A
Project Report
on

TOWARDS EFFICIENT CRYPTOGRAPHIC DATA VALIDATION SERVICE IN EDGE COMPUTING

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Technology
In
COMPUTER SCIENCE AND ENGINEERING
by

ERRAGUNTA HARSHA REDDY

(20EG105312)

GOLLA SANTHOSH KUMAR

(20EG105315)

REDDAVENI SAI KRISHNA

(20EG105330)

Under the guidance of

Mrs. P. Aparna

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANURAG UNIVERSITY

VENTAKAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088

TELANGANA

(2023-2024)

DEPARTMENT OF COMPUTER SCIENCE AND ENGNIEERING

CERTIFICATE

This is to certify that the project entitled **"Towards Efficient Cryptographic Data Validation Service in Edge Computing "** being submitted by **Erragunta Harsha Reddy** bearing hall ticket number **20EG105312, Golla Santhosh Kumar** bearing hall ticket number **20EG105315, Reddaveni Sai Krishna** bearing hall ticket number **20EG105330** in partial fulfilment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** in **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from Academic year 2023 to 2024**.**

The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report has not been submitted to any other University for the award of any other degree or diploma.

Internal Guide
Mrs. P. Aparna
Assistant Professor, CSE

Signature of the Dean
Dr. G. Vishnu Murthy
Dean, CSE

External Examiner

DECLARATION

We hereby declare that the project work entitled **"Towards Efficient Cryptographic Data Validation Service in Edge Computing"** submitted to the **Anurag University** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mrs. P. Aparna, Assistant Professor** and this project work has not been submitted to any other university for the award of any other degree or diploma.

Place: Anurag University, Hyderabad

Date:20/04/2024

ERRAGUNTA HARSHA REDDY

20EG105312

GOLLA SANTHOSH KUMAR

20EG105315

REDDAVENI SAI KRISHNA

20EG105330

ACKNOWLEDGEMENT

It is our privilege and pleasure to express profound sense of respect, gratitude and indebtedness to our guide **Mrs. P. Aparna, Assistant Professor**, **Department of Computer Science and Engineering**, Anurag University for her indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

We extend our sincere thanks to **Dr. V. Vijaya Kumar**, **Dean School of Engineering**, Anurag University, for his encouragement and constant help.

We would like acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, **Dean, Department of Computer Science and Engineering**, Anurag University.

We also express my deep sense of gratitude to **Dr. V.V.S.S.S. Balaram**, Academic coordinator. **Dr. T Shyam Prasad** Project Co-Ordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage our project work.

Erragunta Harsha Reddy

20EG105312

Golla Santhosh Kumar

20EG105315

Reddaveni Sai Krishna

20EG105330

ABSTRACT

Title: “TOWARDS EFFICIENT CRYPTOGRAPHIC DATA VALIDATION SERVICE IN EDGE COMPUTING”

Edge computing brings data computation and storage closer to the mobile device to save response time for decision making. After being processed at the edge, commonly the data will be uploaded to the cloud for further enriched analysis. For privacy concerns, local devices may encrypt the collected data before sending it to the cloud server. However, this treatment increases the server's processing time and makes it hard to pick out the desired data. In this article, to address this problem, we design an encrypted data validation scheme, which enables the edge to clean the encrypted data to be uploaded. Because edge computing encompasses numerous edge devices from different service providers, we explore the public-key cryptographic mechanism to implement our secure data validation scheme. Considering potential risks from quantum computers, we propose to leverage ideal lattice to realize our protocol, which reaches better performance in both time and storage for edge devices. Extensive evaluation results show that our proposed proposal achieves considerable performance improvements in terms of communication and computation aspects. Particularly, compared to prior work, nearly 123×-238× speed up is achieved in our key derivation procedure and the storage cost of the secret key is reduced from 547MB to 1.7MB.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1.INTRODUCTION

1.1. Overview

In the realm of edge computing, where data processing occurs closer to the source for faster decision-making, the conventional approach of uploading processed data to the cloud after encryption presents a trade-off between security and efficiency. To address this challenge, a novel encrypted data validation scheme is proposed in this article. This scheme empowers edge servers to clean encrypted data before uploading it to the cloud, enhancing both data security and processing efficiency. Leveraging public-key cryptographic mechanisms and ideal lattice structures to fortify against potential quantum computing risks, the proposed protocol offers significant performance enhancements in terms of communication and computation. Notably, key derivation procedures see a remarkable speedup of nearly 123×238, while the storage cost of secret keys is drastically reduced from 547MB to 1.7MB, ushering in a new era of secure and efficient edge computing paradigms.

The proposed method revolutionizes data management at the edge by providing enhanced security without compromising processing speed. Unlike traditional approaches, where cloud servers handle data storage and processing, the proposed scheme utilizes edge servers to process encrypted data, thereby mitigating security risks associated with unencrypted processing. By granting access to data for the current day only and autonomously deleting it after use, the edge server ensures data privacy and minimizes exposure to potential breaches. This innovative approach not only saves time by eliminating the need for cloud server involvement in daily data access but also establishes a robust foundation for secure edge computing environments, poised to meet the evolving demands of modern data-driven applications.

1.2. Problem Statement

In existing methods data is managed by cloud servers to store and handle request for user data where edge server is used to process data. To reduce time of data processing edge servers are used to process data without encryption which is not secure even time of data process is decreased security is compromised.

1.3. Problem Illustration

In proposed method we are using edge computing to improve data security and save time to access data. In proposed system when owner uploads data to edge server , sever will handle one day data with encrypted data and give access to users with out involvement of cloud servers for respective day and delete data after access for that day from edge server, Edge server only gives access for that day data where are cloud sever provides access for old data. This process saves time and improves security by encrypting data.

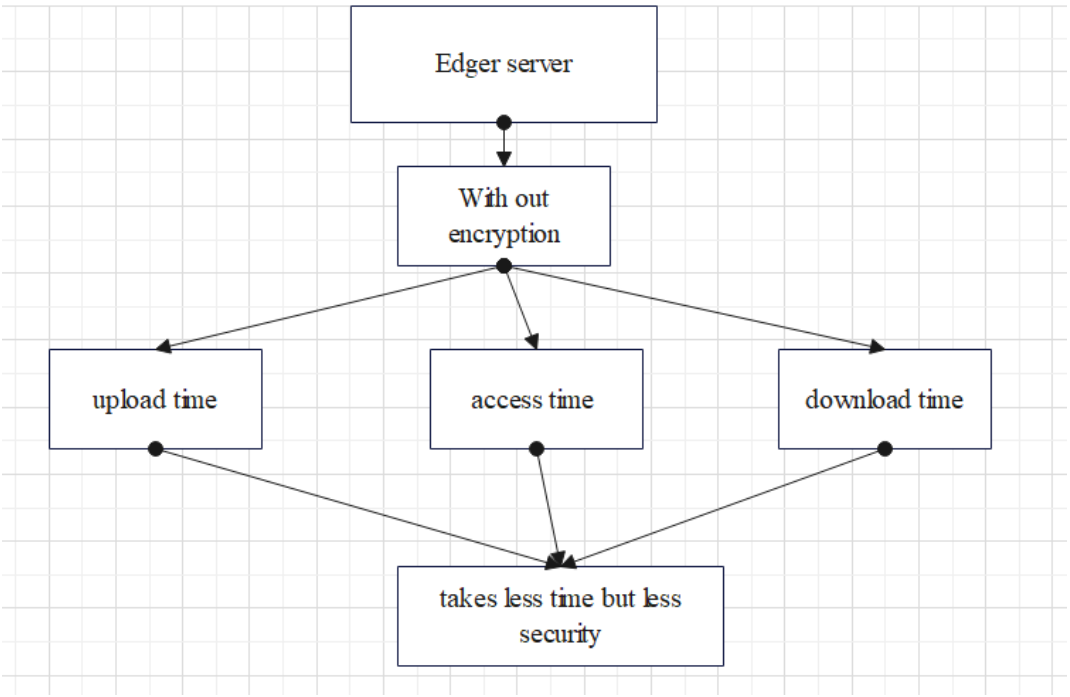


Figure 1.3. Process of Edge computing

2. LITERATURE REVIEW

The literature on mobile edge computing (MEC) spans various aspects, each contributing to our understanding of this evolving field. Mao et al. (2017) offer a comprehensive survey from a communication perspective, outlining MEC architectures, technologies, and communication protocols. Their work provides a foundational understanding of MEC's communication infrastructure, essential for optimizing resource allocation and improving network efficiency. Complementing this, Xiong et al. (2020) delve into the intersection of cloud/edge computing and blockchain networks, proposing novel management strategies using game theory-based approaches. By integrating blockchain technology with cloud/edge services, their work addresses emerging challenges in decentralized computing environments, highlighting the importance of innovative management techniques.

Security remains a critical concern in MEC systems, particularly in sensitive domains like autonomous driving. Ren et al. (2020) examine the security landscape of autonomous driving systems, identifying threats and proposing defense mechanisms to safeguard against cyber-attacks. This research underscores the need for robust security measures to ensure the safety and integrity of autonomous vehicles and their surrounding infrastructure. Additionally, the exponential growth of connected IoT devices, as projected by IDC, emphasizes the urgency for scalable and secure data management solutions (IDC Forecast, 2019). This forecast serves as a call to action for developing efficient data processing and storage mechanisms capable of handling the anticipated data deluge.

Within the realm of secure data management, cryptographic techniques play a pivotal role. Wang et al. (2011) explore methods for effectively utilizing encrypted cloud data while preserving privacy, addressing the inherent trade-off between data security and accessibility. Similarly, Boneh et al. (2004) introduce the concept of public key encryption with keyword search (PEKS), enabling encrypted data search without revealing sensitive information. These cryptographic primitives, alongside advancements in lattice-based cryptography (Regev, 2009; Agrawal et al., 2010; Lyubashevsky, 2012), offer promising avenues for enhancing data security and privacy

in cloud/edge computing environments. Collectively, these works form a rich tapestry of research contributions, shedding light on the multifaceted challenges and opportunities in the domain of mobile edge computing.

2.1. Comparison of Existing Methods :

Sl.No	Author	Strategies	Advantages	Disadvantages
1	Z. Xiong, J. Kang, D. Niyato, P. Wang and H. V. Poor	Alternating Direction Method of Multipliers (ADMM) algorithm	The cloud/edge providers and miners in blockchain using a multi-leader multi-follower game-theoretic approac	which is resource expensive to implement in lightweight devices
2	C. Wang, Q. Wang and K. Re	Fuzzy keyword search aims at accommodating	enabling an encrypted cloud data search service with privacy-assurance is of paramount importance	large capital outlays in the purchase and management of both software and hardware
3	D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano	Public Key Encryption with keyword Search	Public Key Encryption with keyword Search over encrypted data	This method deals with only security access not with time saving
4	O. Regev	SIVP	the RSA function to enable each authority to limit the search .	Access based control is discussed but not edge server related solutions
5	S. Agrawal, D. Boneh and X. Boyen	IBE and a Hierarchical IBE.	solution to the learning problem implies a quantum algorithm for GAPSVP and SIVP	reduction from worst-case lattice problems

Table 2.1 Comparison of Existing Methods

3. PROPOSED METHODS

3.1 Illustration

In the ever-evolving landscape of technology, the feasibility of a proposed system stands as a pivotal determinant of its potential success and impact. With the burgeoning interest and adoption of edge computing paradigms, the need to assess the feasibility of innovative solutions becomes paramount. This introduction sets the stage for evaluating the viability of a proposed encrypted data validation scheme in the context of edge computing.

Edge computing, characterized by decentralized data processing closer to the data source, offers unprecedented opportunities for enhancing efficiency and reducing latency in decision-making processes. However, this shift towards edge-centric architectures necessitates careful consideration of security and privacy implications, particularly concerning the handling of sensitive data. The proposed encrypted data validation scheme emerges as a promising solution to reconcile these competing demands, promising enhanced security while maintaining processing efficiency.

This introduction will delve into the key components of system feasibility, including technical, operational, economic, and schedule feasibility. Through a comprehensive assessment of these factors, we aim to ascertain the practicality and viability of implementing the proposed scheme within the context of edge computing environments. By elucidating the potential benefits, challenges, and considerations associated with the proposed system, this analysis seeks to inform stakeholders and decision-makers about its feasibility and suitability for real-world deployment.

3.2. Existing System :

In existing methods data is managed by cloud servers to store and handle request for user data where edge server is used to process data. To reduce time of data processing edge servers are used to process data without encryption which is not secure even time

of data process is decreased security is compromised. But limitations are there in this method as follows

Security Vulnerabilities: Data processing at the edge without encryption exposes sensitive information to security vulnerabilities, increasing the risk of data breaches or unauthorized access.

Limited Data Privacy: Lack of encryption in the existing system compromises data privacy, potentially leading to breaches of confidentiality and integrity.

Dependence on Cloud Servers: Reliance on cloud servers for data access may introduce latency and dependency issues, impacting overall system performance and responsiveness.

3.3. Proposed System :

In proposed method we are using edge computing to improve data security and save time to access data. In proposed system when owner uploads data to edge server , sever will handle one day data with encrypted data and give access to users with out involvement of cloud servers for respective day and delete data after access for that day from edge server, Edge server only gives access for that day data where are cloud sever provides access for old data. This process saves time and improves security by encrypting data.

Enhanced Data Security: By encrypting data at the edge before processing, the proposed system significantly enhances data security, reducing the risk of unauthorized access or breaches.

Improved Privacy: Automatic deletion of data after daily access ensures user privacy and minimizes exposure to potential security threats.

Optimized Access Time: Granting access to data at the edge without involving cloud servers streamlines the access process, resulting in reduced latency and improved overall efficiency.

3.4. Algorithm :

Skey:

The setup algorithm will choose a bilinear group G0 of prime order p with generator g. Next it will choose two random exponents α,β ∈ Zp. The public key is published as:

PK = G0,g,h = g β ,f = g 1/β,e(g,g) α and the master key MK is (β,gα). (Note that f is used only for delegation.

Key Generation:

The key generation algorithm will take as input a set of attributes S and output a key that identifies with that set. The algorithm first chooses a random r ∈ Zp, and then random rj ∈ Zp for each attribute j ∈ S. Then it computes the key as

SK = D = g (α+r)/β , ∀j ∈ S : Dj = g r · H(j) rj ,D′ j = g rj

Edge server:

Stores data for respective day and give access to users and deletes old data and give access to cloud to give permission for old data.

Encryption:

This is a randomized algorithm that takes as input a message m, a set of attributes γ, and the public parameters PK. It outputs the ciphertext E.

Decryption with Key:

The advantage of an adversary A in this access control is defined as Pr[b ′ = b] − 1 2 . We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries

4.DESIGN

4.1. System Design :

1. To conduct studies and analyses of an operational and technological nature, and
2. To promote the exchange and development of methods and tools for operational analysis as applied to defense problems.

Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams

Physical design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output. In Physical design, following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing Requirements,
5. System control and backup or recovery.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

4.2. UML Diagrams

UML, or Unified Modeling Language, is a graphical tool essential for designing software systems. It offers standardized visual models for representing object-oriented software structures. UML diagrams are crucial for clear and organized communication of design concepts. These diagrams are indispensable in software design, aiding developers in understanding and analyzing intricate systems. They serve as effective tools for conveying design ideas to team members, stakeholders, and clients, ensuring that the software meets required standards of functionality, performance, and quality. Moreover, UML diagrams help in detecting and rectifying errors early in the development process, thereby saving time and reducing costs. They come in two main types: structural diagrams, which depict the static structure of the system, and behavioral diagrams, which illustrate dynamic interactions and Workflows. In summary, UML diagrams play a vital role in streamlining the software development process, providing developers with a clear and efficient means of conceptualizing and refining software systems.

4.2.1 Use case diagrams:

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 4.2.1 Use Case Diagram

4.2.2 Sequence Diagrams:

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.

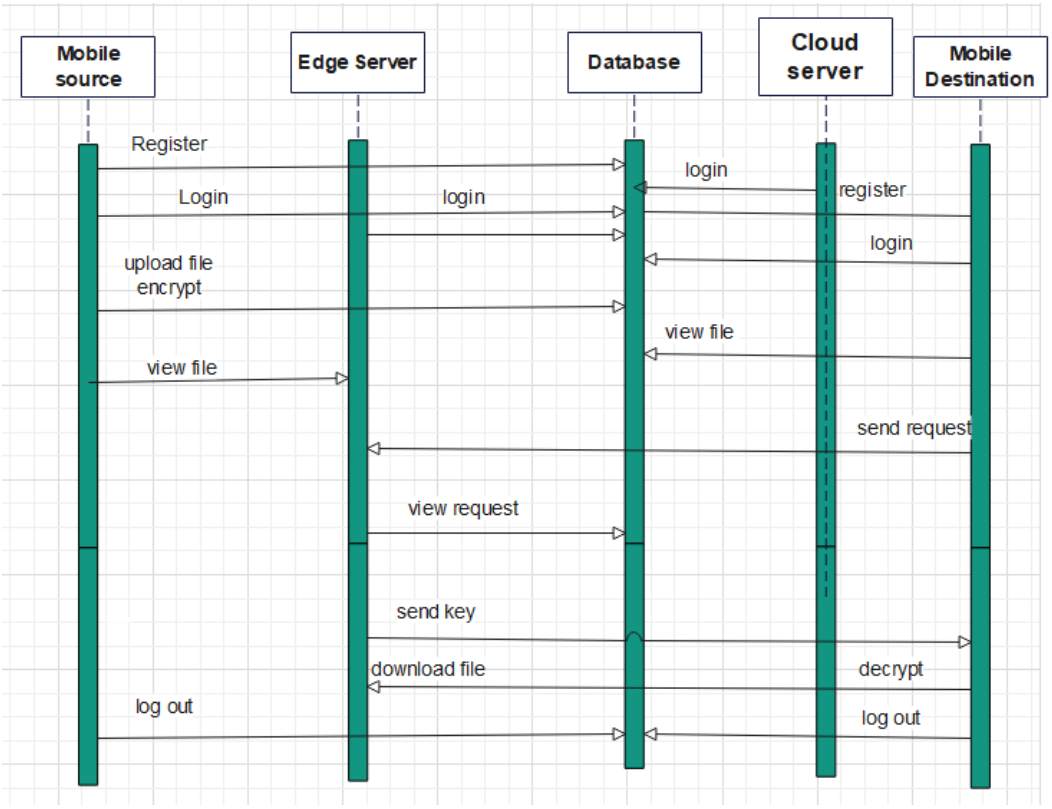


Figure 4.2.2: Sequence diagram

4.2.3. 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.

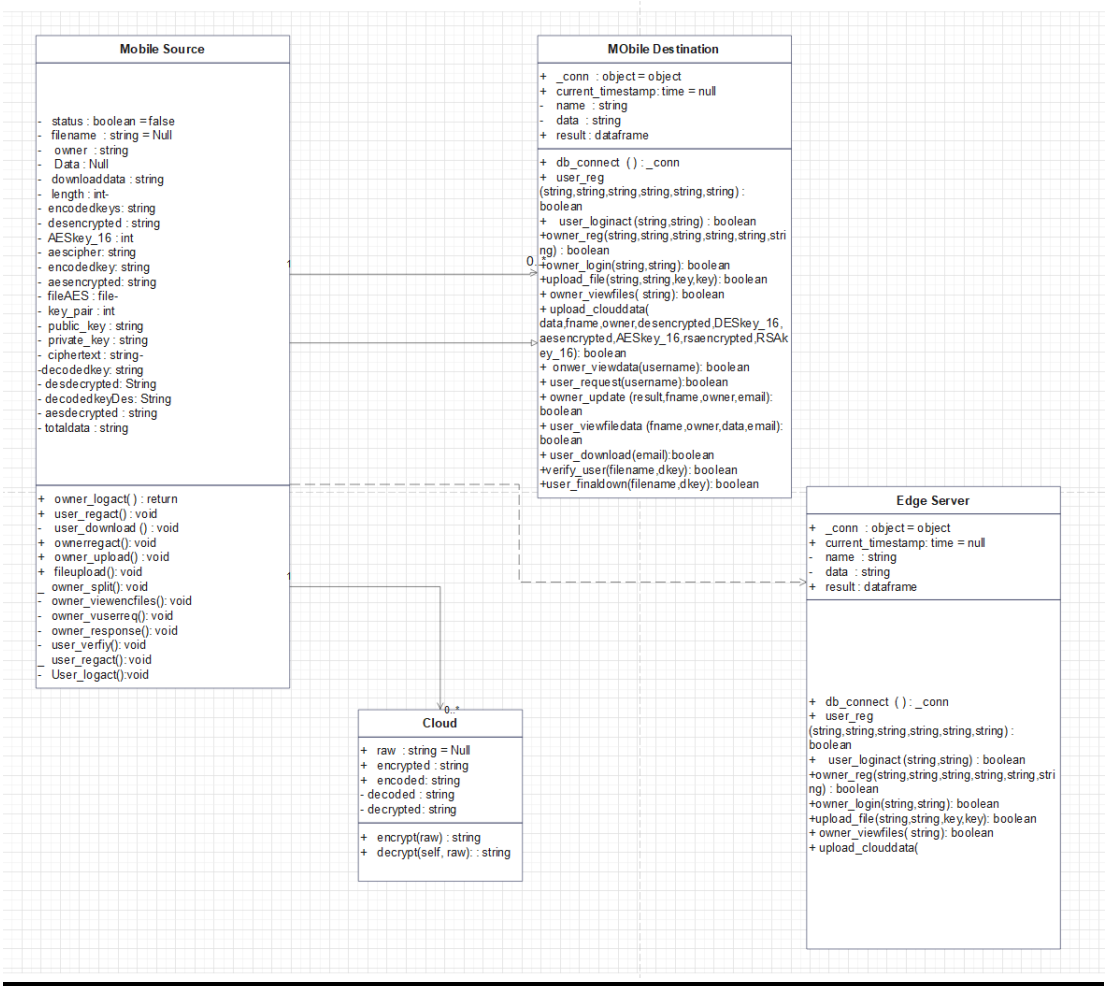


Figure 4.2.3: Class Diagram

4.2.4 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.

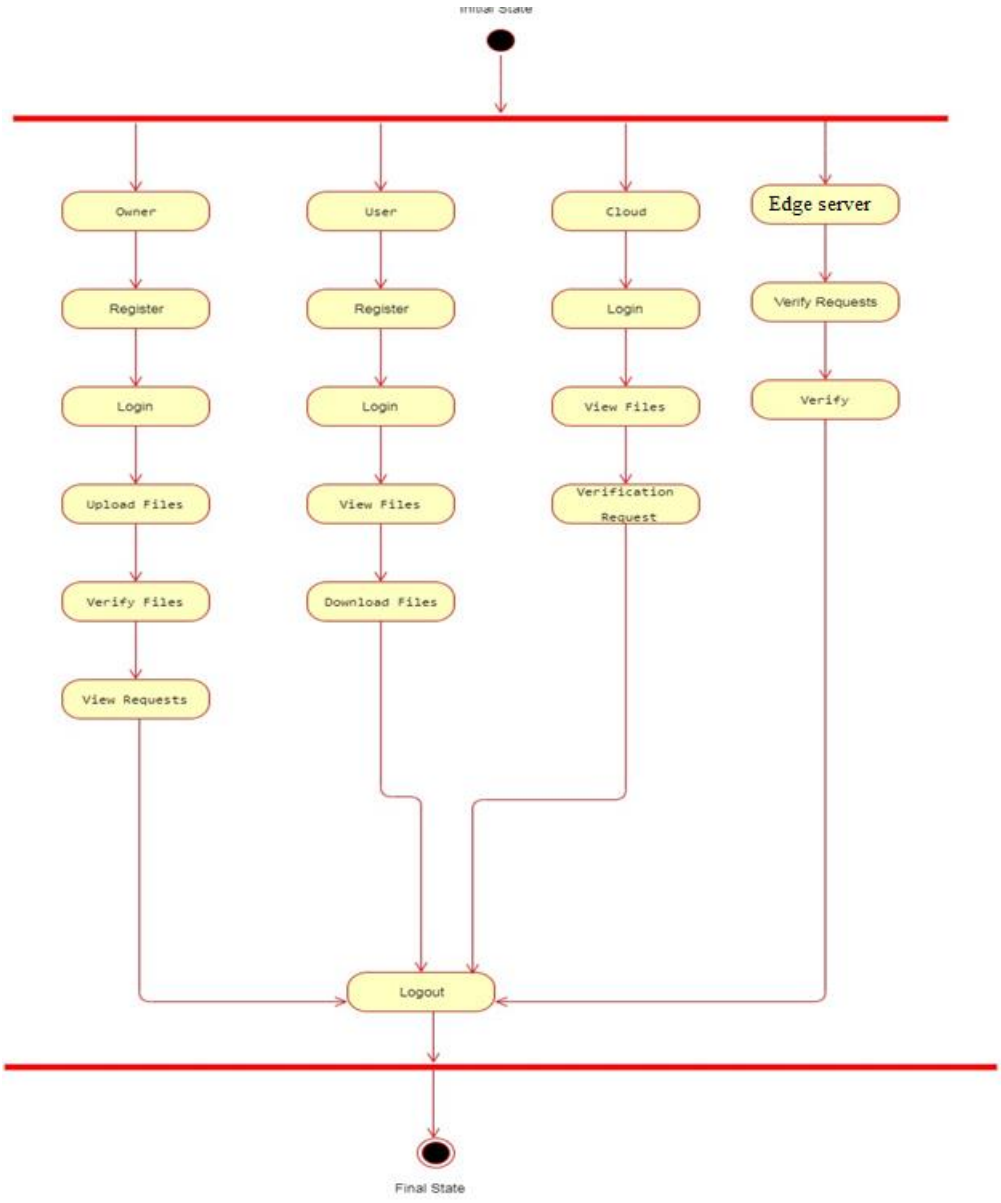


Figure 4.4.4: Activity Diagram

5. IMPLEMENTATION

5.1. Modules

Mobile User:

Using this module mobile user can register with application and login with valid username and password. Mobile user can upload file and secure data by encrypting data and send to cloud through edge server. Mobile user can view upload data and encrypted data and logout from application.

Edge Server :

Using this module edger server can view list of files uploaded by files which are stored for only one day. When user requests data of same day by mobile destination edge server will handle requests and send response to mobile destination. After one day catched data is deleted from the mobile and only cloud can view those files.

Cloud Server:

Using this module cloud will login and view files uploaded by every mobile user and view list of files and upload data to cloud and view uploaded files.

Mobile Destination:

Using this module mobile destination will register with application and login with valid username and password. MD can view upload files and send request to edge server to get today data uploaded by mobile user. After getting request from edger server key is sent to destination user and using this key MD can decrypt encrypted data and download.

5.2. Sample code :

This section presents our identity-based public-key data validation scheme. To support more advanced validation operations, we also show how to extend the above scheme to a range validation scheme in which the gateway can filter the encrypted in a predefined range. We give detailed parameter setting in this work and prove that our scheme is

correct under this setting. Besides, we perform computation and storage complexity analyses of the scheme theoretically and confirm that it could achieve the goal with minimum overhead compared to existing representative works.

5.2.1. System Setup and Secret Key Derivation :

In the system setup stage, the central authority first setups the system to produce the main public key and main secret key and then computes the secret key for each edge gateway deployed in the system. Above goals can be realized by using Setup and Derive functions in Algorithm 1. Concretely, for the given system parameters n; q; k, the central authority first runs the ideal lattice . Thus, the storage cost can be reduced from quadratic complexity to linear complexity.

Setup (1^λ) //Central authority

1: Sample $(a, \mathbf{S}_a) \leftarrow \texttt{IdealTrapGen}(n, k, q, f)$
2: Select $b \xleftarrow{\$} \mathcal{R}_q^\ell$, $h \xleftarrow{\$} \mathcal{R}_q^\ell$, $d \xleftarrow{\$} \mathcal{R}_q^\ell$, $e \xleftarrow{\$} \mathcal{R}_q^\ell$, $u \in \mathcal{R}_q$
3: $\mathrm{m}pk \leftarrow (a, b, h, d, e, u)$, $\mathrm{m}sk \leftarrow \mathbf{S}_a$
4: **return** $\mathrm{m}pk, \mathrm{m}sk$

Derive $(\mathrm{m}sk, \tau \in \mathcal{R}^\ell)$ //Central authority

1: Parse $[\tau_1, \ldots, \tau_\ell] \leftarrow \tau$, $[b_1, \ldots, b_\ell] \leftarrow b$
2: Parse $[h_1, \ldots, h_\ell] \leftarrow h$
3: **for** $i = 1$ to ℓ **do**
4: $\quad f_i \leftarrow b_i + h_i \otimes \tau_i$
5: **end for**
6: Let $f \leftarrow [f_1, \ldots, f_\ell]$, $a_\tau \leftarrow [a|f]$
7: $sk_\tau \leftarrow \texttt{SampleBasisLeft}(a, f, \mathbf{S}_a, \sigma)$
8: **return** sk_τ $\triangleright sk_\tau \in \mathcal{R}^{k+\ell}$ is the basis of $\Lambda^\perp(\mathrm{rot}(\mathbf{a}_\tau))$

5.2.2. Data Encryption

To upload the collected data securely, each edge device encrypts the data and transmits it to the correspondence edge gateway. Look at the Encrypt function in Algorithm 2, for a generated keyword w, the edge device first encodes the keyword with the edge gateway's identity and if it is, it indicates that c contains the keyword w, otherwise not. Observe that, such a validation modular can directly decide if the ciphertext contains a specific keyword without leaking its underlying content.

Encrypt $(mpk, \tau \in \mathcal{R}^{\ell}, w \in \mathcal{R}^{\ell})$ //Edge devices

Parse $[\tau_1, \ldots, \tau_\ell] \leftarrow \tau$, $[w_1, \ldots, w_\ell] \leftarrow w$
Parse $[b_1, \ldots, b_\ell] \leftarrow b$, $[h_1, \ldots, h_\ell] \leftarrow h$
for $i = 1$ to ℓ **do**
  $f_i \leftarrow b_i + h_i \otimes \tau_i$
end for
Parse $[d_1, \ldots, d_\ell] \leftarrow d$, $[e_1, \ldots, e_\ell] \leftarrow e$
for $i = 1$ to ℓ **do**
  $g_i \leftarrow d_i + e_i \otimes w_i$
end for
Let $f \leftarrow [f_1, \ldots, f_\ell]$, $g \leftarrow [g_1, \ldots, g_\ell]$
Set $a_{\tau,w} \leftarrow [a|f|g]$
$s \leftarrow \mathcal{R}_q$, $x \xleftarrow{\$} \Psi^{n}_{\alpha q}$, $y \xleftarrow{\$} \Psi^{kn}_{\alpha'}$, $z \xleftarrow{\$} \Psi^{2\ell n}_{\alpha'}$
$c_1 \leftarrow s \otimes u + x$, $c_2 \leftarrow s \cdot a_{\tau,w} + [y|z]$
$c \leftarrow (c_1, c_2)$
Store c on the server.

5.2.3. Secure Data Validation

To ease the pressure brought by limited resources, the edge gateway performs data validation to tick out unnecessary items before storing them in the cloud. Since the data is encrypted for confidentiality purposes, one needs to complete this procedure under the encrypted status. To this end, we exploit the token in the public key data validation scheme to help check whether the uploaded encrypted data contains the undesired keyword. The whole procedure includes a token generation algorithm TokGen and a test (i.e., Filter in Definition 1) algorithm CipherFilter run by the edge gateway.

TokGen (sk_τ, w)// Edge gateway

Parse τ, w, b, h, d, e as before
for $i = 1$ to ℓ **do**
  $f_i \leftarrow b_i + h_i \otimes \tau_i$, $g_i \leftarrow d_i + e_i \otimes w_i$
end for
Let $f \leftarrow [f_1, \ldots, f_\ell]$, $g \leftarrow [g_1, \ldots, g_\ell]$
Set $a_\tau \leftarrow [a|f]$
$st_{\tau,w} \leftarrow \texttt{SampleLeft}(a_\tau, g, sk_\tau, u, \sigma)$ ▷ filter token
Send $st_{\tau,w}$ to the server.

CipherFilter (c, st_w) // Server

Parse the ciphertext $(c_1, c_2) \leftarrow c$
$\gamma \leftarrow |\phi(c_1 - st_w^{\top} c_2)|$
if $\gamma \leq \lfloor \frac{q}{4} \rfloor$ **then** ▷ here hold on component-wise
  return 1 ▷ the ciphertext matches the token
else
  return 0
end if

5.3. CODE

Databaseconnection

```
package databaseconnection;
import java.sql.*;

public class databasecon
{
      static Connection co;
      public static Connection getconnection()
      {


            try
            {
                  Class.forName("com.mysql.jdbc.Driver");
                  co =
DriverManager.getConnection("jdbc:mysql://localhost:3306/securedata","
root","root");
            }
            catch(Exception e)
            {
                  System.out.println("Database Error"+e);
            }


            return co;
      }

}
```

Decryption

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package action;

/**
 *
 * @author java2
 */
import com.sun.org.apache.xerces.internal.impl.dv.util.Base64;
import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.util.Scanner;
```

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.swing.JOptionPane;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

public class decryption {
//public static void main(String args[])
//{
//            Scanner s=new Scanner(System.in);
//             System.out.println("Enter encrypted Text and key");
//             String text=s.next();
//             String key=s.next();
//    new decryption().decrypt(text,key);
//}


   public String decrypt(String txt, String skey) {
       String decryptedtext = null;
       try {

           //converting string to secretkey
           byte[] bs = Base64.decode(skey);
           SecretKey sec = new SecretKeySpec(bs, "AES");
           System.out.println("converted string to seretkey:" + sec);

           System.out.println("secret key:" + sec);

           Cipher aesCipher = Cipher.getInstance("AES");//getting AES
instance
           aesCipher.init(Cipher.ENCRYPT_MODE, sec);//initiating
ciper encryption using secretkey

           byte[] byteCipherText = new
BASE64Decoder().decodeBuffer(txt); //encrypting data

           //  System.out.println("ciper text:"+byteCipherText);
           aesCipher.init(Cipher.DECRYPT_MODE, sec,
aesCipher.getParameters());//initiating ciper decryption

           byte[] byteDecryptedText =
aesCipher.doFinal(byteCipherText);
           decryptedtext = new String(byteDecryptedText);

           System.out.println("Decrypted Text:" + decryptedtext);
       } catch (Exception e) {
           System.out.println(e);
       }
       return decryptedtext;
   }

    String decrypt(String str, SecretKey sec) {
        throw new UnsupportedOperationException("Not supported yet.");
 //To change body of generated methods, choose Tools | Templates.
    }

 }
```

Download

```
<%@page import="java.sql.*"%>
<%@page import="Distributed.Dbconnection"%>
<%@ page session="true" %>
    <%
    String username = session.getAttribute("user").toString();
    String task = request.getParameter("task");
    String location = request.getParameter("location");
    String status = null;
    String person = null;


    try{


    Connection con=Dbconnection.getConnection();
    PreparedStatement ps=con.prepareStatement("insert into task
values(?,?,?,?,?)");

    ps.setString(1,username);
    ps.setString(2,task);
    ps.setString(3,location);

    ps.setString(4,person);
    ps.setString(5,status);


   |

    int i=ps.executeUpdate();
    if(i>0)
    {
    response.sendRedirect("owner_upload.jsp?msg=Registered");
    }
    else{
    response.sendRedirect("ownerreg.jsp?msg1=Failed");
    }
    %>
    <%
    }

    catch(Exception e)
    {

       out.println(e);

    }
```

5.4 System Requirements :

1.	Operating system	:	Windows XP/7/10.
2.	System	:	Pentium IV 2.4 GHz and above
3.	Ram	:	2GB
4.	Hard Disk	:	200GB
5.	Coding Language	:	Java
6.	Tool	:	Netbeans
7.	Database	:	MYSQL

6.EXPERMENT RESULTS

6.1. Experiment Screen Shots

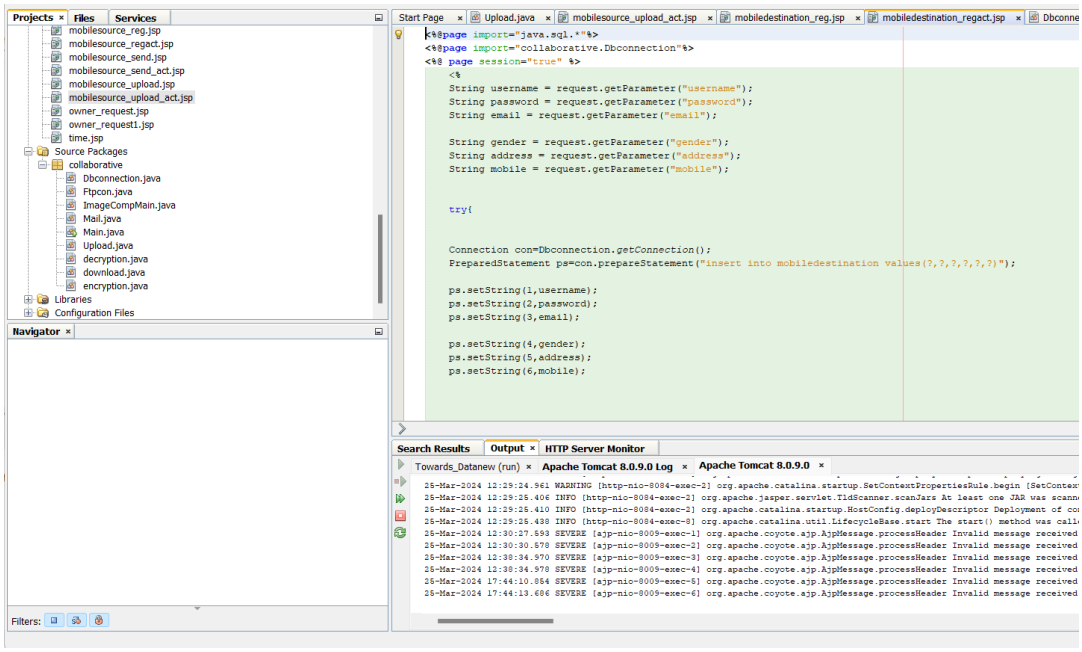


Figure 6.1. NetBeans IDE Environment

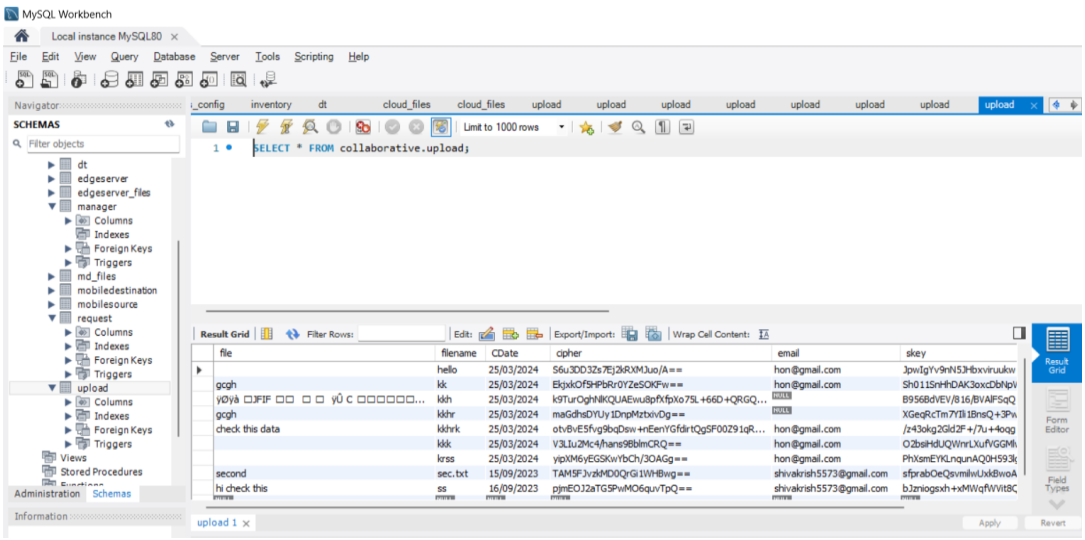


Figure 6.2. MySQL Workbench

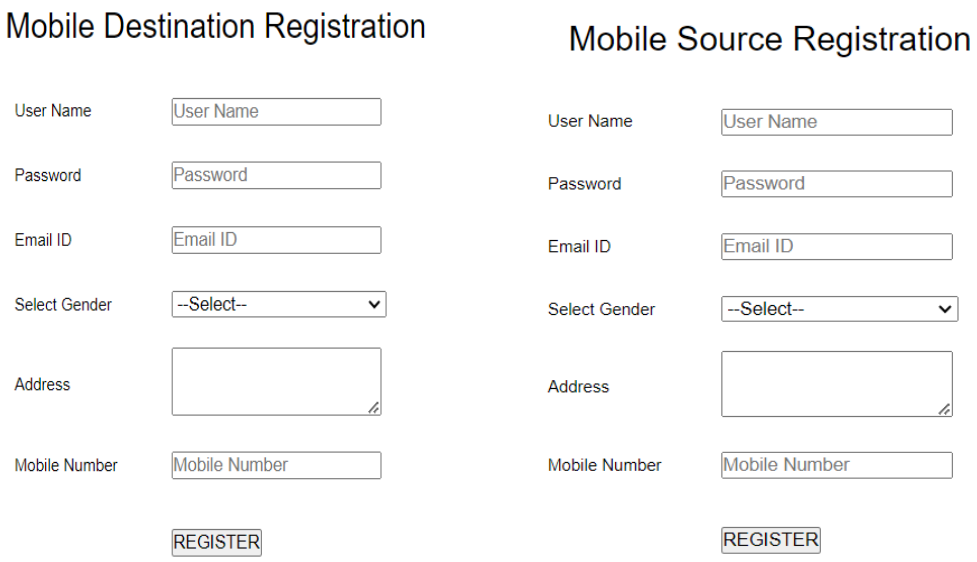


Figure 6.3. Registration Process

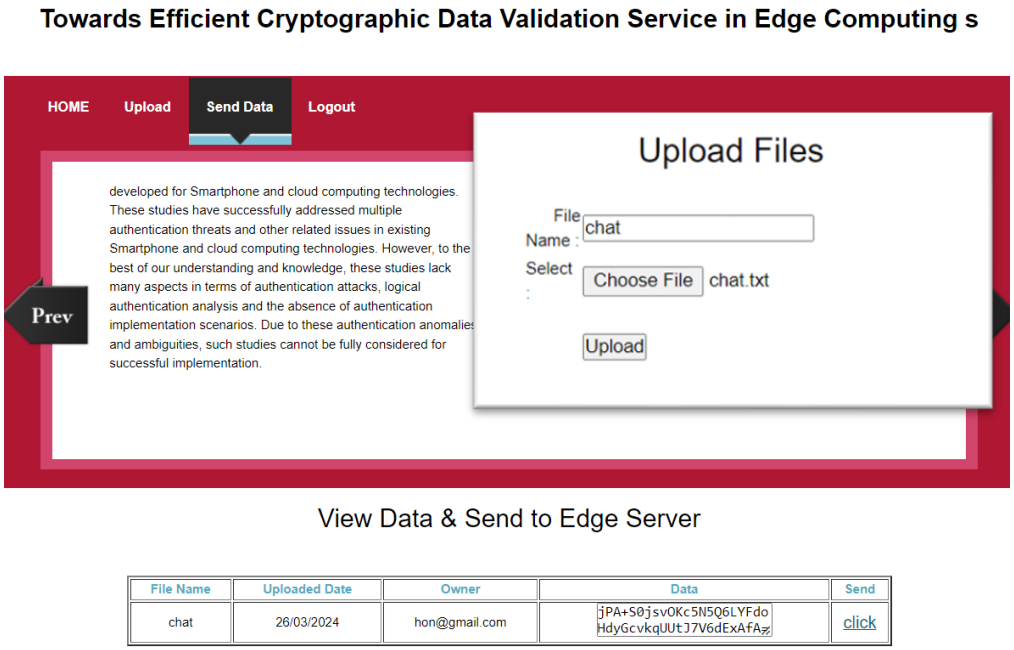


Figure 6.4. Uploading from Source

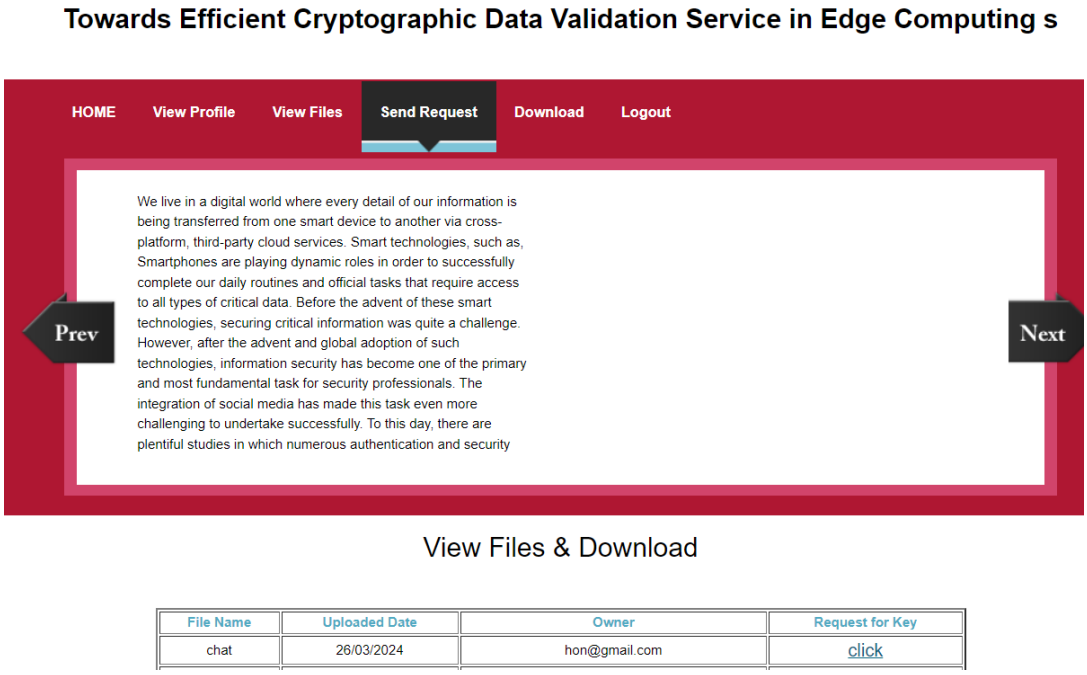


Figure 6.5. Mobile Destination Request of Data

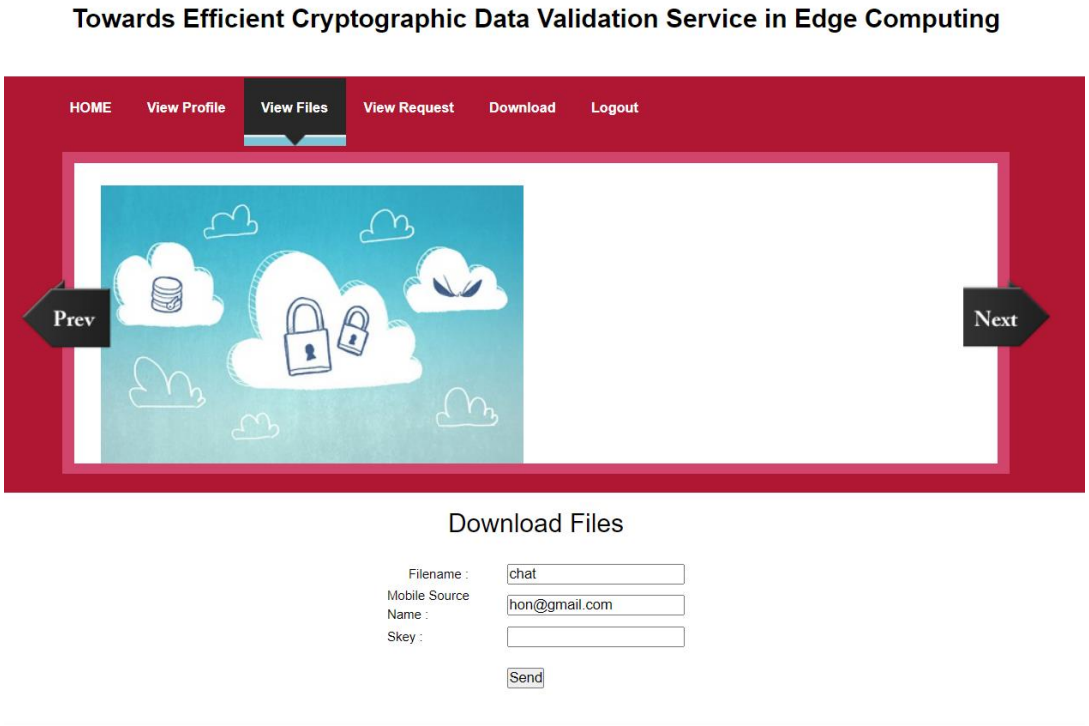
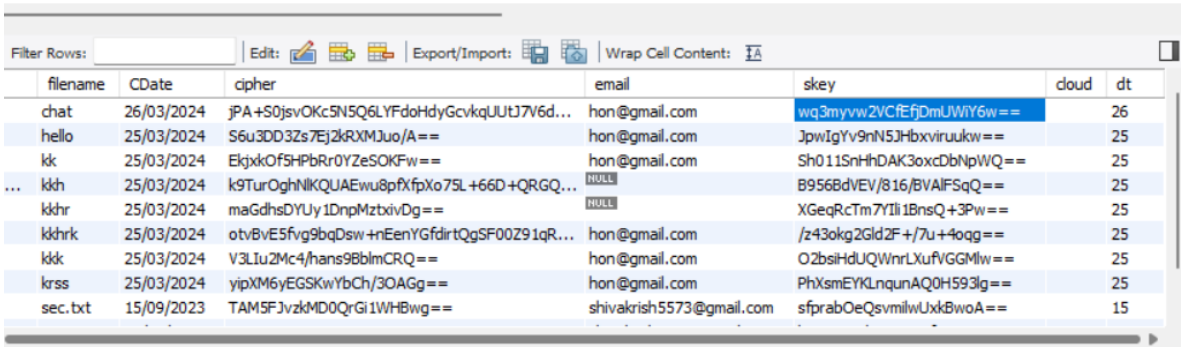


Figure 6.6. Download of Files using Skey



Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

filename	CDate	cipher	email	skey	cloud	dt
chat	26/03/2024	jPA+S0jsvOKc5N5Q6LYFdoHdyGcvkqUUtJ7V6d...	hon@gmail.com	wq3myvw2VCfEfjDmUWiY6w==		26
hello	25/03/2024	S6u3DD3Zs7Ej2kRXMJuo/A==	hon@gmail.com	JpwIgYv9nN5JHbxviruukw==		25
kk	25/03/2024	EkjxkOf5HPbRr0YZeSOKFw==	hon@gmail.com	Sh011SnHhDAK3oxcDbNpWQ==		25
kkh	25/03/2024	k9TurOghNlKQUAEwu8pfXfpXo75L+66D+QRGQ...	NULL	B956BdVEV/816/BVAlFSqQ==		25
kkhr	25/03/2024	maGdhsDYUy1DnpMztxivDg==	NULL	XGeqRcTm7YIi1BnsQ+3Pw==		25
kkhrk	25/03/2024	otvBvE5fvg9bqDsw+nEenYGfdirtQgSF00Z91qR...	hon@gmail.com	/z43okg2Gld2F+/7u+4oqg==		25
kkk	25/03/2024	V3LIu2Mc4/hans9BblmCRQ==	hon@gmail.com	O2bsiHdUQWnrLXufVGGMlw==		25
krss	25/03/2024	yipXM6yEGSKwYbCh/3OAGg==	hon@gmail.com	PhXsmEYKLnqunAQ0H593lg==		25
sec.txt	15/09/2023	TAM5FJvzkMD0QrGi1WHBwg==	shivakrish5573@gmail.com	sfprabOeQsvmilwUxkBwoA==		15

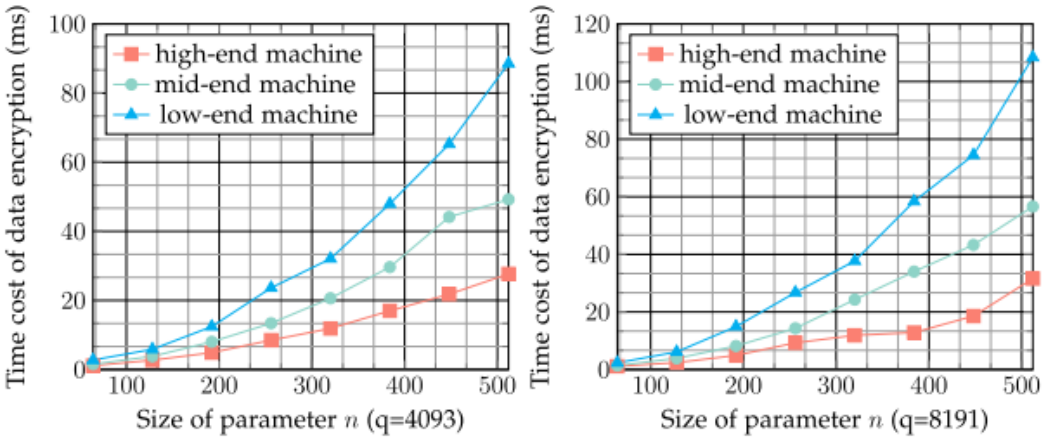
Figure 6.7. Skey in MySQL Workbench

7. DISCUSSION OF RESULTS

7.1. Time Cost of Data Encryption

The encryption algorithm consists of several polynomial multiplication and addition operations over ring Ro as well as some Gaussian sample algorithms. We utilize the classic Euclidean algorithm and convolution function to implement this algorithm. We draw the time cost of encrypting a random keyword by one random selected identity in Fig. 7.1 .

Performance on Machines With Different Computing Capabilities. Considering the heterogeneous computation performance of different edge devices (computing node), we conduct more experiments on machines with different computing capabilities to evaluate the efficiency of the proposed scheme, particularly for the encryption algorithm which is implemented by the edge devices. Specifically, we run the encryption algorithm on three selected laptops with the Intel Core i9, 2.4GHz processor and 64GB memory running (high-end), Intel Core i7, 1.8GHz processor and 16GB running (mid-end), Intel Core i5, 2.2GHz processor and 4GB memory running (low-end), respectively. The experimental results are reported



(a) Time cost of data encryption (b) Time cost of data encryption

Figure 7.1. Computing capabilities on Encryption

7.2. Comparison With Prior Work

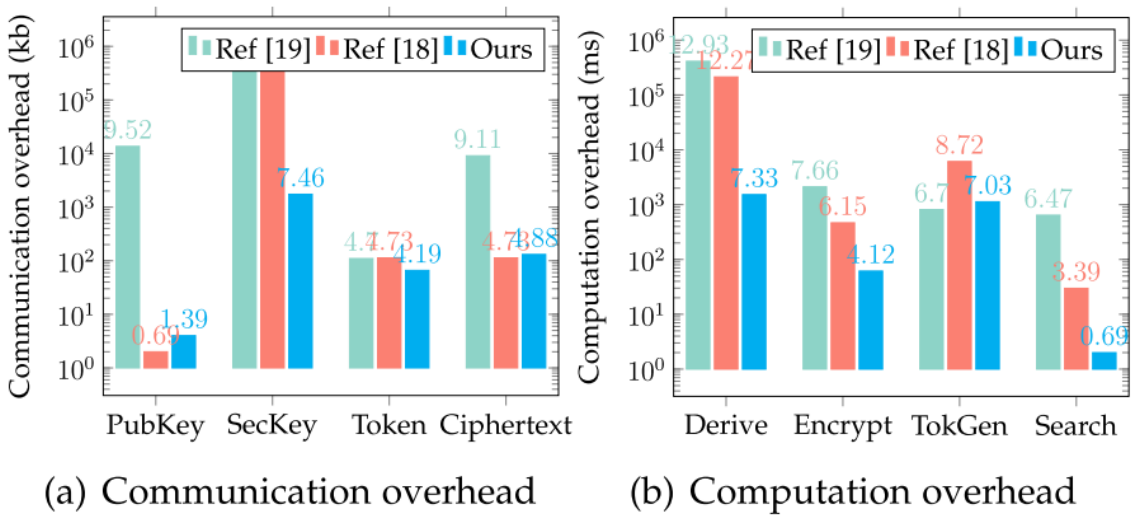


Figure 7.2. Computation with prior work

We compare our scheme with some representative lattice work in this subsection to validate the advantage of our scheme. Both of the selected works here are lattice based constructions and one of them is an identity-based searchable encryption scheme. Considering the fairness, we select the experiment parameters referring to the same security level. In terms of communication performance, we calculate the size of the pubic key, secret key, token and ciphertext of selected schemes. Here, the public key refers to the encrypted key of edge gateways rather than the main public key. From Fig. 7.2. The filtering costs increase as the ring multiplication operation is involved, which requires more computation cost than the inner product.

.

8. CONCLUSION

In conclusion, the proposed encrypted data validation scheme represents a significant advancement in the realm of edge computing, offering a comprehensive solution to reconcile the trade-off between data security and processing efficiency. Through the utilization of edge computing infrastructure and advanced cryptographic techniques, the proposed system addresses the shortcomings of existing approaches by enhancing data security, preserving privacy, and optimizing access time. By leveraging edge servers to handle encrypted data and granting access only to daily data without involving cloud servers, the proposed scheme ensures robust security measures while minimizing latency in data access. The automatic deletion of data after daily access further enhances privacy protection, mitigating the risk of unauthorized access or breaches.

The feasibility of the proposed system is underscored by its potential to revolutionize data management practices at the edge, paving the way for secure and efficient edge computing environments. The significant performance enhancements achieved, including notable speedups in key derivation procedures and substantial reductions in storage costs, demonstrate the practicality and viability of the proposed scheme. Overall, the proposed encrypted data validation scheme embodies a paradigm shift in edge computing paradigms, offering a holistic approach to data security and privacy while optimizing processing efficiency. As edge computing continues to evolve and proliferate, the proposed system stands poised to meet the growing demands of modern data-driven applications, ushering in a new era of secure and efficient edge computing environments.

9. REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A survey on mobile edge computing: The communication perspective", IEEE Commun. Surv. Tutorials, vol. 19, no. 4, pp. 2322-2358, Oct–Dec. 2017.

[2] Z. Xiong, J. Kang, D. Niyato, P. Wang and H. V. Poor, "Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based ADMM for pricing", IEEE Trans. Serv. Comput., vol. 13, no. 2, pp. 356-367, Mar./Apr. 2020.

[3] K. Ren, Q. Wang, C. Wang, Z. Qin and X. Lin, "The security of autonomous driving: Threats defenses and future directions", Proc. IEEE, vol. 108, no. 2, pp. 357-372, Feb. 2020.

[4] "The growth in connected IoT devices is expected to generate 79.4zb of data in 2025 according to a new IDC forecast", 2019, [online] Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219.

[5] C. Wang, Q. Wang and K. Ren, "Towards secure and effective utilization over encrypted cloud data", Proc. IEEE 31st Int. Conf. Distrib. Comput. Syst. Workshops, pp. 282-286, 2011.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search", Proc. Int. Conf. Theory Appl. Cryptographic Techn., pp. 506-522, 2004.

[7] L. Xu, S. Sun, X. Yuan, J. K. Liu, C. Zuo and C. Xu, "Enabling authorized encrypted search for multi-authority medical databases", IEEE Trans. Emerg. Top. Comput., vol. 9, no. 1, pp. 534-546, Jan.–Mar. 2021.

[8] O. Regev, "On lattices learning with errors random linear codes and cryptography", J. ACM, vol. 56, no. 6, pp. 34:1-34:40, 2009.

[9] S. Agrawal, D. Boneh and X. Boyen, "Efficient lattice (H)IBE in the standard model", Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn., pp. 553-572, 2010.

[10] V. Lyubashevsky, "Lattice signatures without trapdoors", Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn., pp. 738-755, 2012.

[11] D. Micciancio and O. Regev, "Lattice-based cryptography", Lecture Notes iComput. Sci., vol. 4117, pp. 1-2, 2013.

[12] M. R. Albrecht, R. Player and S. Scott, "On the concrete hardness of learning with errors", J. Math. Cryptol., vol. 9, no. 3, pp. 169-203, 2015.

[13] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring", Proc. IEEE Annu. Symp. Found. Comput. Syst., pp. 124-134, 1994.

[14] L. Chen et al., Report on Post-Quantum Cryptography, 2016.

[15] T. Monz et al., "Realization of a scalable shor algorithm", Science, vol. 351, no. 6277, pp. 1068-1070, 2016.

[16] X. Zhang, C. Xu, H. Wang, Y. Zhang and S. Wang, "Fs-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things", IEEE Trans. Dependable Secure Comput., vol. 18, no. 3, pp. 1019-1032, May/Jun. 2021.

[17] Y. Zhang, C. Xu, J. Ni, H. Li and X. S. Shen, "Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage", IEEE Trans. Cloud Comput..

[18] L. Xu, X. Yuan, R. Steinfeld, C. Wang and C. Xu, "Multi-writer searchable encryption: An lwe-based realization and implementation", Proc. ACM Asia Conf. Comput. Commun. Secur., pp. 122-133, 2019.

[19] R. Behnia, M. O. Ozmen and A. A. Yavuz, "Lattice-based public key searchable encryption from experimental perspectives", IEEE Trans. Dependable Secure Comput., vol. 17, no. 6, pp. 1269-1282, Nov./Dec. 2020.

[20] M. Zeng, H. Qian, J. Chen and K. Zhang, "Forward secure public key encryption with keyword search for outsourced cloud storage", IEEE Trans. Cloud Comput..

[21] V. Lyubashevsky, C. Peikert and O. Regev, "On ideal lattices and learning with errors over rings", J. ACM, vol. 60, no. 6, pp. 43:1-43:35, 2013.

[22] M. Agiwal, A. Roy and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey", IEEE Commun. Surv. Tutorials, vol. 18, no. 3, pp. 1617-1655, Jul.–Sep. 2016.

[23] N. Developer, "Jetson nano developer kit", Jun. 2021, [online] Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.