead and communication complexity, particularly in multi-owner scenarios. Verifiable searching mechanisms may necessitate multiple rounds of communication between data owners, leading to increased latency and resource consumption. This complexity poses practical challenges in real-world deployments, especially in large-scale cloud environments where efficiency is paramount.
- Limited Support for Dynamic Operations: Some schemes struggle to support dynamic operations such as data addition and deletion while maintaining security and verifiability. As cloud data continually evolves, the inability to accommodate changes dynamically restricts the scalability and flexibility of the system, limiting its applicability in dynamic environments.
- Vulnerabilities to Attacks: Existing schemes may be susceptible to various attacks, including access pattern disclosure, search pattern leakage, and chosen-keyword attacks. These vulnerabilities undermine the confidentiality and integrity of encrypted data, jeopardizing the overall security of the system. Addressing these vulnerabilities is crucial to ensuring the robustness and resilience of searchable encryption solutions in the face of evolving threats.

Addressing these disadvantages is essential for advancing the state-of-the-art in searchable encryption and enabling secure, efficient, and reliable information retrieval in cloud environments.

CHAPTER 3
PROPOSED METHODS

3.1 PROPOSED SYSTEM

The proposed system aims to address the limitations of existing semantic searching schemes by introducing a secure verifiable semantic searching approach. Firstly, the system formulates a Word Transportation (WT) problem to calculate the Minimum Word Transportation Cost (MWTC) as a measure of similarity between queries and documents. In this paper, we propose a secure verifiable semantic searching scheme that treats matching between queries and documents as an optimal matching task. We treat the document words as “suppliers,” the query words as “consumers,” and the semantic information as “product,” and design the Minimum Word Transportation Cost (MWTC) as the similarity metric between queries and documents. Therefore, we introduce word embeddings to represent words and compute Euclidean distance as the similarity distance between words, then formulate the Word Transportation (WT) problems based on the word embeddings representation. However, the cloud server could learn sensitive information in the WT problems, such as the similarity between words. For semantic optimal matching on the ciphertext, we further propose a secure transformation to transform WT problems into random Linear Programming (LP) problems. In this way, the cloud can leverage any readymade optimizer to solve the RLP problems and obtain the encrypted MWTC as measurements without learning sensitive information. Considering the cloud server may be dishonest to return wrong/forged search results, we explore the duality theorem of Linear Programming (LP) and derive a set of necessary and sufficient conditions that the intermediate data produced in the matching process must satisfy. Thus, we can verify whether the cloud solves correctly RLP problems and further confirm the correctness of search results.

For example, consider a query "machine learning" and a document containing the phrase "artificial intelligence and machine learning." The system calculates the MWTC by determining the minimum cost to transform the words in the query to those in the document, considering semantic relationships and context.

Next, the system transforms the WT problems into random Linear Programming (LP) problems to obtain encrypted MWTC. This ensures that the matching process is performed securely over encrypted data, preserving confidentiality. To achieve verifiability, the system leverages the duality theorem of LP to design a verification mechanism. This mechanism uses intermediate data produced during the matching process to verify the correctness of search results. For instance, if the search result claims a high similarity score between the query and a document, the verification mechanism checks the consistency of this score with the encrypted MWTC obtained during the matching process. By combining semantic matching, encryption, and verification mechanisms, the proposed system ensures both accuracy and security in semantic searching over encrypted data.

3.2 ADVANTAGES OF PROPOSED SYSTEM

The proposed system offers several advantages over existing solutions in the field of searchable encryption and semantic searching:

- Verifiability: One of the key advantages is the integration of a verification mechanism, ensuring the correctness of search results. By leveraging the LP duality theorem and intermediate data from the matching process, users can verify the integrity of the retrieved documents without compromising security.
- Enhanced Security: The scheme guarantees both verifiability and confidentiality. Security analysis demonstrates the robustness of the proposed system against various attacks, thereby ensuring the privacy of the stored data and search queries.
- Semantic Searching: Unlike traditional searchable encryption schemes, which often fail to utilize semantic information effectively, the proposed system incorporates semantic searching capabilities. By formulating word transportation problems and leveraging random LP transformations, it can evaluate the relevance between queries and documents more accurately, improving search accuracy.
- Flexibility: The system aims to provide retrieval services for arbitrary words, allowing for flexible queries and search results. This flexibility is crucial in real-world applications where users may need to search for a wide range of terms and concepts.

- Optimal Matching: Through the formulation of the word transportation problem, the system calculates the Minimum Word Transportation Cost (MWTC) as the similarity measure between queries and documents. This approach ensures optimal matching on ciphertext, leading to more precise search results.
- Experimental Validation: The proposed system's efficacy is supported by experimental results on two datasets, demonstrating higher accuracy compared to existing schemes. This empirical validation reinforces the practical utility and effectiveness of the proposed approach.

The proposed system offers a comprehensive solution that addresses the limitations of existing searchable encryption schemes by providing verifiability, enhanced security, semantic searching capabilities, flexibility, optimal matching, and empirical validation of its effectiveness. These advantages make it a promising approach for secure information retrieval in public cloud environments.

3.3 SYSTEM REQUIREMENTS

The system requirements for the development and deployment of the project as an application are specified in this section. The system requirements for implementing a secure verifiable semantic searching scheme over encrypted data in a public cloud environment include:

3.3.1 SOFTWARE REQUIREMENTS

The software requirements for this project encompass a comprehensive stack to support its development and deployment. MySQL serves as the database server, with SQLYOG as the client for database management. Apache Tomcat functions as the server, while Java forms the platform for application development. JEE technologies such as Servlets and JSP are utilized on the server-side, complemented by HTML, CSS, JavaScript, and AJAX for client-side functionalities. Eclipse serves as the integrated development environment (IDE), facilitating coding and debugging tasks. Rational Rose and SQL-Developer aid in UML design and E-R modeling. Testing is conducted using JUNIT for ensuring code reliability. Finally, DriveHQ provides cloud services for storage and deployment. This comprehensive suite of software ensures a robust and efficient development environment for the project.

Database Server	: MySQL
Database Client	: SQLYOG
Server	: Apache Tomcat
Platform	: Java
Server side Technologies	: JEE (Servlets, JSP)
Client Side Technologies	: HTML, CSS, JavaScript, AJAX
IDE	: Eclipse
UML Design/E-R Modeling Tools	: Rational Rose, SQL-Developer
Testing	: JUNIT
Cloud	: DriveHQ

3.3.2 HARDWARE REQUIREMENTS

The hardware requirements for implementing the proposed secure verifiable semantic searching scheme are relatively modest. A standard server infrastructure capable of running cryptographic operations efficiently is essential. This typically includes modern processors with support for cryptographic instructions for symmetric encryption operations, and optimized libraries for performing mathematical operations related to encryption schemes. Additionally, sufficient storage capacity is needed to store the encrypted data and intermediate results generated during the searching process. While there are no highly specialized hardware requirements, ensuring adequate computational resources and storage capacity is crucial for achieving efficient and scalable performance of the proposed scheme in practical cloud environments.

Processor	: i5
Ram	: 4GB
Hard Disk	: 500 GB

3.3.3 IMPLEMENTATION TECHNOLOGIES

The development of the secure verifiable semantic searching scheme involved a range of key technologies and tools. Here is a brief note on the implementation technologies used:

Java: Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language. Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak.

Some of applications of JAVA are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in, javatpoint.com, etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics

MySQL Database: MySQL is an open-source relational database management system (RDBMS) widely used for managing structured data. It offers a robust, scalable, and reliable platform for storing and retrieving data efficiently. MySQL supports various storage engines, providing flexibility to optimize performance and functionality for different use cases. Its SQL-based querying language enables users to interact with the database, execute complex queries, and perform data manipulation operations easily. MySQL's extensive community and documentation make it accessible for developers and administrators, offering a wealth of resources for troubleshooting and optimization. Additionally, MySQL integrates seamlessly with popular programming languages and frameworks, making it a preferred choice for web applications, e-commerce platforms, content management systems, and more. Overall, MySQL's combination of performance, reliability, and ease of use has made it a staple in the database ecosystem for both small-scale projects and large-scale enterprises.

Eclipse: Eclipse is an Integrated Development Environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

Java Development Tools: The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHP eclipse is a plug-in to eclipse that provides complete development tool for PHP.

In summary, the development of a Picture Translation App using Python involves a combination of programming languages, libraries, frameworks, and tools to enable image processing, text extraction, translation, user interface development, and data management. The choice of specific technologies may vary based on project requirements and developer preferences.

CHAPTER 4

SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture for secure verifiable semantic searching presents a comprehensive framework that seamlessly integrates encryption, matching algorithms, and verification mechanisms. At its core, the architecture utilizes a Word Transportation (WT) problem formulation to calculate the Minimum Word Transportation Cost (MWTC) for optimal matching on ciphertext, ensuring semantic similarity between queries and documents. This is achieved through a secure transformation of WT problems into random Linear Programming (LP) problems, allowing for encrypted MWTC computation. The LP duality theorem is then leveraged to design a verification mechanism, utilizing intermediate data from the matching process to verify the correctness of search results. By combining these elements, the architecture ensures both confidentiality and verifiability, providing a robust solution for secure information retrieval in public cloud environments.

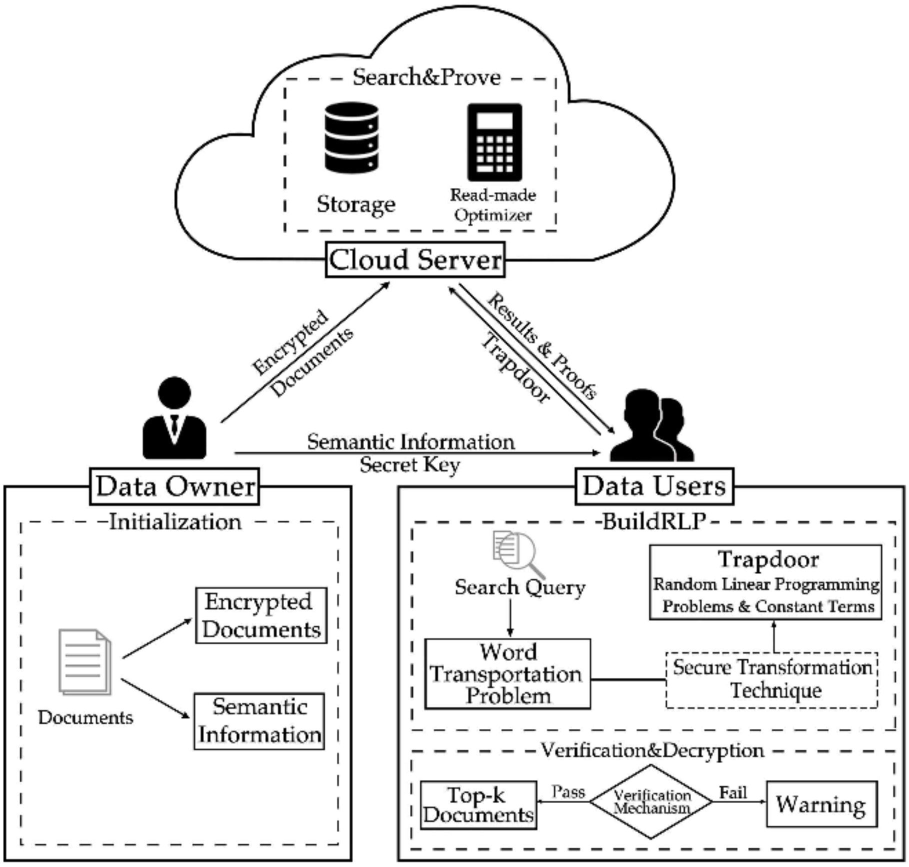


Figure 4.1.1 System Architecture

4.2 APPLICATION MODULES

The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams and in some cases pseudo codes.

4.2.1 ARCHITECTURAL DESIGN:

MVC stands for Model View and Controller. It is a design pattern that separates the business logic, presentation logic and data.
MVC Structure has the following three parts:

Controller acts as an interface between View and Model. Controller intercepts all the incoming requests.
Model represents the state of the application i.e. data. It can also have business logic.
View represents the presentation i.e. UI (User Interface).
Advantage of MVC Architecture

1. Navigation Control is centralized
2. Easy to maintain the large application

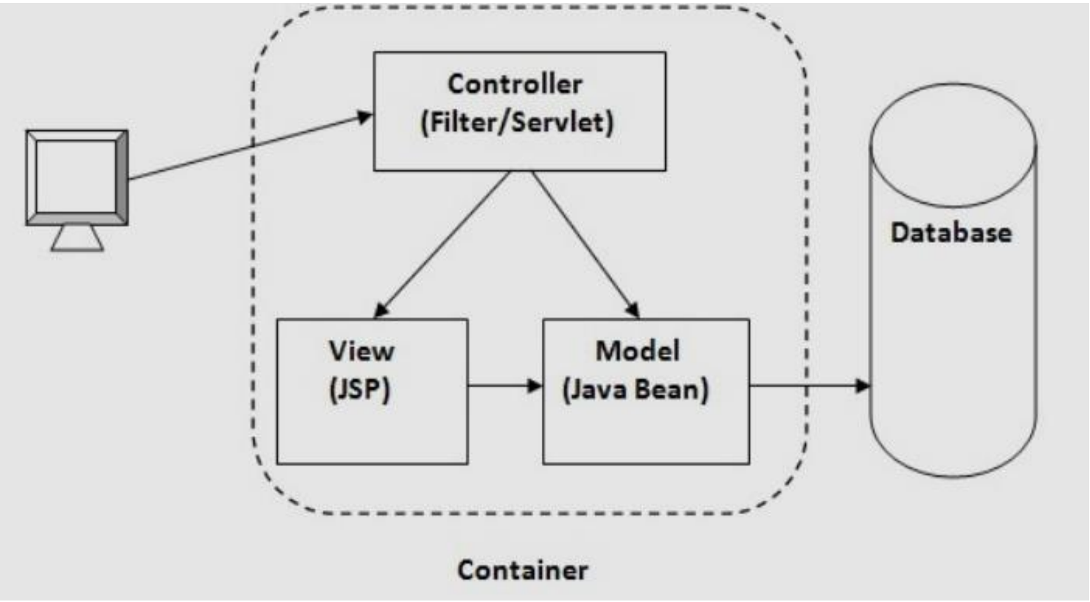


Figure 4.2.1 Architectural Design

4.2.2 DATA FLOW DIAGRAM:

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data Flow Diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

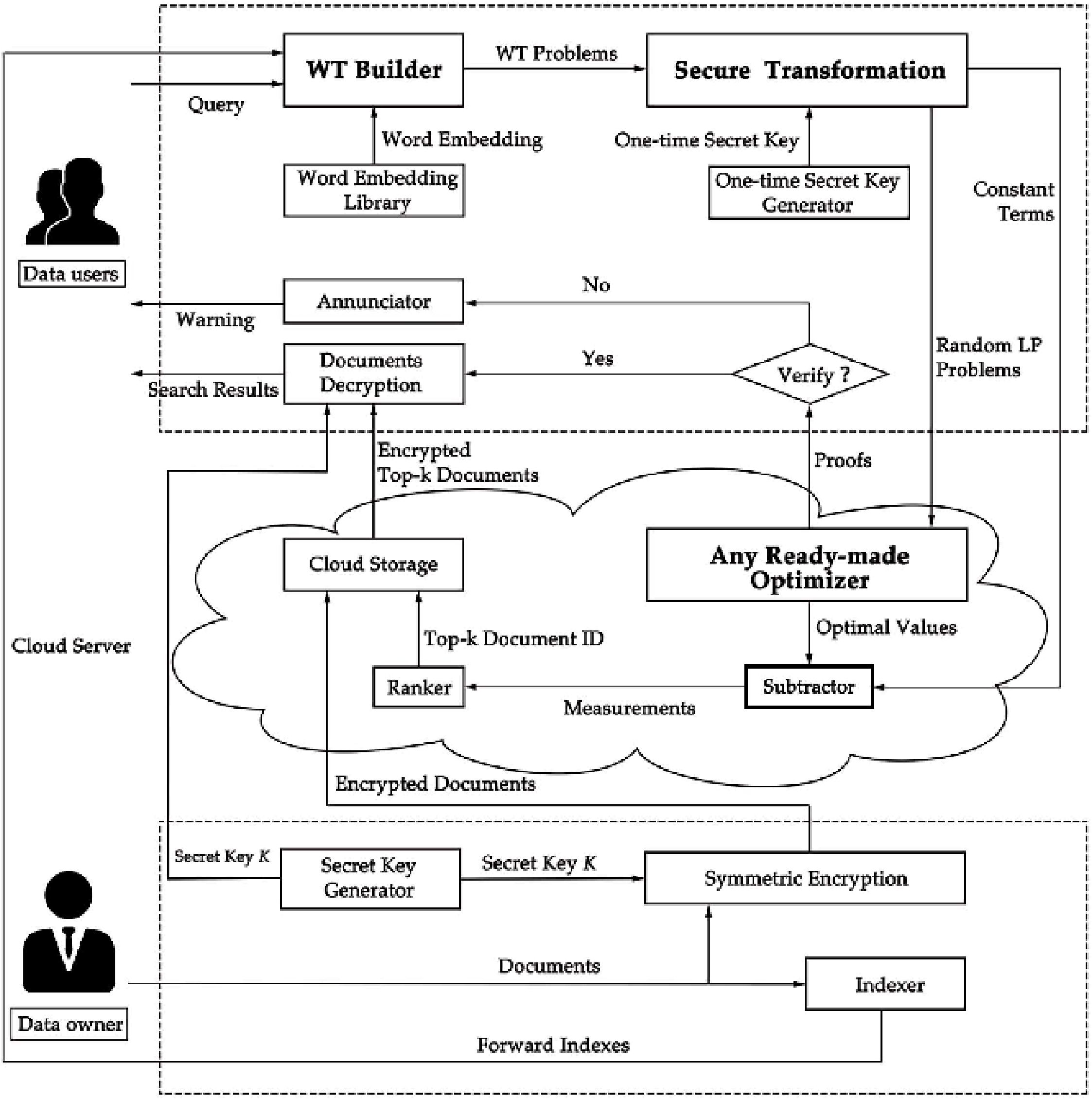


Figure 4.2.2 Data Flow Diagram

4.2.3 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

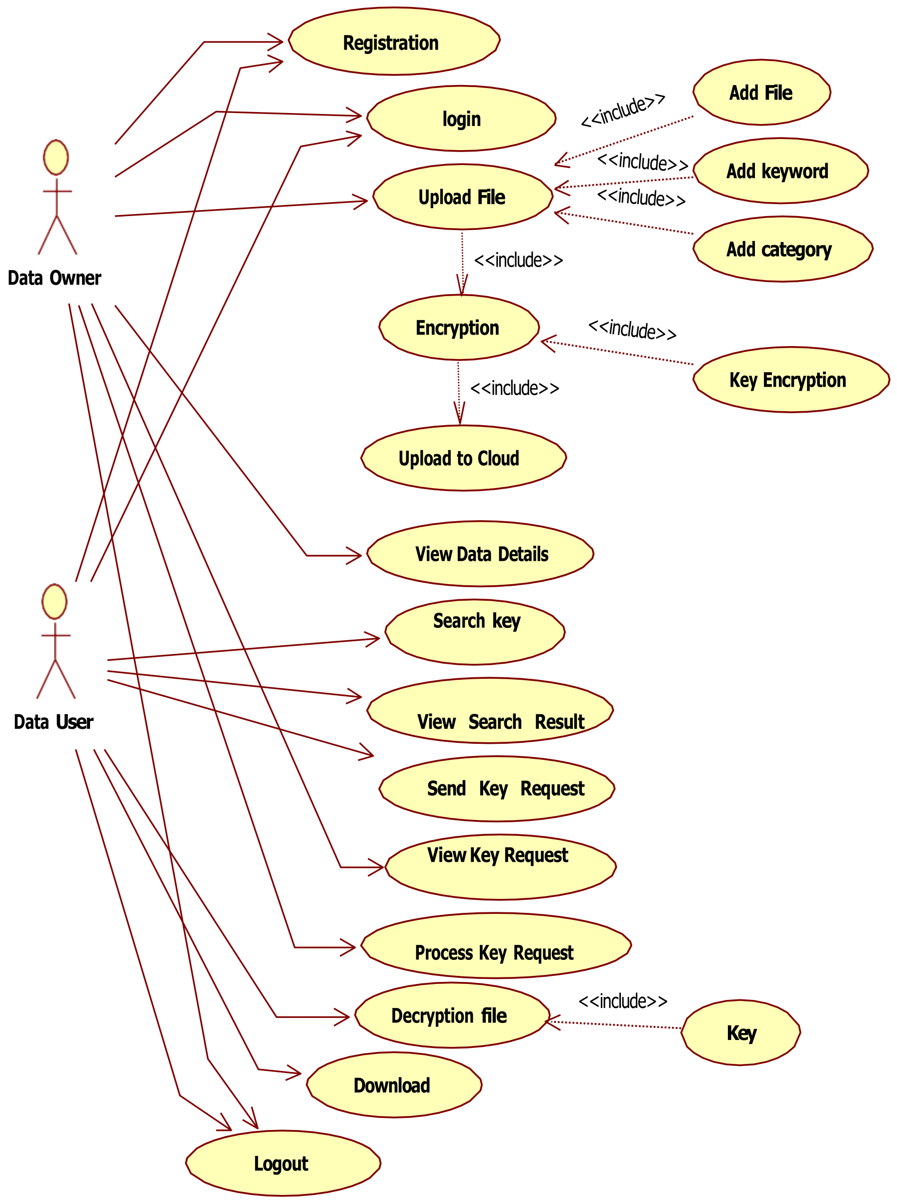


Figure 4.2.3 Use case diagram

4.2.4 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

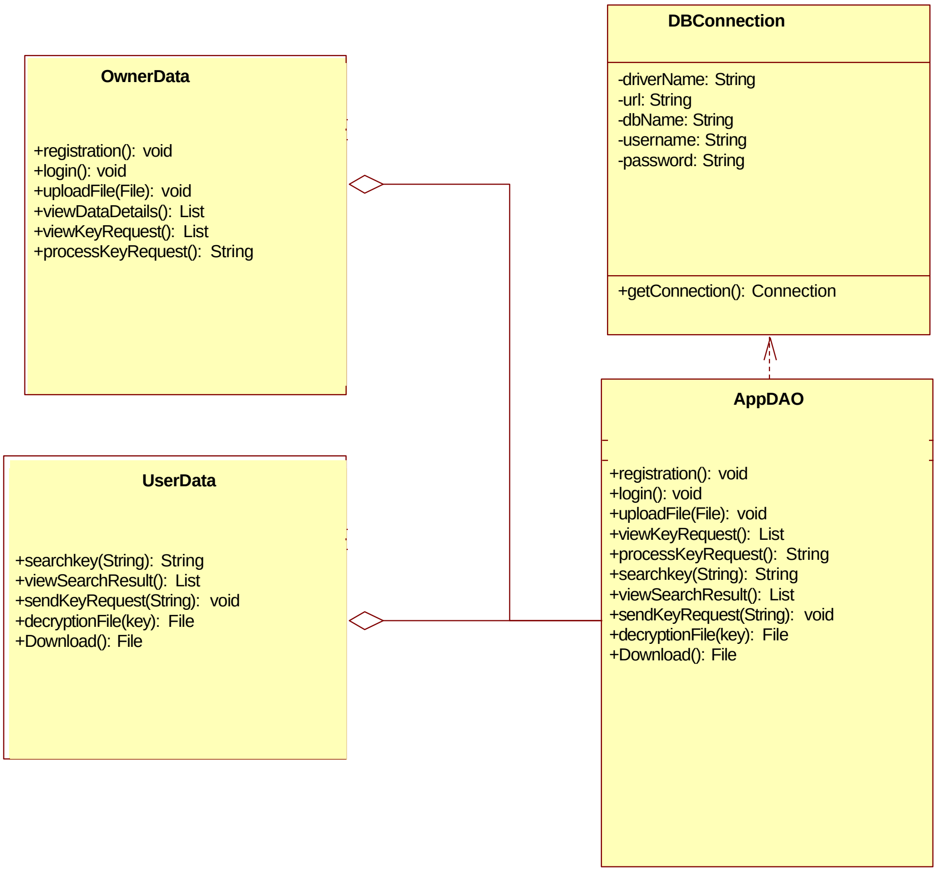


Figure 4.2.4 Class diagram

4.2.5 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

4.2.6 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

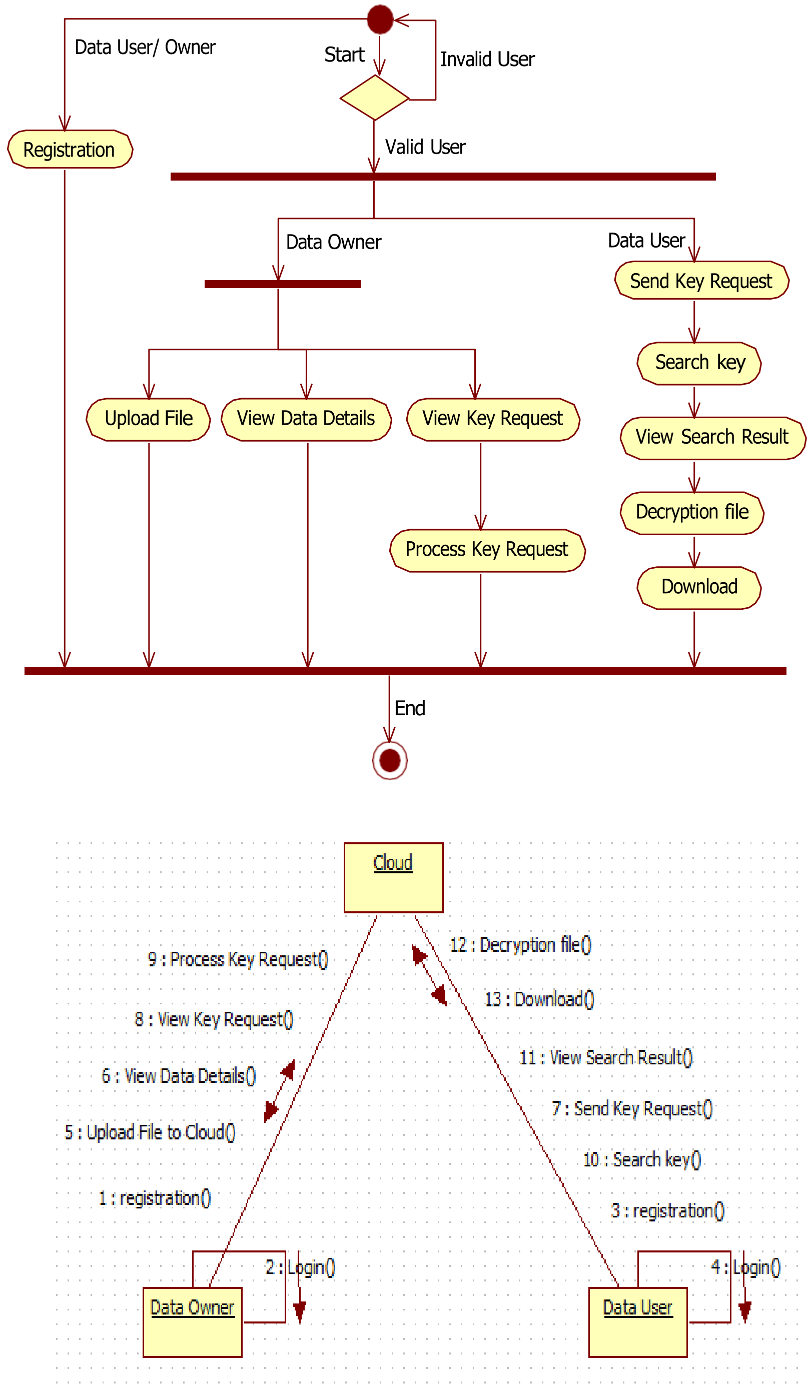


Figure 4.2.6 Activity diagram

4.2.7 DEPLOYMENT DIAGRAM

There may be more steps involved, depending on what specific requirements you have, but below are some of the main steps:

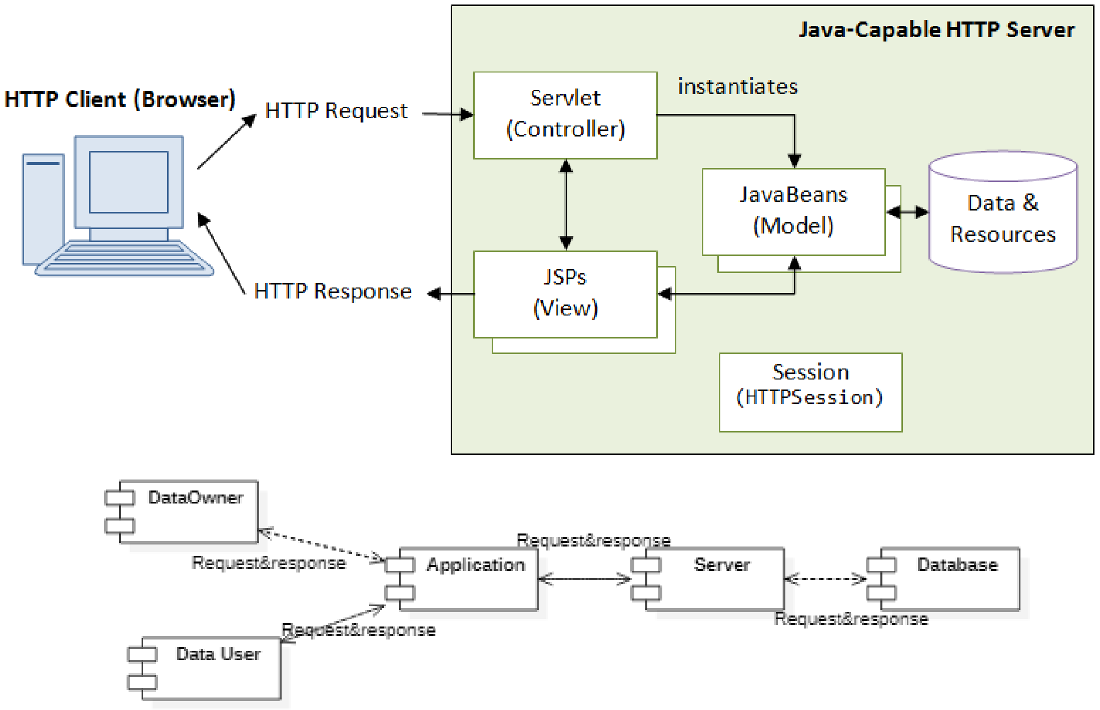


Figure 4.2.7 Deployment diagram

A deployment diagram in the Unified Modelling Language models the physical deployment of artic rafts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI). The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers. Device nodes are physically computing resources with processing memory and services to execute software, such as typical computer or mobile phones. An execution environment node is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

4.2.8 ENTITY RELATIONSHIP DIAGRAM:

An entity-relationship (ER) diagram is a graphical representation of entities and their relationships to each other, typically used in computing regarding the organization of data within databases or information systems.

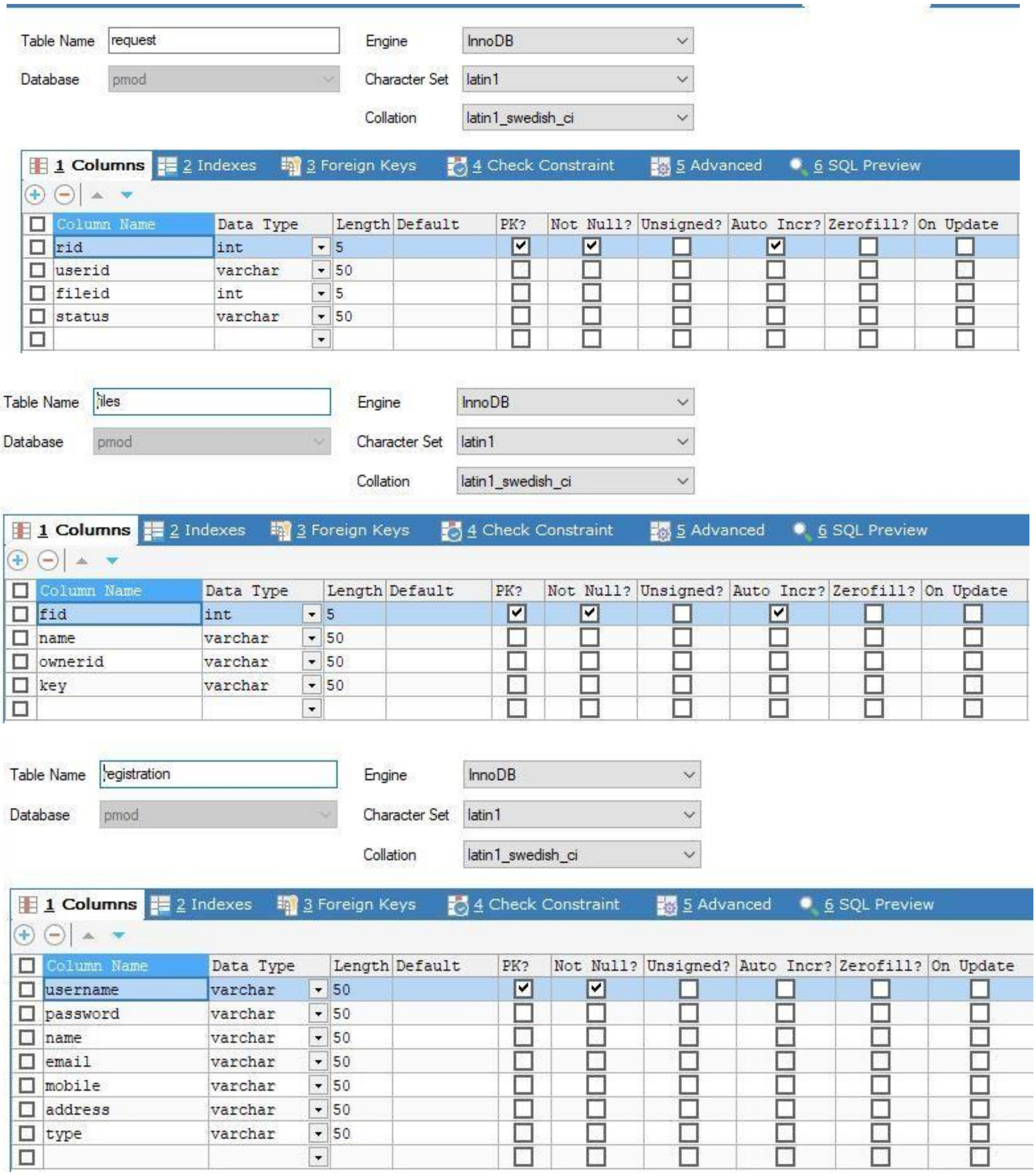


Table Name: request
Database: pmod
Engine: InnoDB
Character Set: latin1
Collation: latin1_swedish_ci

□	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
□	rid	int	5		☑	☑	□	☑	□	□
□	userid	varchar	50		□	□	□	□	□	□
□	fileid	int	5		□	□	□	□	□	□
□	status	varchar	50		□	□	□	□	□	□
□					□	□	□	□	□	□

Table Name: files
Database: pmod
Engine: InnoDB
Character Set: latin1
Collation: latin1_swedish_ci

□	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
□	fid	int	5		☑	☑	□	☑	□	□
□	name	varchar	50		□	□	□	□	□	□
□	ownerid	varchar	50		□	□	□	□	□	□
□	key	varchar	50		□	□	□	□	□	□
□					□	□	□	□	□	□

Table Name: registration
Database: pmod
Engine: InnoDB
Character Set: latin1
Collation: latin1_swedish_ci

□	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
□	username	varchar	50		☑	☑	□	□	□	□
□	password	varchar	50		□	□	□	□	□	□
□	name	varchar	50		□	□	□	□	□	□
□	email	varchar	50		□	□	□	□	□	□
□	mobile	varchar	50		□	□	□	□	□	□
□	address	varchar	50		□	□	□	□	□	□
□	type	varchar	50		□	□	□	□	□	□
□					□	□	□	□	□	□

Figure 4.2.8 Entity Relationship diagram

CHAPTER 5
IMPLEMENTATION

5.1 IMPLEMENTATION WITH HYPOTHETICAL SCENARIOS

In this section, we will delve into the implementation. We will cover various aspects of the implementation with hypothetical scenarios to illustrate how the project functions.

Here's a breakdown of the software architecture:

5.1.1 Login Page:

This page serves as the entry point for users, allowing them to authenticate and access the system's features securely.

5.1.2 User Registration Page:

Here, new users can create accounts by providing necessary information, such as username, email, and password, enabling them to utilize the system's functionalities.

5.1.3 Data Owner Registration Page:

This page facilitates the registration process for data owners, enabling them to register their data securely within the system for subsequent management and retrieval.

5.1.4 File Upload Page:

Users can utilize this page to upload files securely to the system, ensuring that their data is stored and managed in a protected environment.

5.1.5 Verification Page:

Upon certain actions or transactions, this page prompts users to verify their identity or confirm specific details, enhancing security and trust within the system's operations.

5.2 SOURCE CODE

- **<u>Logic to Connect Database:</u>**

```
 public static Connection getConnection() {
  Connection conn = null;
  String url = "jdbc:mysql://localhost:3306/";
  String dbName = "pri_multiKey";
  String driver = "com.mysql.jdbc.Driver";
  String userName = "root";
  String password = "root";
  try {
      Class.forName(driver).newInstance();
      conn = DriverManager.getConnection(url + dbName, userName, password);
      System.out.println("Connected to the database");
} catch (Exception e) {
      e.printStackTrace();
  }
  return conn;
}
```

- **<u>Logic to Connect to Cloud Server:</u>**

```
public synchronized boolean connect() {
   try {
      URL url = new URL("ftp://" + user + ":" + password + "@" + host + "/" +
      remoteFile + ";type=i");
      m_client = url.openConnection();
System.out.println(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.."+"ftp://"
+ user + ":" + password + "@" + host + "/" + remoteFile +
      ";type=i"); return true;
   } catch (Exception ex) {
```

```
            ex.printStackTrace();
            StringWriter sw0 = new StringWriter();
            PrintWriter p0 = new PrintWriter(sw0, true);
            ex.printStackTrace(p0);
            erMesg = sw0.getBuffer().toString();
            return false;
        }
    }
```

- **Logic to Upload File**

```
public synchronized boolean uploadFile(InputStream is) {
    //public synchronized boolean uploadFile(String localfilename) {
        try {
         //  InputStream is = new FileInputStream(localfilename);
            BufferedInputStream bis = new BufferedInputStream(is);
            OutputStream os = m_client.getOutputStream();
            BufferedOutputStream bos = new BufferedOutputStream(os);
            byte[] buffer = new byte[1024];
            int readCount;
            while ((readCount = bis.read(buffer)) > 0) {
                bos.write(buffer, 0, readCount);
            }
            bos.close();
            this.succMesg = "Uploaded!";
            return true;
        } catch (Exception ex) {
            ex.printStackTrace();
            StringWriter sw0 = new StringWriter();
            PrintWriter p0 = new PrintWriter(sw0, true);
            ex.printStackTrace(p0);
            erMesg = sw0.getBuffer().toString();
            return false;
        }
    }
```

- **Logic to Download File**

```
public synchronized boolean downloadFile(String localfilename) {
        try {
            InputStream is = m_client.getInputStream();
            BufferedInputStream bis = new BufferedInputStream(is);
            System.out.println(">>>>>>>>>>>"+localfilename);
            OutputStream os = new FileOutputStream(localfilename);
            BufferedOutputStream bos = new BufferedOutputStream(os);
            byte[] buffer = new byte[1024];
            int readCount;

            while ((readCount = bis.read(buffer)) > 0) {
                bos.write(buffer, 0, readCount);
            }
            bos.close();
            is.close(); // close the FTP inputstream
            this.succMesg = "Downloaded!";
            return true;
        }
        catch (Exception ex) {
            ex.printStackTrace();
            StringWriter sw0 = new StringWriter();
            PrintWriter p0 = new PrintWriter(sw0, true);
            ex.printStackTrace(p0);
            erMesg = sw0.getBuffer().toString();
            return false;
        }
```

- **Logic to Encrypt file**

```
public String encrypt(String unencryptedString) {
        String encryptedString = null;
        try {

            cipher.init(Cipher.ENCRYPT_MODE, key);

            byte[] plainText = unencryptedString.getBytes(UNICODE_FORMAT);
            byte[] encryptedText = cipher.doFinal(plainText);
            encryptedString = new String(Base64.encodeBase64(encryptedText));

        } catch (Exception e) {
            e.printStackTrace();
        }

        return encryptedString;

    }
```

- **Logic to Decrypt**

```
    public String decrypt(String encryptedString) {
        String decryptedText=null;
        try {

            cipher.init(Cipher.DECRYPT_MODE, key);

            byte[] encryptedText = Base64.decodeBase64(encryptedString);
            byte[] plainText = cipher.doFinal(encryptedText);
            decryptedText= new String(plainText);
        } catch (Exception e) {
            e.printStackTrace();
        }

        return decryptedText;


    }
```

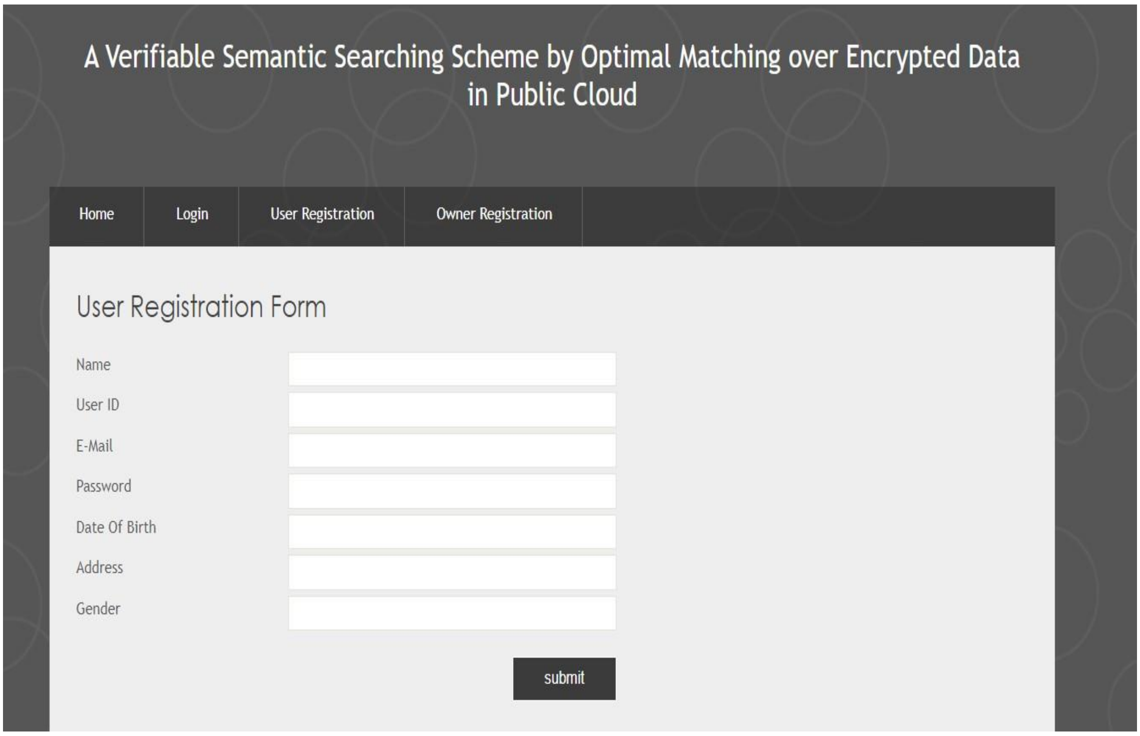
CHAPTER 6

RESULTS

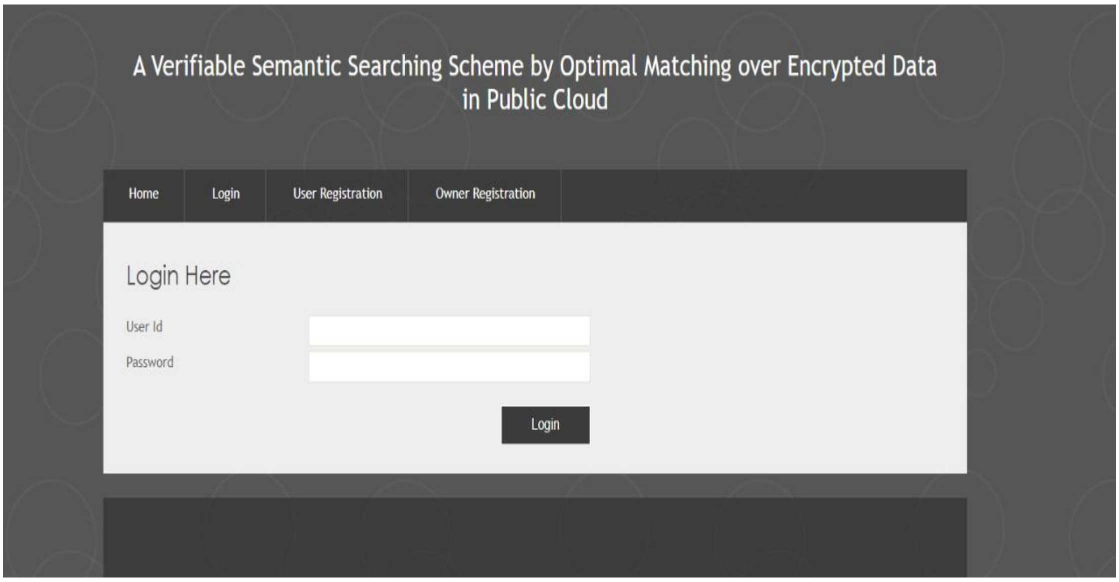
The proposed secure verifiable semantic searching scheme addresses limitations of existing approaches by introducing a novel method based on the word transportation problem and random linear programming. Experimental results on two datasets validate the effectiveness of the proposed scheme, exhibiting higher accuracy compared to prior approaches.

EXPERIMENT SCREENSHOTS

The following are the results of the implementation of the developed application:



User Registration Page



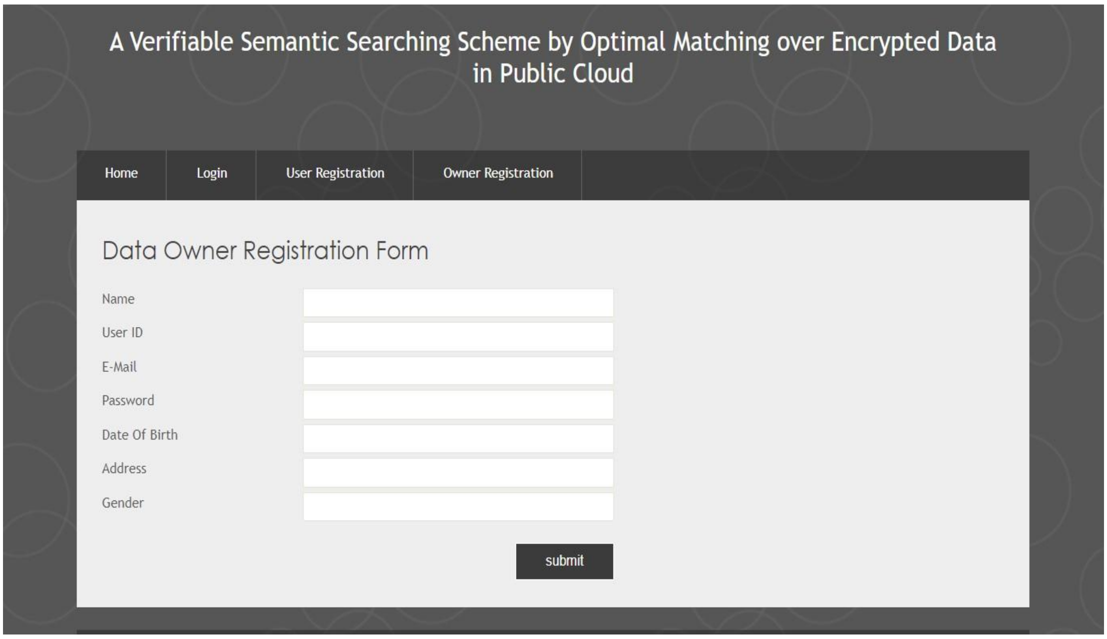
A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Login Here

User Id

Password

Login Page



A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Data Owner Registration Form

Name

User ID

E-Mail

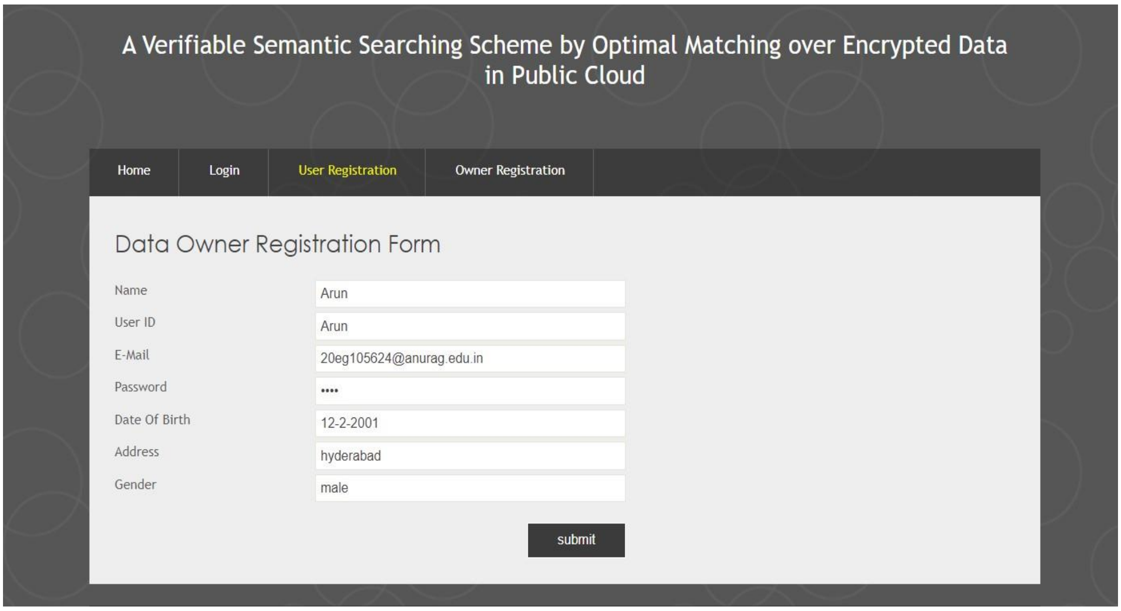
Password

Date Of Birth

Address

Gender

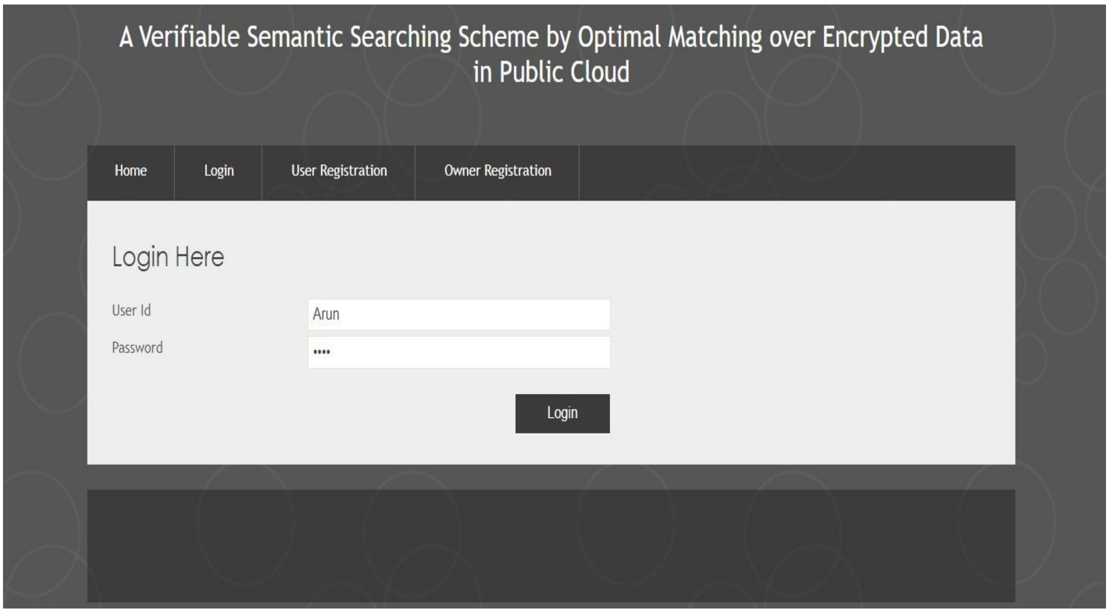
Data Owner Registration Page


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud
Home
Login
User Registration
Owner Registration
submit

Data Owner Registration Form

Name	Arun
User ID	Arun
E-Mail	20eg105624@anurag.edu.in
Password	••••
Date Of Birth	12-2-2001
Address	hyderabad
Gender	male

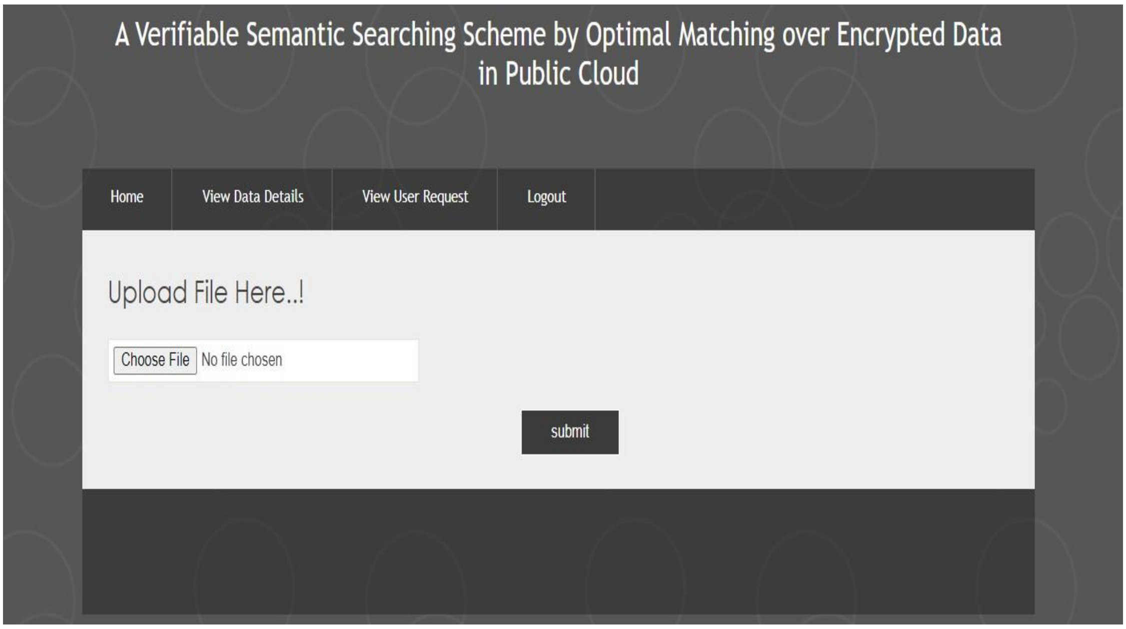
Signing in Data Owner Registration Form


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud
Home
Login
User Registration
Owner Registration
Login

Login Here

User Id	Arun
Password	••••

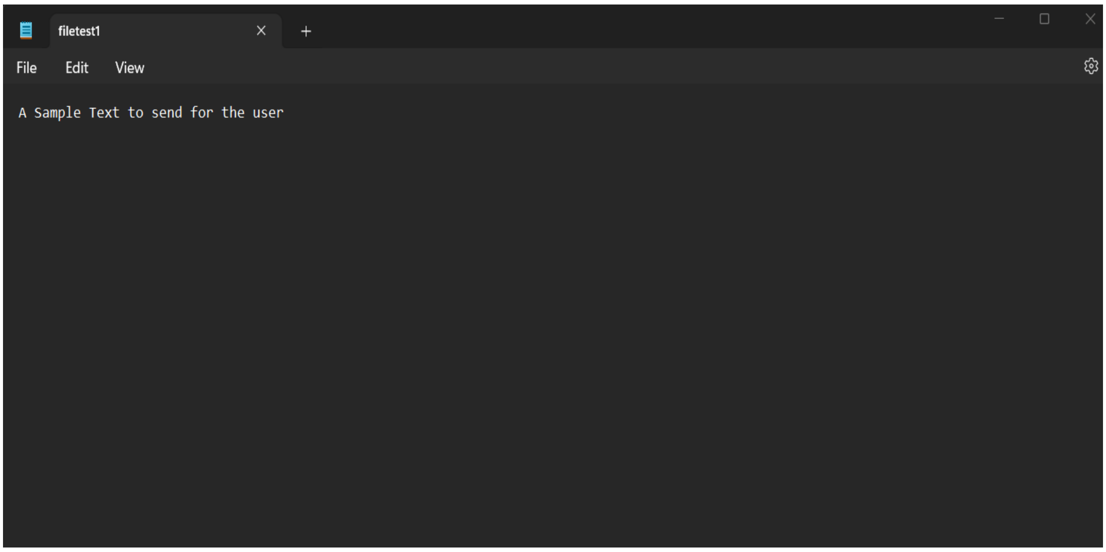
Loging in



A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

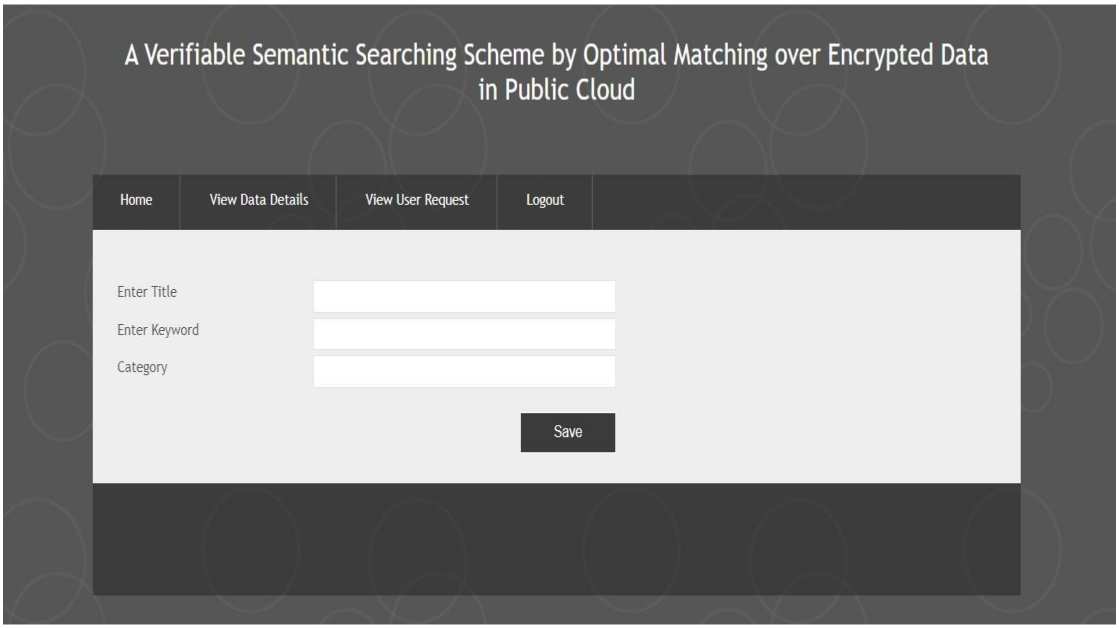
Upload File Here..!

File Upload Page

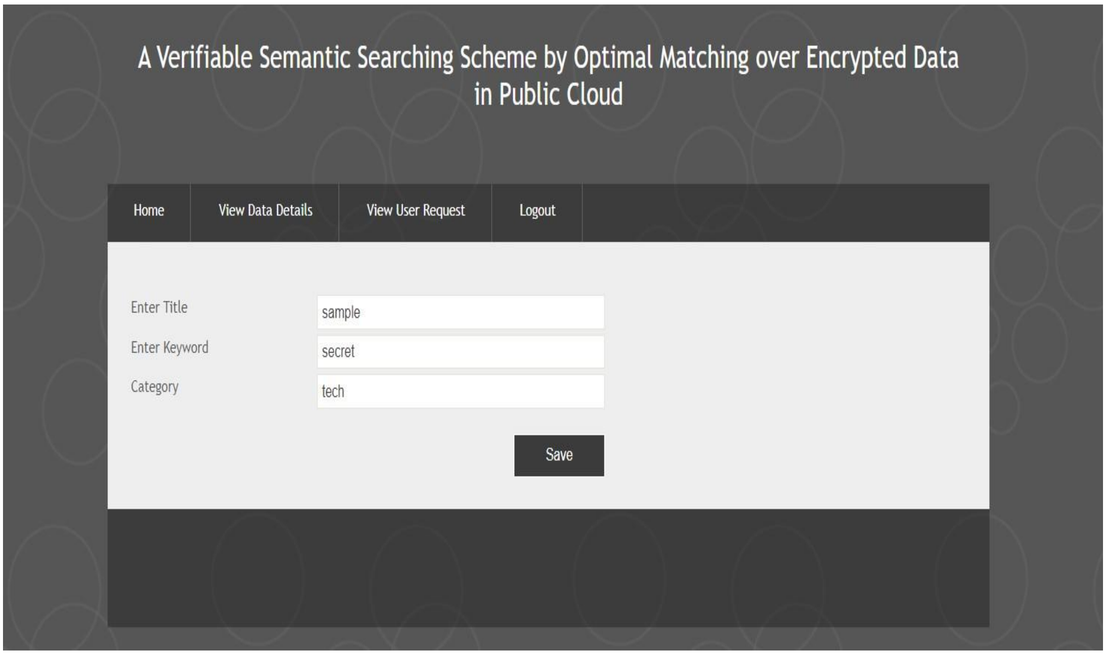


```
A Sample Text to send for the user
```

Contents of the File


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud
Home
View Data Details
View User Request
Logout
Enter Title
Enter Keyword
Category
Save

Text Details

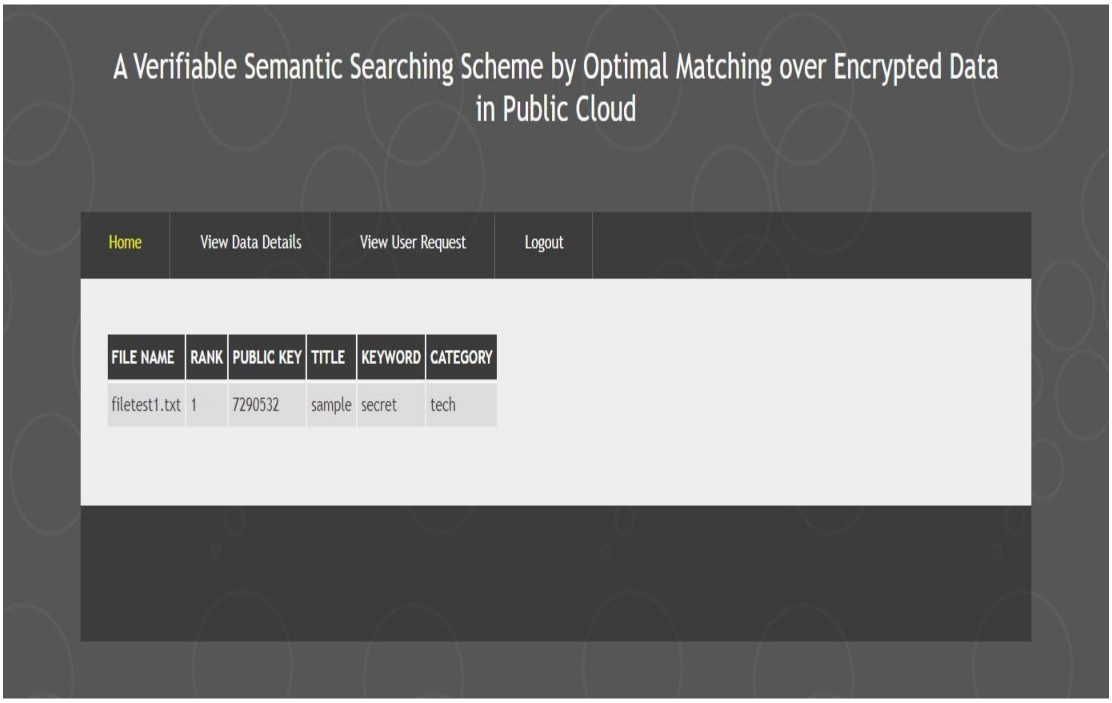

A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud
Home
View Data Details
View User Request
Logout
Save

Enter Title: sample

Enter Keyword: secret

Category: tech

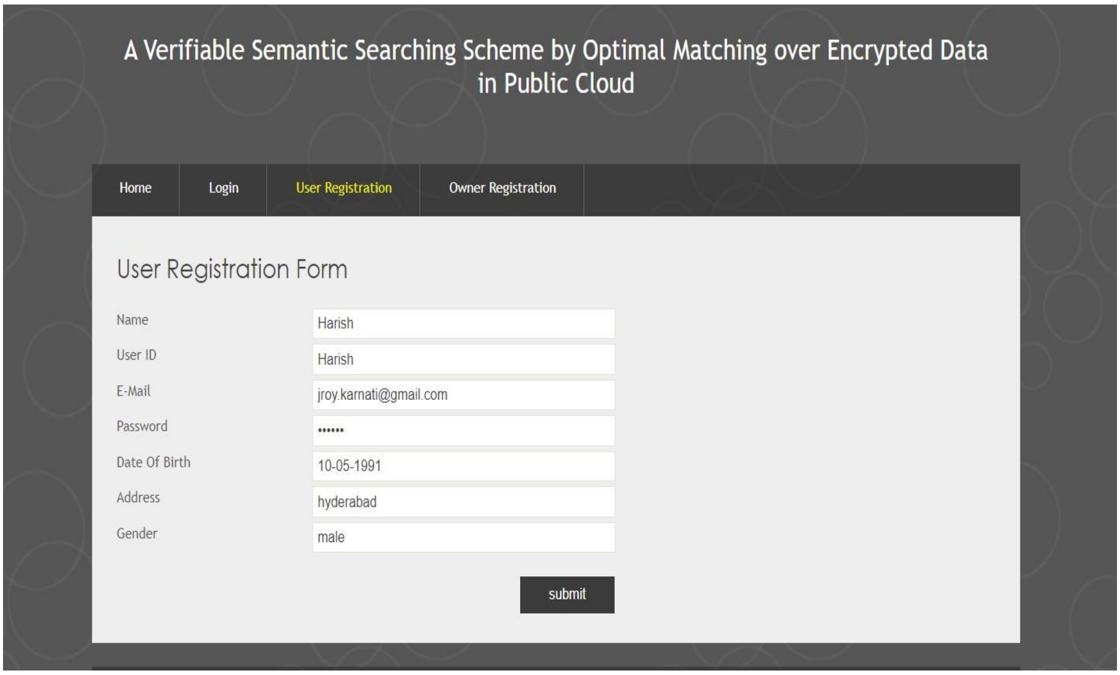
Filling Text Details



A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

FILE NAME	RANK	PUBLIC KEY	TITLE	KEYWORD	CATEGORY
filetest1.txt	1	7290532	sample	secret	tech

Displaying File Details

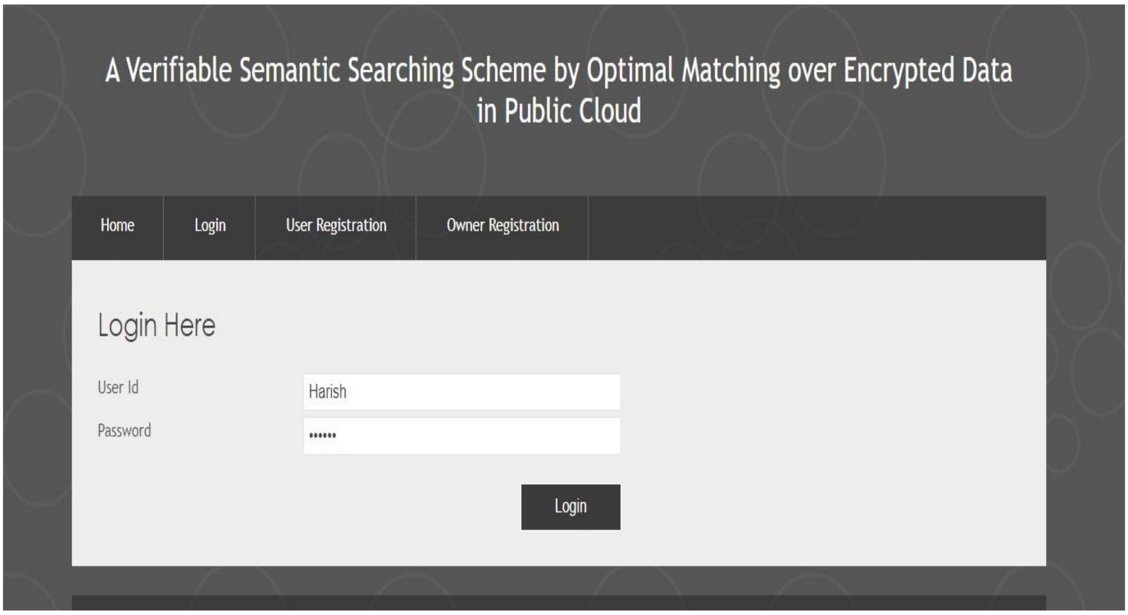


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

User Registration Form

Name	Harish
User ID	Harish
E-Mail	jroy.karnati@gmail.com
Password	••••••
Date Of Birth	10-05-1991
Address	hyderabad
Gender	male

User Registration Page



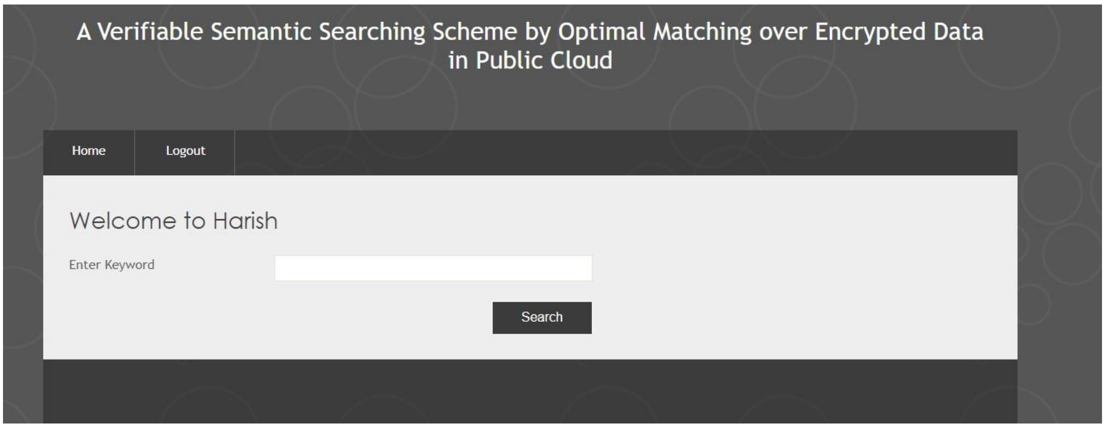
A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Login Here

User Id: Harish

Password: ••••••

User Loging in

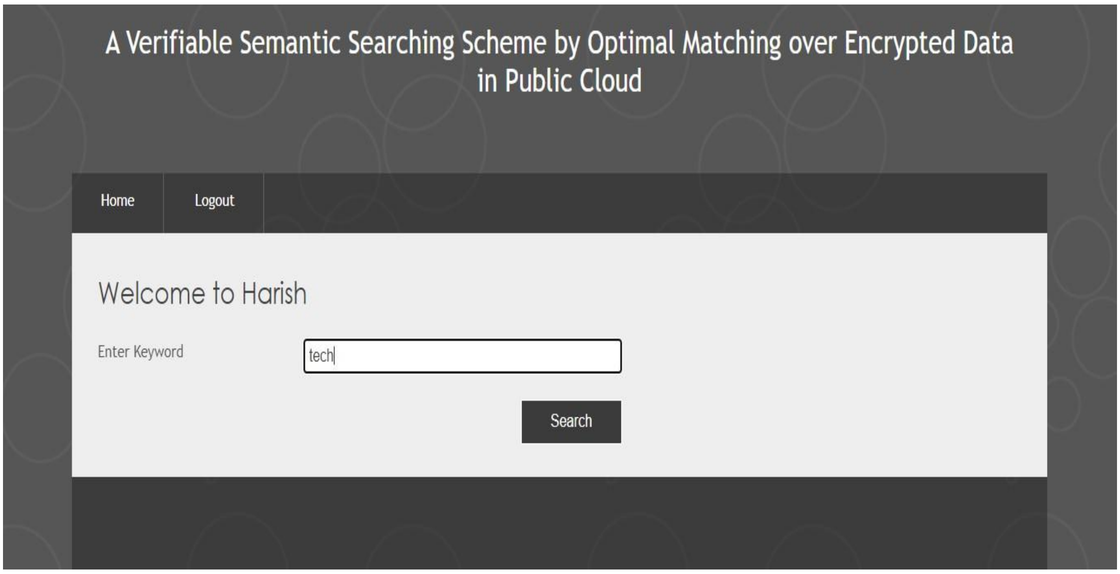


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Welcome to Harish

Enter Keyword:

Search Page

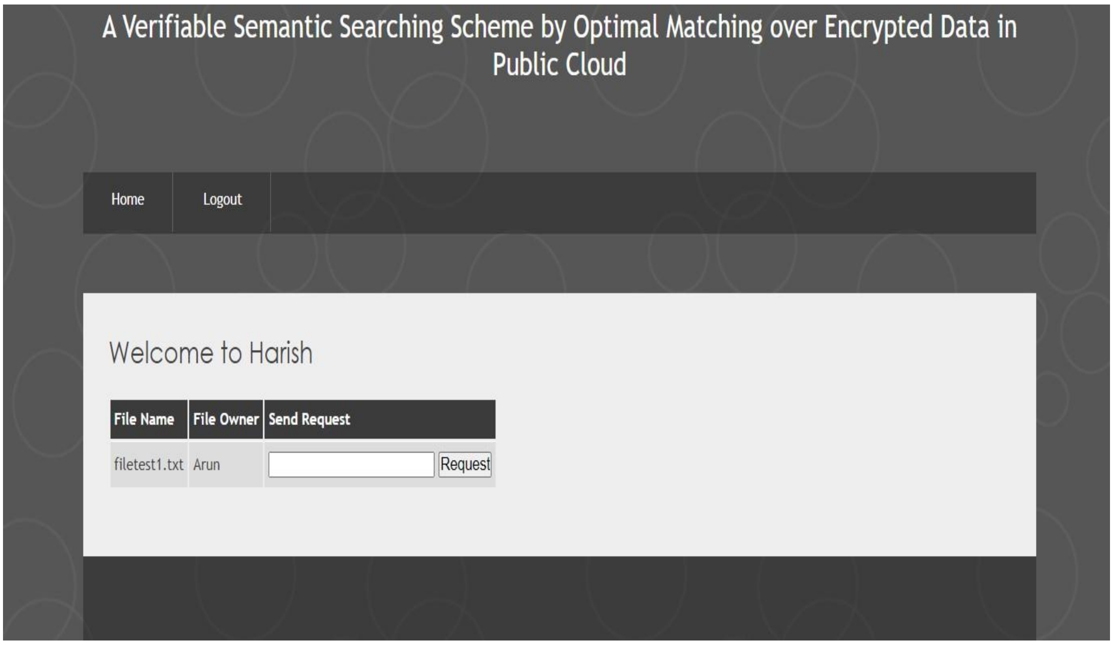


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Welcome to Harish

Enter Keyword: tech

Searching the Keyword

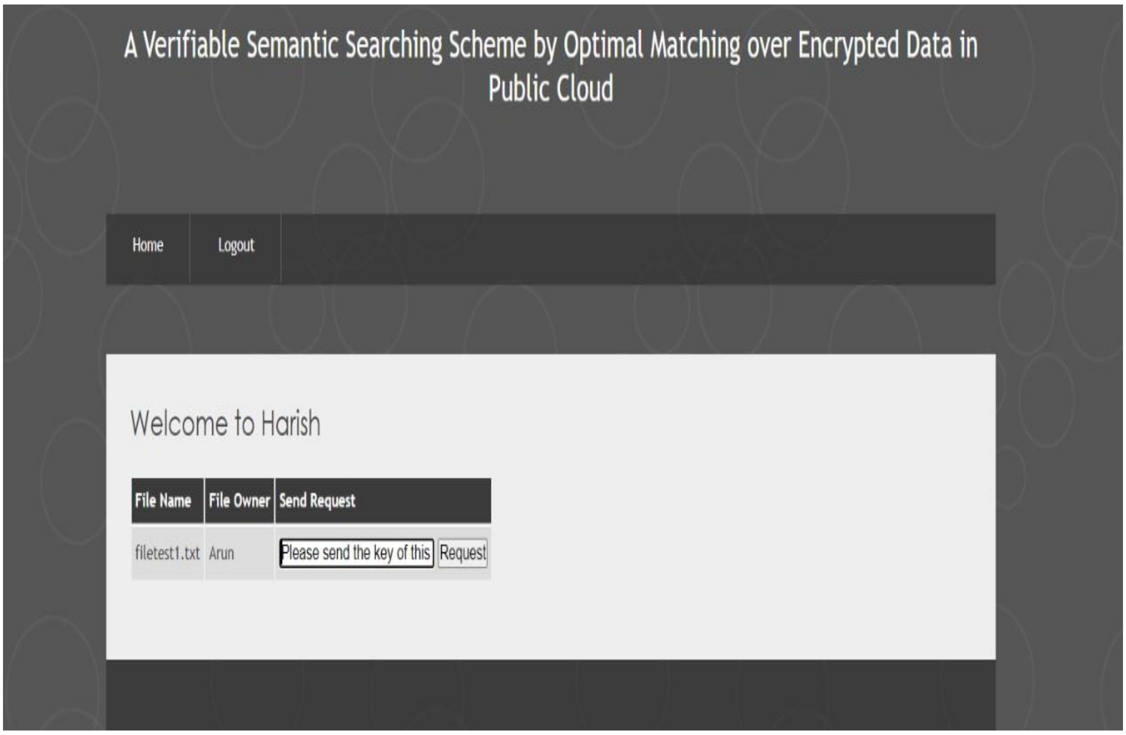


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Welcome to Harish

File Name	File Owner	Send Request
filetest1.txt	Arun	Request

Requesting an Email

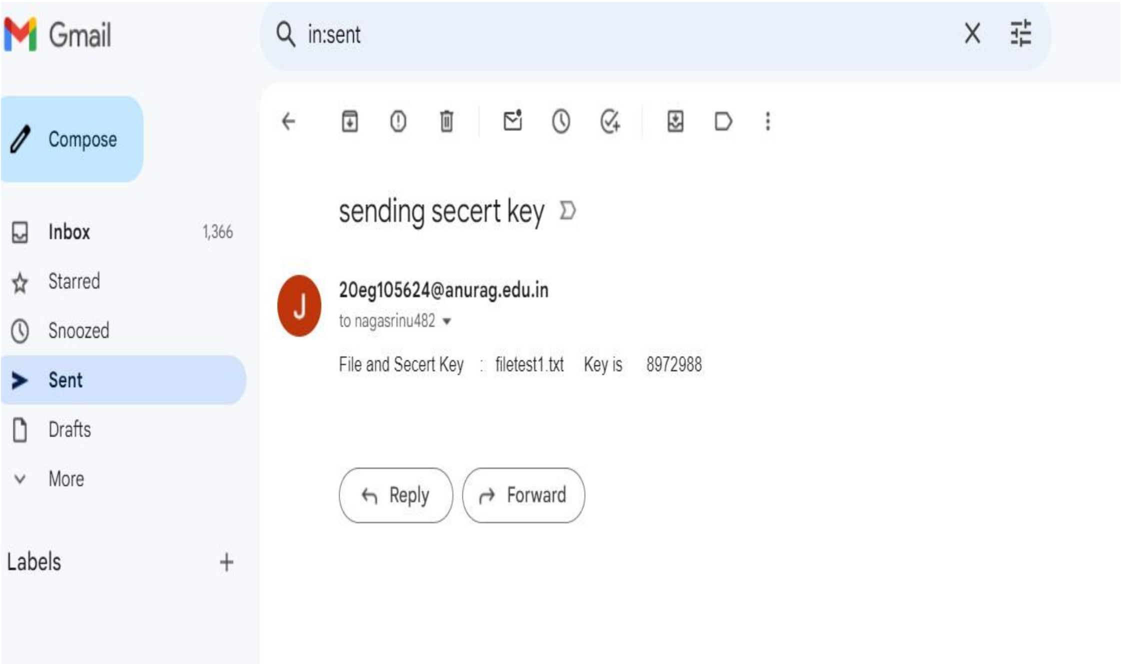


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Welcome to Harish

File Name	File Owner	Send Request
filetest1.txt	Arun	Please send the key of this Request

Sending an email request for Secret Key



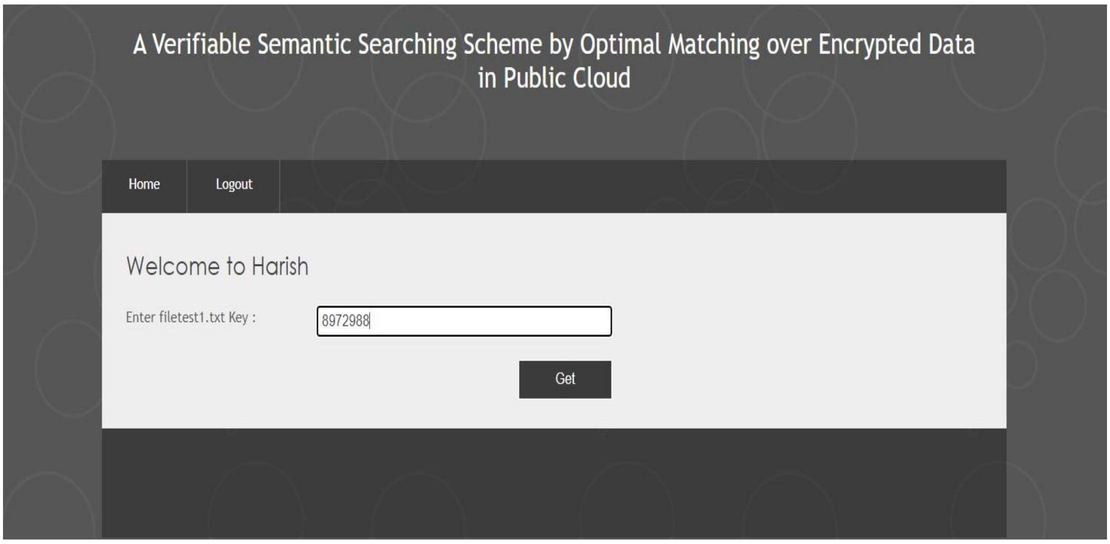
sending secert key

20eg105624@anurag.edu.in

to nagasrinu482

File and Secert Key : filetest1.txt Key is 8972988

Secret Key received through Email

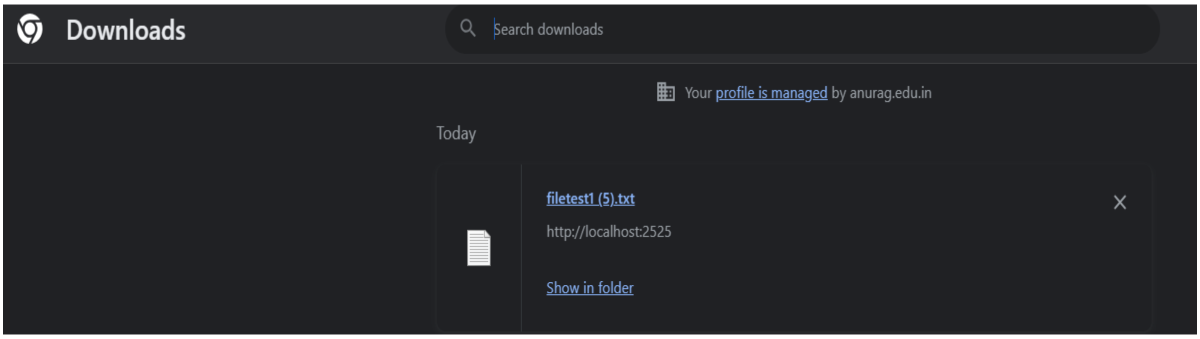


A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Welcome to Harish

Enter filetest1.txt Key : 8972988

Using secret Key to Access the File



Downloads

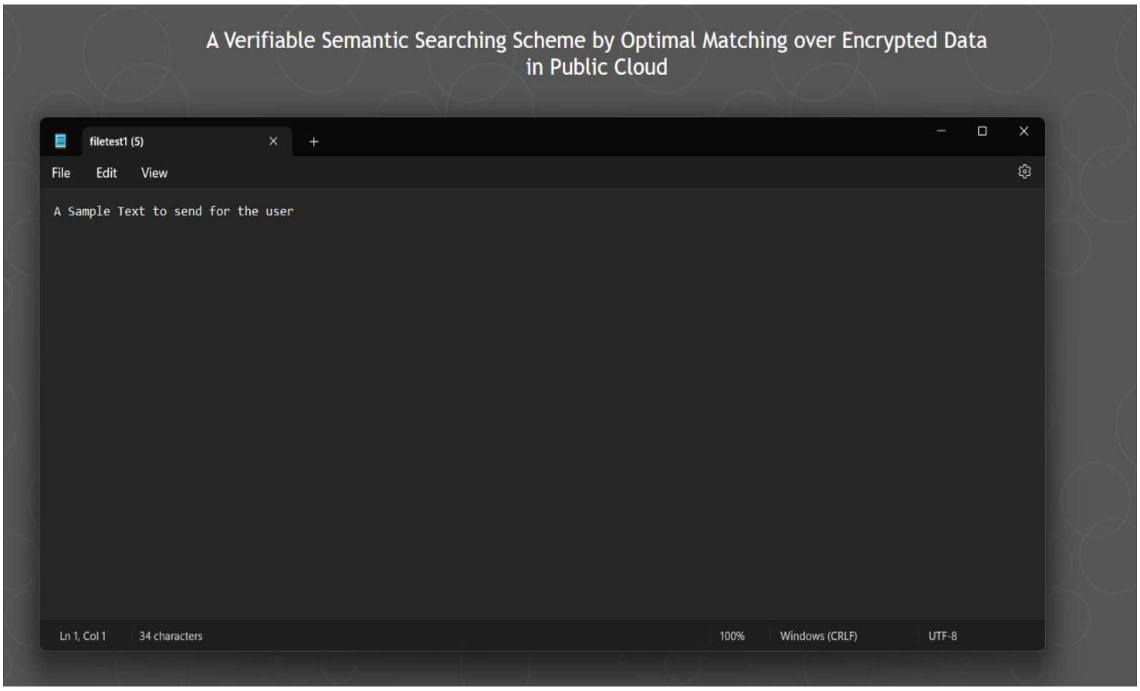
Your profile is managed by anurag.edu.in

Today

filetest1 (5).txt

http://localhost:2525

Downloading the File



A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

```
A Sample Text to send for the user
```

Accessing File Contents

Test Cases:

Test ID	Test Name	Inputs	Process	Excepted Output	Actual Output	Status
1	Login Test	User Name, Password	Validate Username and password on database	Need to redirect to user home page	It's Redirected to user home	Success
2	Registration Test	Username, password, email, mobile number, email etc…	Insert the users into database	Need to insert the user details into database	It's inserted	Success
3	Upload file	File, UserId.	Insert the files into the cloud	Need to insert the files into the database	It's inserted	Success
4	Grant access	User id, Permission	Grant the access to the user to access the files	Need to grant permission to user	Permission granted	Success
5	Send Mail	User id, mail id, key, message	Send these inputs to specified user mail id	Need to send the details to mail id	Mail forwarded	success
6	Download File	Userid, file id, key permission	If given user having access to files and if he entered the right file key to download the file	Download the file if he specified the right details	File downloaded	Success

CHAPTER 7
CONCLUSION

We propose a secure verifiable semantic searching scheme that treats matching between queries and documents as a word transportation optimal matching task. Therefore, we investigate the fundamental theorems of Linear Programming (LP) to design the Word Transportation (WT) problem and a result verification mechanism. We formulate the WT problem to calculate the Minimum Word Transportation Cost (MWTC) as the similarity metric between queries and documents, and further propose a secure transformation technique to transform WT problems into random LP problems. Therefore, our scheme is simple to deploy in practice as any ready-made optimizer can solve the RLP problems to obtain the encrypted MWTC without learning sensitive information in the WT problems. Meanwhile, we believe that the proposed secure transformation technique can be used to design other privacy preserving linear programming applications. We bridge the semantic-verifiable searching gap by observing an insight that using the intermediate data produced in the optimal matching process to verify the correctness of search results. Specifically, we investigate the duality theorem of LP and derive a set of necessary and sufficient conditions that the intermediate data must meet. The experimental results on two TREC collections show that our scheme has higher accuracy than other schemes. In the future, we plan to research on applying the principles of secure semantic searching to design secure cross- language searching schemes.

SUMMARY

Existing semantic searching schemes in public cloud environments face challenges in providing verifiable searching while ensuring flexibility in queries and search results. These schemes often rely on predefined keywords for forecasted results, leading to limitations in accuracy. Additionally, queries are expanded on plaintext, restricting the scope of exact matching with semantically extended words. To address these shortcomings, a novel secure verifiable semantic searching scheme is proposed in this paper.

The proposed scheme introduces a formulation of the Word Transportation (WT) problem to compute the Minimum Word Transportation Cost (MWTC) for semantic optimal matching on ciphertext. By transforming WT problems into random linear programming (LP) problems, the encrypted MWTC is obtained, ensuring confidentiality. Moreover, leveraging the duality theorem of LP, a verification mechanism is designed using intermediate data generated during the matching process to verify the correctness of search results. This approach not only enhances security but also ensures the verifiability of retrieved information.

Through comprehensive security analysis, the proposed scheme is shown to guarantee both verifiability and confidentiality. Experimental evaluation conducted on two datasets demonstrates that the scheme outperforms existing methods in terms of accuracy, making it a promising solution for secure semantic searching over encrypted data in public cloud environments.

FUTURE ENHANCEMENTS AND DISCUSSIONS

Future enhancements and discussions for this proposed secure verifiable semantic searching scheme could focus on several aspects to further improve its effectiveness and applicability in real-world scenarios.

Firstly, exploring the scalability of the scheme is crucial, especially as data volumes continue to grow exponentially. Investigating methods to efficiently handle large-scale datasets while maintaining computational efficiency and maintaining security standards would be valuable. This could involve optimizing the cryptographic operations, refining the matching algorithms, or exploring parallel computing techniques to distribute the workload across multiple processing units.

Secondly, considering the evolving nature of language and the diversity of user queries, enhancing the scheme to handle more complex semantic relationships and nuances in language semantics would be beneficial. This could involve incorporating advanced natural language processing techniques, such as deep learning-based models, to better understand the context and intent behind user queries. Additionally, integrating feedback mechanisms to adapt and improve the matching accuracy over time based on user interactions and feedback would enhance the overall user experience.

Furthermore, extending the scheme to support multi-keyword searches and more sophisticated search functionalities could broaden its utility. This could include enabling boolean operators, phrase searching, or proximity searching to allow users to express more complex search criteria. Additionally, exploring ways to integrate the scheme with existing cloud-based search platforms or services to provide a seamless and integrated search experience for users would be valuable. This could involve developing APIs or plugins that enable easy integration with popular cloud platforms and services, thereby facilitating adoption and deployment in real-world environments. Overall, these future enhancements and discussions aim to further refine and extend the proposed scheme to meet the evolving demands of secure and flexible information retrieval in public cloud environments.

CHAPTER 8

REFERENCES

[1] Guoxiu Liu, Geng Yang, Shuangjie Bai and Qiang Zhou "Effective Fuzzy Semantic Searchable Encryption Scheme Over Encrypted Cloud Data", IEEE,2020

[2] S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan and C. Glackin, "A new secure and lightweight searchable encryption scheme over encrypted cloud data", IEEE Trans. Emerg. Topics Comput., vol. 7, no. 4, pp. 530-544, Oct. 2019.

[3] Xuelong Dai, Hua Dai, Chunming Rong and Fu Xiao,"Enhanced Semantic-Aware Multi-Keyword Ranked Search Scheme Over Encrypted Cloud Data"Oct.-Dec. 2022, pp. 2595-2612, vol. 10

[4] H. Shen, L. Xue, H. Wang, L. Zhang, and J. Zhang.B+-tree based multi-keyword ranked similarity search scheme over encrypted cloud data.IEEE Access, 9:150865–150877, 2021a.

[5] D. Sharma.Searchable encryption : A survey.Information Security Journal: A Global Perspective, 32(2):76–119, Mar. 2023.

[6] P. Srivani, S. Ramachandram, and R. Sridevi.Multi-key searchable encryption technique for index-based searching.International Journal of Advanced Intelligence Paradigms, 22(1-2):84–98, Jan. 2022.

[7] G. Sucharitha, V. Sitharamulu, S. N. Mohanty, A. Matta, and D. Jose.Enhancing secure communication in the cloud through blockchain assisted-cp-dabe.IEEE Access, 11:99005–99015, 2023.

[8] Y. Tang, Y. Chen, Y. Luo, S. Dong, and T. Li.VR-PEKS: A Verifiable and Resistant to Keyword Guess Attack Public Key Encryption with Keyword Search Scheme, 2023.

[9] M. Ali, H. He, A. Hussain, M. Hussain, and Y. Yuan.Efficient Secure Privacy Preserving Multi Keywords Rank Search over Encrypted Data in Cloud Computing.Journal of Information Security and Applications, 75:103500, 2023.

[10] Lanxiang Chen; Yujie Xue; Yi Mu; Lingfang Zeng; Fatemeh Rezaeibagha," CASE-SSE: Context-Aware Semantically Extensible Searchable Symmetric Encryption for Encrypted Cloud Data" ,IEEE,March-April 2023D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Secur. Privacy, 2000, pp. 44– 55.

[11] Z. Fu, J. Shu, X. Sun, and N. Linge, “Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data,” IEEE Trans. Consum. Electron., vol. 60, no. 4, pp. 762–770, 2014.

[12] T. S. Moh and K. H. Ho, “Efficient semantic search over encrypted data in cloud computing,” in Proc. IEEE. Int. Conf. High Perform. Comput. Simul., 2014, pp. 382

[13] N. Jadhav, J. Nikam, and S. Bahekar, “Semantic search supporting similarity ranking over encrypted private cloud data,” Int. J. Emerging Eng. Res. Technol., vol. 2, no. 7, pp. 215–219, 2014.

[14] Z. H. Xia, Y. L. Zhu, X. M. Sun, and L. H. Chen, “Secure semantic expansion based search over encrypted cloud data supporting similarity ranking,” J. Cloud Comput., vol. 3, no. 1, pp. 1–11, 2014. [7] Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, “Semantic- aware searching over encrypted data for cloud computing,” IEEE Trans. Inf. Forensics Security, vol. 13, no. 9, pp. 2359–2371, Sep. 2018.

[15] Z. J. Fu, X. L. Wu, Q. Wang, and K. Ren, “Enabling central keywordbased semantic extension search over encrypted outsourced data,” IEEE Trans. Inf. Forensics Security., vol. 12, no. 12, pp. 2986–2997, 2017.

[16] Y. G. Liu and Z. J. Fu, “Secure search service based on word2vec in the public cloud,” Int. J. Comput. Sci. Eng., vol. 18, no. 3, pp. 305–313, 2019. [10] E. J. Goh, “Secure indexes.” IACR Cryptology ePrint Archive, vol. 2003, pp. 216–234, 2003.

[17] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” J. Comput. Secur., vol. 19, no. 5, pp. 895–934, 2011.

[18] M. S. Islam, M. Kuzu, and M. Kantarcioglu, “Access pattern disclosure on searchable encryption: Ramification, attack and mitigation.” in Proc. ISOC Network Distrib. Syst. Secur. Symp., vol. 20, 2012, pp. 12–26.