A MAIN PROJECT REPORT ON

Design Secure and Efficient Biometric for Cloud Services

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

Computer Science and Technology

By

M. Anand Kumar (19A81A0644)

Under the Esteemed Supervision of

Mrs. R. Padmaja M. Tech.

Assistant Professor

Department of Computer Science and Technology (Accredited by N.B.A.)

SRI VASAVI ENGINEERING COLLEGE(Autonomous)

(Affiliated to JNTUK, Kakinada)

Pedatadepalli, Tadepalligudem-534101, A.P 2022-23

SRI VASAVI ENGINEERING COLLEGE (Autonomous)

Department Of Computer Science and Technology

Pedatadepalli, Tadepalligudem

Certificate

This is to certify that the Project Report entitled **"Design Secure and Efficient Biometric for Cloud Services"** submitted by **M. Anand Kumar (19A81A0644)** for the award of the degree of Bachelor of Technology in the Department of Computer Science and Technology during the academic year 2022-2023.

Name of Project Guide

Mrs. R. Padmaja M.Tech.
Assistant Professor

Head of the Department

Dr. Jaya Kumari M.Tech., Ph.D.
Professor & HOD.

External Examiner

DECLARATION

I hereby declare that the project report entitled **"Design Secure and Efficient Biometric for Cloud Services"** submitted by us to Sri Vasavi Engineering College (Autonomous), Tadepalligudem, affiliated to JNTUK Kakinada in partial fulfilment of the requirement for the award of the degree of B. Tech in Computer Science and Technology is a record of Bonafide project work carried out by us under the guidance of **Mrs. R. Padmaja,** M.Tech. Assistant Professor. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or University.

Project Associate

M. Anand Kumar (19A81A0644)

ACKNOWLEDGEMENT

First and foremost, I sincerely salute to my esteemed institute **SRI VASAVI ENGINEERING COLLEGE,** for giving me this golden opportunity to fulfil my warm dream to become an engineer.

My sincere gratitude to my project guide **Mrs. R. Padmaja,** M. Tech. Assistant Professor Department of Computer Science and Technology**,** for her timely cooperation and valuable suggestions while carrying out this project.

I express my sincere thanks and heartful gratitude to **Dr. D. Jaya Kumari**, Professor & Head of the Department of Computer Science and Technology, for permitting me to do my project.

I express my sincere thanks and heartful gratitude to **Dr. G.V.N.S.R. Ratnakara Rao**, Principal, for providing a favourable environment and supporting me during the development of this project.

I special thanks to the management and all the teaching and non-teaching staff members, Department of Computer Science and Techonolgy, for their support and cooperation in various ways during my project work. It is my pleasure to acknowledge the help of all those respected individuals.

I would like to express my gratitude to my parents, friends who helped to complete this project.

Project Associate

M. Anand Kumar (19A81A0644)

Table of Contents

ABSTRACT

Biometric identification has developed rapidly in recent years because of its convenience and reliability. Due to the sensitivity of biometric data, many privacy-preserving biometric identification schemes have been put forward, exploiting either homomorphic encryption or matrix-transformation. However, existing schemes based on homomorphic encryption generally suffer from low computational efficiency, and existing matrix-transformation-based schemes are insufficiently secure. In this paper, we demonstrate that the matrix-transformation-based privacy-preserving biometric identification scheme recently proposed by Zhu et al. is vulnerable to a known-plaintext attack (KPA). To remedy this security flaw, we propose a new privacy-preserving biometric identification scheme, in which the property of the orthogonal matrix and additional randomness are utilized. Security analysis and comparisons indicate that our scheme can resist not only the KPA attack but also the more powerful chosen-plaintext attack (CPA), which is a reasonable attack in practice. Moreover, our scheme enjoys higher computational efficiency than other similar schemes, which implies our scheme can better support a huge database for practical biometric identification, and it also enhances privacy security of sensitive biometric data.

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 Introduction

When it comes to cyberattacks, phishing remains as one of the ever-popular techniques Biometric identification provides a promising method in access control to authenticate users by their biometric traits, i.e. physiological traits (e.g. fingerprints, iris, face) and behavioral traits (e.g. voice, gait, typing rhythms). Biometric traits can not be lost, stolen, or forgotten like passwords, since they are naturally bound up with individuals [1], [2]. Just as Schneier has said: ''You are your key'' [3]. Due to such a strong bond, biometric identification is a much more reliable and convenient approach than passwords, the most traditional authentication method. With the boom of smartphones, the integration of biometric sensors into mobile phones has boosted the adoption of biometric technologies. For instance, more online banks seek to incorporate biometric identification into their systems [4]. In addition to the online payment, there are many other access control scenarios, which will lead to a growing demand for biometric technologies, such as industrial IoT deployment [5], telecare medical information system [6], smart city [7] and other applications [8]–[10]. According to Research and Markets Ltd., the largest market research store, the global biometric market will reach $42.4 billion by 2021 [11].

Despite the positive prospect and trend in biometric identification, there are still many challenges concerned with the privacy, security, and efficiency since biometric data is highly sensitive and impossible to be revoked and replaced once leaked [2]. For example, if a person's fingerprint is compromised, he/she can not change it like traditional passwords and will no longer rely on it as a security mechanism. Moreover, the biometric data may also reveal sensitive personal information such as genetic information and some information about users' diseases [12], [13]. Therefore, appropriate security and privacy protection scheme should be proposed to resist the disclosure and misuse of biometric data(i.e. biometric template).

1.2 Motivation

Developing a software system is an arduous process which contains planning, analysis, design,implementation, testing, integration, and maintenance. A software engineer is expected to develop a software system on time and within limited the budget which are determined during the planning phase. During the development process, there can be some defects such as improper design, poor functional logic, improper data handling, wrong coding, etc. and these

defects may cause errors which lead to rework, increases in development and maintenance costs decrease in customer satisfaction.

1.3 Scope

It can used in Educational and Private Organizations. So that a group of people in a specify organization can only access to the files

1.4 Project Outline

Chapter1 Introduction

Chapter2 Literature Survey

Chapter 3 System Study and Analysis

Chapter 4 System Design

Chapter 5 Technologies

Chapter 6 Implementation

Chapter 7 Testing

Chapter 8 Screenshots

Chapter 9 Conclusion and future work Reference

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

Biometrics-based privacy-preserving user authentication scheme for cloud-based Industrial Internet of Things deployment:

Due to the widespread popularity of Internet-enabled devices, Industrial Internet of Things (IIoT) becomes popular in recent years. However, as the smart devices share the information with each other using an open channel, i.e., Internet, so security and privacy of the shared information remains a paramount concern. There exist some solutions in the literature for preserving security and privacy in IIoT environment. However, due to their heavy computation and communication overheads, these solutions may not be applicable to wide category of applications in IIoT environment. Hence, in this paper, we propose a new biometric-based privacy preserving user authentication (BP2UA) scheme for cloud-based IIoT deployment. BP2UA consists of strong authentication between users and smart devices using preestablished key agreement between smart devices and the gateway node. The formal security analysis of BP2UA using the well-known real-or-random model is provided to prove its session key security. Moreover, an informal security analysis of BP2UA is also given to show its robustness against various types of known attacks. The computation and communication costs of BP2UA in comparison to the other existing schemes of its category demonstrate its effectiveness in the IIoT environment. Finally, the practical demonstration of BP2UA is also done using the NS2 simulation.

Robust biometric-based user authentication scheme for wireless sensor networks:

Wireless sensor networks (WSNs) are applied widely a variety of areas such as real-time traffic monitoring, measurement of seismic activity, wildlife monitoring and so on. User authentication in WSNs is a critical security issue due to their unattended and hostile deployment in the field. In 2010, Yuan et al. proposed the first biometric-based user authentication scheme for WSNs. However, Yoon et al. pointed out that Yuan et al.'s scheme is vulnerable to the insider attack, user impersonation attack, GW-node impersonation attack and sensor node impersonate attack. To improve security, Yoon et al.'s proposed an improved scheme and claimed their scheme could withstand various attacks. Unfortunately, we will show Yoon et al.'s scheme is vulnerable to the denial-of-service attack (DoS) and the sensor node impersonation attack. To overcome the weaknesses in Yoon et al.'s scheme, we propose a new biometric-based user authentication scheme for WSNs. The analysis shows our scheme is more suitable for practical applications.

- "Secure and Efficient Biometric Systems: A Survey" by Jiaqing Huang, Wei-Qi Yan, and Jiankun Hu. This paper provides a comprehensive survey of the latest advances in secure and efficient biometric systems, including various biometric modalities, feature extraction, template protection, and evaluation metrics.

- "Design of Secure and Efficient Biometric Systems: A Review" by S. B. Sreeja and G. R. Bindu. This paper reviews the various techniques used to design secure and efficient biometric systems, including feature extraction, matching, and template protection. It also discusses the challenges and future directions in this field.

- "Biometric Systems: Security and Efficiency Analysis" by Carlos Ramos and Paulo Mateus. This paper provides a comprehensive analysis of the security and efficiency of biometric systems, including fingerprint, face, and iris recognition. It also discusses the limitations and future directions in this field.

- "A Survey of Biometric Cryptosystems and Cancelable Biometrics" by Yagiz Sutcu and Alexiou Athanasios. This paper provides a survey of the latest advances in biometric cryptosystems and cancelable biometrics, which are two important techniques used to enhance the security of biometric systems.

- "Efficient and Secure Biometric Systems: A Survey" by Mustafa Ulutas and Betul Karakus. This paper provides a survey of the latest research in efficient and secure biometric systems, including various biometric modalities, feature extraction, template protection, and performance evaluation.

CHAPTER 3

SYSTEM STUDY AND ANALYSIS

3. SYSTEM STUDY AND ANALYSIS

3.1 Problem Statement

As there is a drastic increase in the technology there is chance of having attack on the organization. No matter the method, all these hacking methods are focused on getting you to expose your data. That could be personal data or financial information and can be able to modify the data. Regardless, these cybercriminals are trying to steal from you for their own financial benefit.

3.2 Existing System

The increasing demand for a reliable and convenient authentication promotes the development of biometric identification. However, many data breaches raise increasing concerns on privacy issues recently. Various solutions on privacy-preserving biometric identification have been recommended.

Directly encrypting biometric template plaintexts and matching template ciphertexts bit by bit seem to be the most robust method to protect sensitive template data from disclosure [2]. However, since there are inherent noises in biometric feature extracting process, direct encrypting method tends to amplify small differences, generally resulting in the failure of identification. To improve the fault tolerance of template ciphertexts matching, Jin et al. [14] proposed a two-factor authentication scheme based on iterated inner products between the template and tokenized pseudo-random numbers, well-known as BioHashing. But the scheme's performance is not as good as claimed when an attacker steals the tokenized pseudo-random numbers [15]. Later, by resorting to error correcting codes, Juels and Sudan [16] presented a fuzzy vault scheme, but it is extremely vulnerable to record-multiplicity attacks, in which an attacker can recover a particular template from a collection of multiple enrollment template encodings [17].

Instead of bitwise matching of template ciphertexts, calculating the Euclidean distance between two templates is another way to determine whether they are from the same user. However, the computation of distance is usually conducted by the cloud server, who holds only the ciphertexts of the biometric templates for the sake of privacy protection. This leads to a challenge in how the cloud server computes the distance of template plaintexts through operations on the corresponding ciphertexts. Therefore, the promising homomorphic encryption

8

is introduced to this area. Barni et al. [18] proposed a privacy-preserving fingerprint authentication scheme based on an additively homomorphic encryption called Pallier scheme [19]. However, due to the low performance, their scheme is limited by the size of database and number of concurrent requests. Later, Catalano and Fiore [20] presented a method to boost additively homomorphic encryption to a more complicated cryptosystem supporting degree-2 computation on encrypted data. Based on Dario et al.'s work and Paillier cryptosystem, Im et al. [21] implemented a palm print authentication. But the efficiency is not yet satisfactory. Further, Zhu et al. [22] designed a more efficient system model by utilizing BGN cryptosystem [23], a somewhat homomorphic encryption which is able to evaluate 2-DNF formulas on ciphertexts. Nevertheless, their experimental results are performed on a small dataset named FVC2006 which contains only 150 fingers, so it seems that their scheme can hardly support a huge database for practical usage.

To achieve higher performance and scalability of biometric identification, privacy-preserving schemes based on matrix transformation were proposed [24]–[27] as alternatives to those schemes based on homomorphic encryption. Yuan and Yu [24] presented the first efficient privacy-preserving biometric identification scheme based on matrix transformation. However, Zhu et al. [25] pointed out that their system can be destroyed by a collusion attack launched by malicious users and the cloud. To remedy the deficiency of [24], two improved protocols were put forward, in which additional randomness is introduced [26], [27]. Nevertheless, the computational efficiency of Hu et al.'s scheme [27] is not suitable for deployment in practical scenarios, which is even lower than [24]. Moreover, we find that both Zhu et al.'s scheme [26] and Hu et al.'s scheme [27] are still insufficiently secure as they are vulnerable to known plaintext attack (KPA) under their security assumption.

3.3 Limitations of the Existing System

- The method of using PIN to protect the data
- Although this method is simple and efficient, it can be unlocked by random passwords generator that checks all probabilities of PIN, Since the length of the PIN is short. Which has severe limitations.
- Two step verification is also not much efficient.

3.4 Proposed System

We are planning a design a model to further exploit the high performance of matrix transformation [28] and remedy the security flaws in previous schemes, we propose an efficient biometric identification with enhanced privacy security to resist not only the KPA attack but also the chosen-plaintext attack (CPA), which is reasonable in practice.

Based on the typical security assumption mentioned in Zhu et al.'s work [26], we demonstrate their scheme is not KPA-secure as they claimed, in which an attacker can recover any template by the collusion of cloud server with malicious users.

3.5 Advantages of Proposed System

• We consider a more adversarial setting — CPA attack, which has been used as a de facto standard for checking the security of cryptographic schemes in classical cryptology [29]. Besides the typical security assumption of biometric identification [26], we find the CPA attack is also very reasonable in practice. We formally present a more powerful threat model by extending the typical security model with the reasonable CPA attack.

• We propose a new biometric identification scheme with enhanced privacy security. The security analysis shows our scheme achieves a higher level of privacy protection, in the sense that our proposed scheme can defend against not only various attacks [26] but also the CPA attack.

• We present a detailed implementation of the proposed model provides higher computational efficiency than existing biometric identification protocols.

3.6 Functional Requirements

Functional requirements drive the application architecture of a system. A requirement will generate use cases after gathering and validating a set of functional requirements. Functional requirements may be technical details, data manipulation and other some functionality of the project is providing the information to use. Software that is scalable has the ability to handle a widevariety of system configuration sizes.

The functional requirements should specify the ways in which the system may be expected to scale up. Our system can be easily expandable. Functional requirements are expressed in the

form system shall do<requirement>. The plan for implementing functional requirements is detailed in the system design. In requirements engineering, functional requirements specify particular results of a system.

If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; It influences choices made during the design, development, testing and deployment of the system.In software engineering, a functional requirement defines a function of a software system-oriented component. A function is described as a set of inputs, the behavior, and outputs (seal also software).

Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

Behavioral requirements describing all the cases where the system uses the functional requirements are captured in usecases.

Generally, functional requirements are expressed in the form system shalldo<requirement>. The plan for implementing functional requirements is detailed in the system design. In requirements engineering, functional requirements specify particular results of a system.

Functional requirements drive the application architecture of a system. A requirements analyst generates usecases after gathering and validating a set of functionalrequirements.

The hierarchy of functional requirements is: user / stakeholder request-> feature->use case business rule.

Functional requirements drive the application architecture of a system. A requirements analyst generates use cases after gathering and validating a set of functional requirements.

Functional requirements may be technical details, data manipulation and other specific functionality of the project is to provide the information to the use.

3.7 Non -Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.The project non-functional requirements include the following.

Availability: A system's "availability" or "uptime" is the amount of time that is operational and available for use. It's related to is the server providing the service to the users in displaying

images. As our system will be used by thousands of users at any time our system must be available always. If there are any cases of update, they must be performed in a short interval of time without interrupting the normal services made available to the users.

Efficiency: Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, band width etc. All of the above-mentioned resources can be effectively used by performing most of the validations at client side and reducing the work load on server by using JSP instead of CGI which is being implemented now.

Flexibility: If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules or modifying the logical database schema of the existing applications.

Portability: Portability specifies the ease with which the software can be installed on all necessary platforms, and the platforms on which it is expected to run. By using appropriate server versions released for different platforms our project can be easily operated on any operating system, hence can be said highly portable.

Scalability: Software that is scalable can handle a wide variety of system configuration sizes. The nonfunctional requirements should specify the ways in which the system may be expected to scale up(by increasing hardware capacity, adding machines etc.). Our system can be easily expandable. Any additional requirements such as hardware or software which increase the performance of the system can be easily added. An additional server would be useful to speed up the application.

Integrity: Integrity requirements define the security attributes of the system, restricting access to features or data to certain users and protecting the privacy of data entered into the software. Certain features access must be disabled to normal users such as adding the details of files, searching etc. which is the sole responsibility of the server. Access can be disabled by providing appropriate logins to the users for only access.

Usability: Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users. Hyperlinks will be provided for each and every service the system provides through which navigation will be easier. A system that has high usability makes the work of the user easier.

Performance: The performance constraints specify the timing characteristics of the software.

3.8 System Requirements

3.9 3.8.1Software Requirements

Operating System	Windows 8 / 10
IDE	NetBeans
Coding Language	Java
Dataset	A dataset of Users With respective Fingerprints

Packege imported:

    fingerprint. Match.

3.8.2 Hardware Requirements

Processor-Corei5 **,RAM-** 4GB

Hard Disk- 20GB

Keyboard –Standard Keyboard

CHAPTER 4

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 System Architecture Design

It consists of three modeling diagrams:

Workflow Diagram.

Use Case Diagram.

Class Diagram.

Activity diagram.

System Architecture:
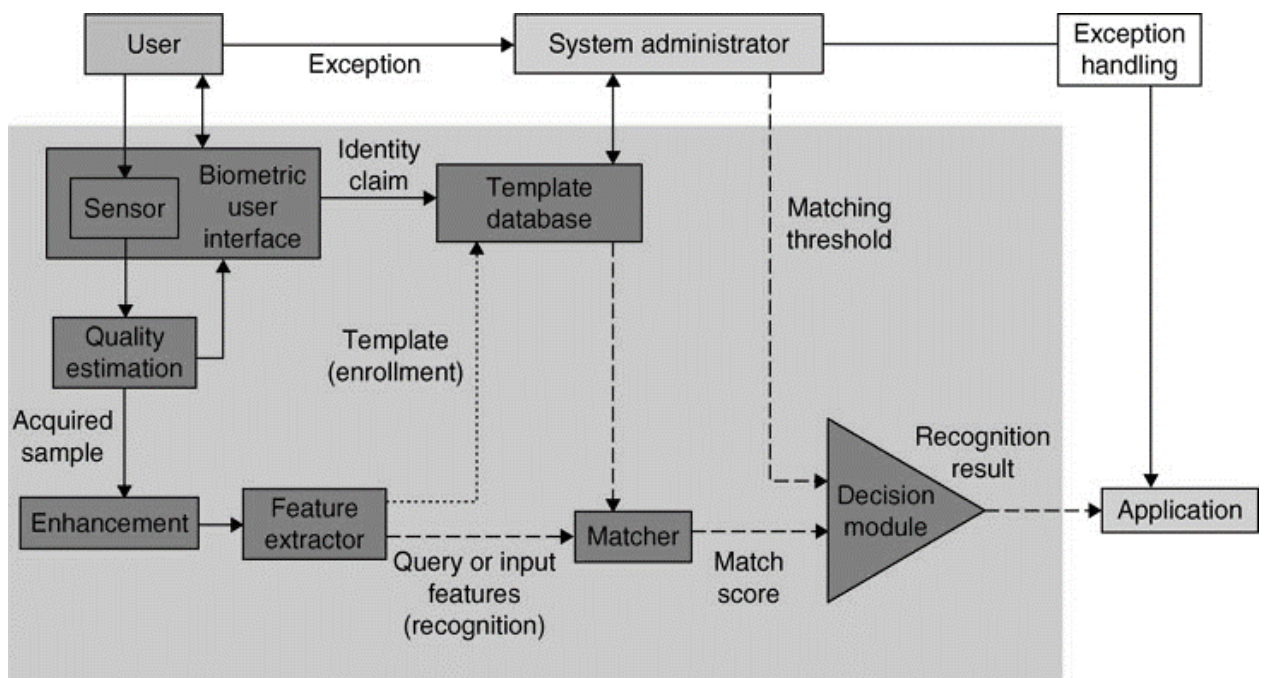
Figure.1: System Architecture

4.2 UML Diagrams:

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.
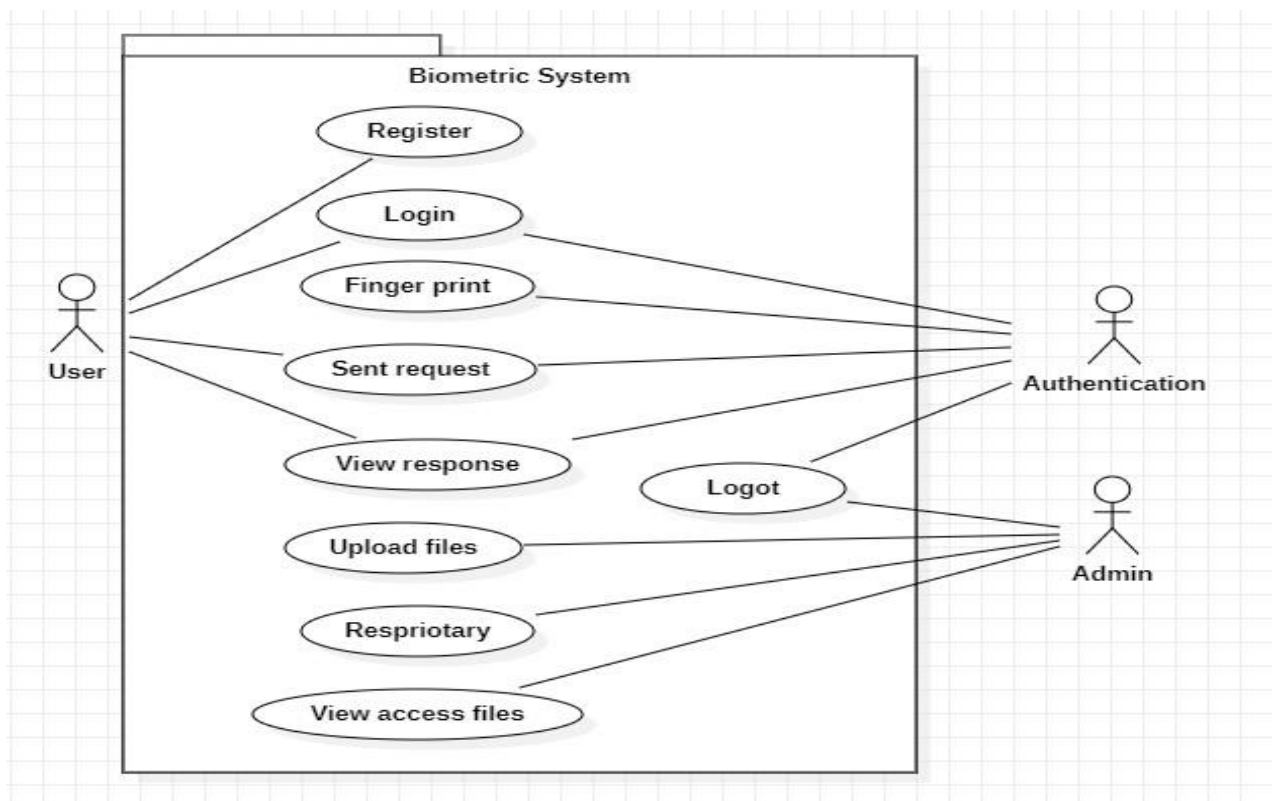
Figure 2: Use Case Diagrams

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.
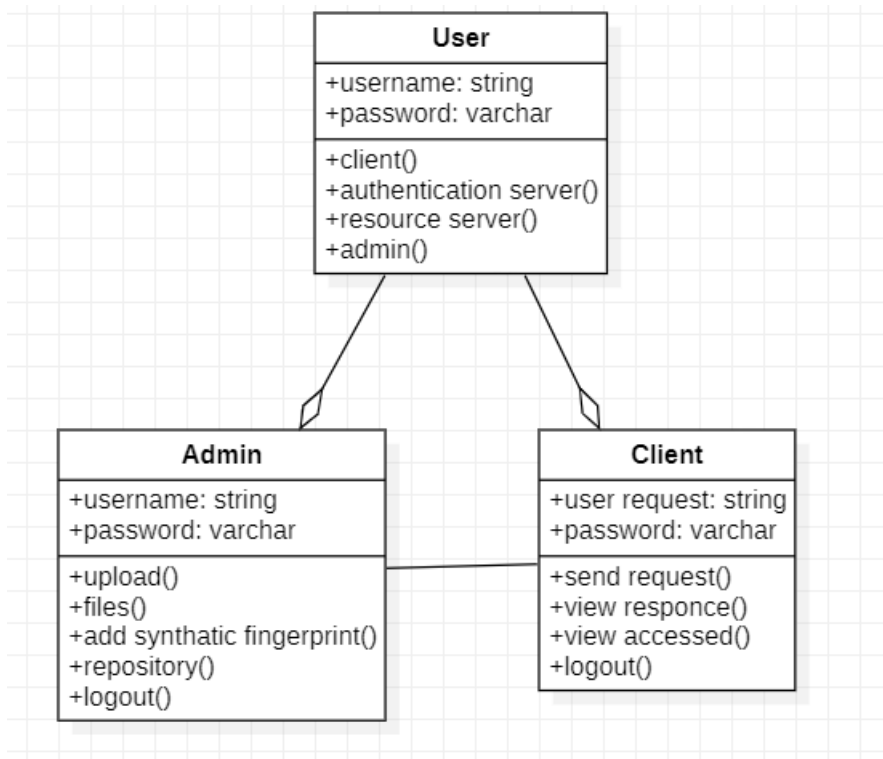
Figure 3: Class Diagram

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.
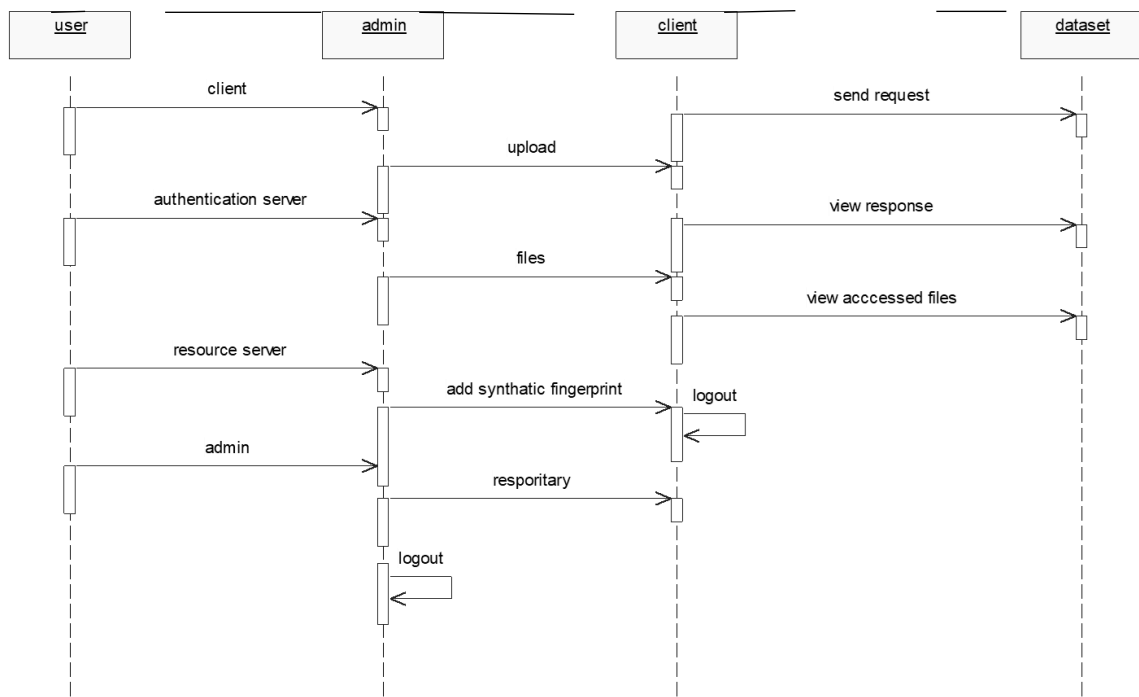
Figure 4: Sequence Diagram

COLLABORATION DIAGRAM: -
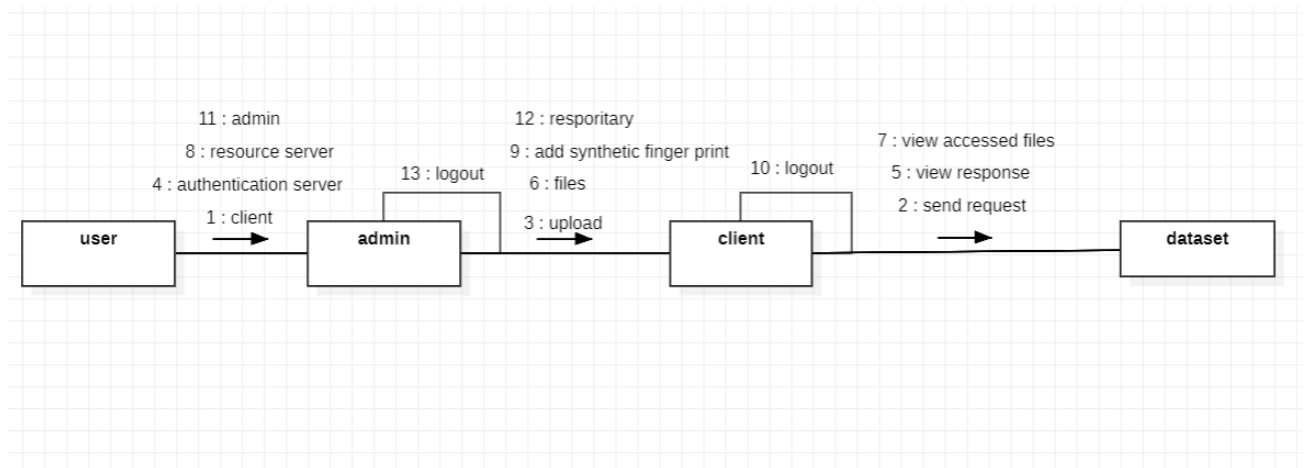
Figure 5: Collaboration Diagram

ACTIVITY DIAGRAM:

Figure 6: Activity Diagram

19
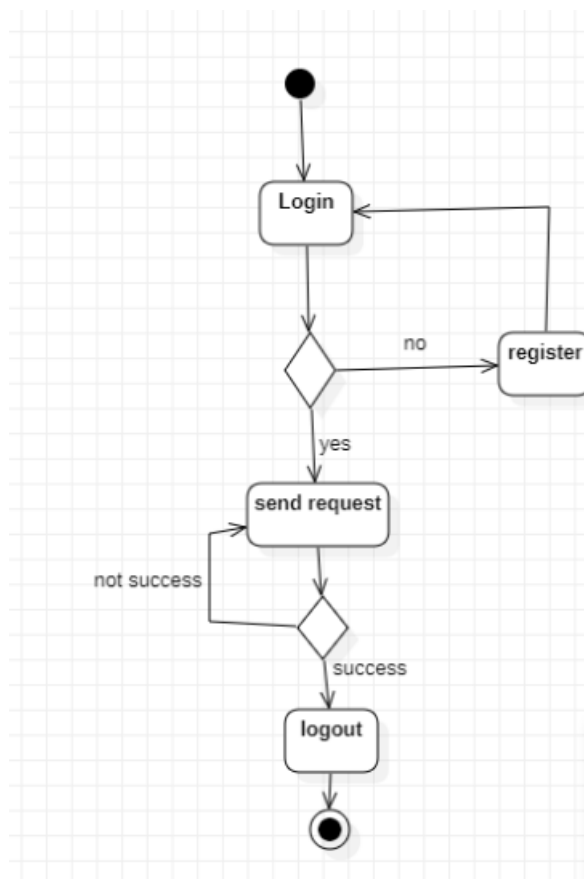
CHAPTER 5

TECHNOLOGIES

5.TECHNOLOGIES

5.1 About Java

Java is a popular high-level programming language that is widely used for developing web, mobile, and desktop applications. It is an object-oriented language that offers a range of features, including inheritance, polymorphism, and encapsulation.

One of the major advantages of using Java is its platform independence. Java programs can run on any machine that has a Java Virtual Machine (JVM), which means that the code written once can be executed on multiple platforms without any modification. This feature has made Java an attractive language for building large-scale, cross-platform applications.

Another strength of Java is its robustness and reliability. Java has a strong memory management system and built-in exception handling capabilities, which makes it less prone to crashes and errors. Additionally, Java has a rich set of libraries and APIs that provide developers with a wide range of functionality, such as networking, database connectivity, and user interface development.

Java also supports multi-threading, which allows developers to write programs that can perform multiple tasks simultaneously. This feature makes Java well-suited for developing high-performance applications that require parallel processing.

In terms of security, Java has a number of built-in features that help to prevent common security issues such as buffer overflows and memory leaks. Additionally, Java has a sandbox security model that restricts the code from performing certain actions that could potentially harm the system.

Overall, Java is a powerful and versatile programming language that offers a range of features and benefits for developers. Its platform independence, robustness, reliability, and security make it an ideal choice for developing complex, large-scale applications.

5.2 JSP:

JavaServer Pages (JSP) is a technology used to create dynamic web pages and web applications. It is a server-side technology that allows developers to embed Java code in HTML pages. JSP pages are compiled into servlets by the server and can be deployed on any web server that supports Java.

JSP technology provides a robust, scalable, and platform-independent way to build web applications. With JSP, developers can create dynamic web pages that can display data from databases, process user input, and generate dynamic HTML or XML content. JSP pages can also be used to create reusable components, such as header and footer sections, that can be included on multiple pages.

One of the significant benefits of using JSP technology is that it separates presentation logic from business logic. This separation allows developers to focus on writing business logic in Java and leaves the presentation layer to be handled by the JSP pages.

To develop JSP pages, developers use an integrated development environment (IDE) such as Eclipse . These IDEs provide a rich set of features for developing and debugging JSP pages. JSP pages can also be developed using a text editor and the Java Development Kit (JDK).

JSP technology is widely used in enterprise web applications and is a core technology for the Java Platform, Enterprise Edition (Java EE). It provides a powerful way to create dynamic web applications that can be scaled to handle millions of users. With JSP technology, developers can create web applications that are fast, secure, and highly reliable.

5.3 NetBeans:

NetBeans is an open-source integrated development environment (IDE) for Java and other programming languages such as C++, HTML, JavaScript, and PHP. It provides a variety of features and tools that enable developers to create robust, high-quality applications efficiently. Some key features of NetBeans include:

Code editor: NetBeans provides a code editor that supports code highlighting, auto-completion, and error highlighting. It also provides support for version control systems such as Git, SVN, and Mercurial.

Debugging and Profiling: NetBeans provides a powerful debugger that helps developers to identify and fix issues in their code. It also provides profiling tools that enable developers to analyze the performance of their applications.

GUI Builder: NetBeans provides a GUI Builder that enables developers to create GUIs quickly and easily without writing any code. It supports drag-and-drop functionality and provides a wide range of pre-built components.

Plugin system: NetBeans has a plugin system that enables developers to extend its functionality by installing plugins. There are many plugins available for NetBeans that provide additional features and tools.

Multi-platform support: NetBeans runs on Windows, macOS, and Linux, and it supports multiple programming languages.

Overall, NetBeans is a powerful IDE that provides a rich set of features and tools for developing applications in Java and other programming languages. It is widely used by developers around the world and has a large and active community that provides support and plugins for the IDE.

5.4 SQL:

SQL Enterprise refers to the enterprise version of SQL Server, which is a relational database management system (RDBMS) developed by Microsoft. SQL Server Enterprise offers advanced features such as high availability, security, and business intelligence capabilities. It is designed for enterprise-level organizations with high data transactional volumes, complex data processing requirements, and the need for scalability and availability.

SQL Server Enterprise supports features like advanced analytics, machine learning, and in-memory technology that allows faster data processing, real-time analytics, and faster decision-making. It also includes additional features like transparent data encryption, dynamic data masking, and row-level security to ensure data security and compliance with regulatory standards.

SQL Server Enterprise also provides high-availability solutions such as Always On Availability Groups and Failover Clustering, which help ensure maximum uptime and minimize data loss in case of a disaster or system failure.

Overall, SQL Server Enterprise is a comprehensive, high-performance RDBMS designed for mission-critical applications, large-scale data processing, and demanding enterprise-level workloads.

Database Connection:

```java
package com.database;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class Dbconnection{
    public static Connection getcon(){
        Connection con = null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Designing", "root", "root");
        }
        catch(Exception e){
            e.printStackTrace();
        }
        return con;
    }
}
```

Queries:

```java
package com.database;


import java.sql.*;

public class Queries {
    public static ResultSet rs;
    public static int i;

    public static ResultSet getExecuteQuery(String query){
        try{
        Connection con=Dbconnection.getcon();
     Statement st=con.createStatement();
     rs=st.executeQuery(query);
        }catch(Exception e){
            System.out.println(e);
        }
        return rs;
    }
    public static int getExecuteUpdate(String query){
        try{
        Connection con=Dbconnection.getcon();
     Statement st=con.createStatement();
     i=st.executeUpdate(query);
        }catch(Exception e){
            System.out.println(e);
        }
        return i;
    }
}
```

CHAPTER 6

IMPLEMENTATION

6. IMPLEMENTATION

6.1 Implementation

PRELIMINARIES:

In this section, we first introduce the system model and how to represent biometric templates. Then, we present the formalized threat model and security definition for biometric identification. In the description of threat model, we also analyze rationality of the CPA attack in practice.

Fingerprint matching is the process of comparing two fingerprint images to determine whether they belong to the same person. Fingerprint matching is a common method used for biometric authentication and is widely used in law enforcement and forensic investigations.

Fingerprints are unique to each individual, and no two fingerprints are identical. The pattern of ridges and valleys on a fingerprint is unique to each individual and can be used to identify a person with a high degree of accuracy. Fingerprint matching algorithms use several methods to compare the patterns of two fingerprints.

The most common method used for fingerprint matching is minutiae matching. This method involves identifying the key features, or minutiae, on a fingerprint, such as ridge endings and bifurcations. The algorithm then compares the location, orientation, and shape of these minutiae between two fingerprint images to determine if they match.

Another method used for fingerprint matching is pattern matching. This method involves comparing the overall pattern of ridges and valleys on a fingerprint to determine if they match. Pattern matching is less precise than minutiae matching but can still be used for rapid identification in large-scale applications.

Fingerprint matching technology has improved significantly in recent years, and modern algorithms can achieve very high accuracy rates. Fingerprint matching is a fast and reliable method of biometric authentication and is widely used in applications such as border control, law enforcement, and mobile device security.

A. SYSTEM MODEL:

The model in this paper follows the typical model introduced by Zhu et al's scheme [26]. As shown in Fig. 1, there exist three entities: data owner, cloud server and users. The data owner holds a database containing numerous biometric data $hbii \; m \; i=1$ (i.e. reference template) which has been enrolled by users in the system. Based on this model, a biometric identification scheme generally includes three stages: preparation stage, request stage and identification stage. In the preparation stage, the data owner encrypts reference templates bi and outsources ciphertexts $Ci \leftarrow Enc(sk, bi)$ to the cloud server for storage. In the request stage, when a user requests for identification and sends his/her biometric trait bc (i.e. sample template) to the data owner, the data owner encrypts the sample template plaintext bc and sends the ciphertext $Cc \leftarrow Enc(sk, bc)$ to the cloud server. In the identification stage, upon receipt of the request from the data owner, the cloud server performs operations on ciphertexts to figure out whether there is a matched reference template. Finally, the identification result will be sent to the data owner and user successively.

and user successively. From the description of the system model, we can find the security of biometric templates depends on the operation $Enc \; (sk, \cdot)$. Additionally without loss of generality, we adopt FingerCode [30] to represent biometric templates similar to the existing work [24], [26], [27].

B. BIOMETRIC TEMPLATE REPRESENTATION:

We apply FingerCode which is got by a filter-based algorithm [30] to represent biometric templates. Given a fingerprint image, the filter-based algorithm uses a bank of Gabor filters to extract features in the fingerprint and then outputs a compact fixed length (generally set as 640) vector, i.e. FingerCode

C. THREAT MODEL :

According to the typical threat model [26], the cloud server is assumed ''honest-but-curious'' or ''semi-honest'', which means the cloud server strictly executes the designed protocol, but tries to analyze the received messages to learn additional information about the honest users' biometric templates or the data owner's secret key. The semi-honest cloud server may even collude with malicious users further to attack the biometric identification system. On the basis of attack abilities, attackers are classified into three levels [26], with respect to ciphertext-only attack, known-candidate attack and known-plaintext attack respectively.

In addition, we further introduce a new level through considering the more powerful CPA attack, which is reasonable in practice.

Remark : As indicated in the survey of security and privacy issues for biometric-based remote authentication in cloud [31], malicious users have the ability to forge biometric templates during enrollment, which means the attacker can get any plaintext of fake reference template. Then in the preparation step, the reference template plaintext will be encrypted by the data owner and the ciphertext will be sent to the semi-honest cloud server. For the reason that the semi-honest cloud server can collude with malicious users, the attacker will get arbitrary plaintext and corresponding ciphertext of fake reference templates. Therefore, CPA attack for biometric identification is reasonable in practice.

As described above, the CPA attack is also reasonable in practice. So we extend the typical threat model [26] by adding the CPA attack as level-4. To better formalize the strength of attackers, we classify attackers as follows:

• Level 1: Attackers can only observe the encrypted template in the cloud. This follows the well-known ciphertext-only attack model [29].

• Level 2: In addition to the encrypted templates in the cloud, attackers can access to some template plaintexts but do not know the corresponding ciphertexts in the encrypted database, similar to the known-candidate attack model [32].

• Level 3: Besides all the abilities in level-2, attackers are able to get a set of template plaintexts and know the corresponding ciphertexts. This level follows the known-plaintext attack (KPA) model [29].

• Level 4: With enhanced ability, malicious users can forge templates during biometric database enrollment and collude with the semi-honest cloud server [31]. So, attackers can get any template plaintext and the corresponding ciphertext, which follows the chosen-plaintext attack (CPA) model [29].

D. SECURITY DEFINITION :

For the threat model, a higher-level attack is more powerful than a lower-level one. If a scheme can defend against a higher-level attack, it can resist a lower-level one as well. So we define security resisting above threat model based on level-4 attack, i.e. CPA attack.

PERFORMANCE ANALYSIS :

To evaluate the performance of our scheme, we compare both computational and communication complexity with existing works, and then we fully implemented them to evaluate the practicality numerically. The analysis shows our scheme may downgrade the efficiency in the preparation stage, but achieves a significant improvement of performance in the identification stage. Due to database outsourcing is only a one-off process and identification is the most frequent operation, our scheme is more suitable for practical applications.

COMPLEXITY ANALYSIS :

As described in Section V, our scheme can be decomposed into three stages. In the 1st preparation stage, to outsource the whole database, the owner should encrypt every record in the database by performing vector-matrix multiplication, whose computational complexity is $O(n^2)$. Therefore, the total complexity of this stage is $O(mn^2)$, where m is the total number of the Finger Code records in the database. In the 2nd request stage, a sample template is sent to data owner for identification. Then the data owner will encrypt it similar to what has been done in the 1st stage. So, the encryption also costs $O(n^2)$. In the 3rd identification stage, the encrypted sample template is submitted to the cloud server. All the ciphertexts are vectors. Thus, the total time complexity of the inner product is $O(mn)$. For communication complexity, besides the one-off outsource cost of $O(mn)$ for the 1st stage, the request costs $O(n)$ and the cost of identification response is $O(1)$.

To compare our protocol with previous schemes [26], [27] intuitively, we illustrate the complexities in Table 3. As shown in Table 3, for computation complexity, in the preparation stage, our scheme has similar cost $O(mn^2)$ as Zhu et al.'s scheme [26] does, lower than the cost $O(mn^3)$ of Hu et al.'s scheme [27]. In the identification stage, the data owner of our scheme need only to encrypt the sample template with overhead $O(n^2)$. Besides higher cost $O(n^3)$ in template encryption, the data owner of the other two works still has to compute the Euclidean distance between the sample template and the reference template corresponding to the retrieved index. The cloud server of our protocol will spend only $O(mn)$ working out whether the identification is successful.

	Participant	Stage	Zhu et al.'s scheme [26]	Hu et al.'s scheme [27]	Our proposed scheme
Computation	data owner	preparation	$O(mn^2)$	$O(mn^3)$	$O(mn^2)$
		request	$O(n^3)$	$O(n^3)$	$O(n^2)$
		retrieval	$O(n)$	$O(n)$	/
	cloud server	identification	$O(mn^2)$	$O(mn^3)$	$O(mn)$
Communication	data owner	preparation	$O(mn)$	$O(mn^2)$	$O(mn)$
		request	$O(n^2)$	$O(n^2)$	$O(n)$
		retrieval/result	$O(1)$	$O(1)$	$O(1)$
	cloud server	identification	$O(1)$	$O(1)$	$O(1)$
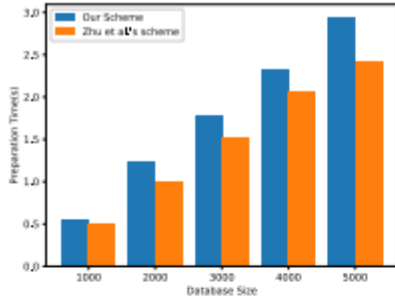
(a) Time costs in preparation stage of our scheme and [26]
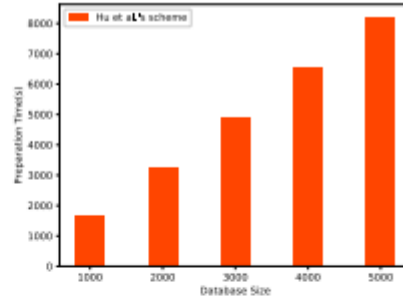
(b) Time costs in preparation stage of [27]
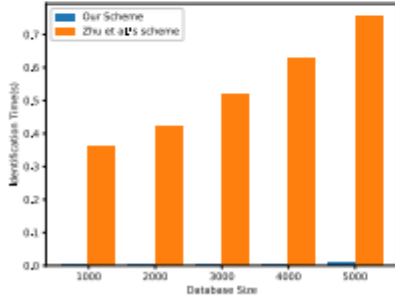
(c) Time costs in identification stage of our scheme and [26]
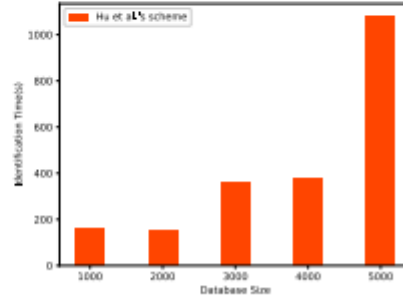
(d) Time costs in identification stage of [27]

Comparison of communication cost in different stages between our proposed scheme and [26], [27].

	Zhu et al.'s scheme [26]	Hu et al.'s scheme [27]	Our proposed scheme
preparation	$5.14(m+1)$KB	$6.29m$MB	$5.17m$KB
identification	3.14MB	6.29MB	5.17KB

While the cloud server of [26] and [27] will afford heavier cost $O(mn^2)$ and $O(mn^3)$ respectively to calculate the relative distance Pi and find the most closely matched template's index. So the identification efficiency of our scheme is much better than the other two works. For communication cost, on the data owner side, in the preparation stage, our scheme and [26] transmit encrypted vectors to the cloud, so the complexity is $O(mn)$. While [27] transfers matrix ciphertexts with cost $O(mn^2)$. In the identification stage, our scheme costs $O(n)$ to send a request containing a vector to the cloud. However, [26] and [27] transmit request consisting of matrices with complexity $O(n^2)$. At last, all schemes cost $O(1)$ to return the identification result to user. On the cloud server side, all schemes will send the cloud computing result (identification result or the most closely matched index) to the data owner with communication overhead $O(1)$.

B. EXPERIMENTAL EVALUATION

1) EXPERIMENT SETUP :

To evaluate the improvement of our scheme, compared with the existing schemes [26], [27], we use a synthetic datasets consisting of randomly generated 640-dimensional vectors to represent the FingerCodes, as [26] and [27] does.
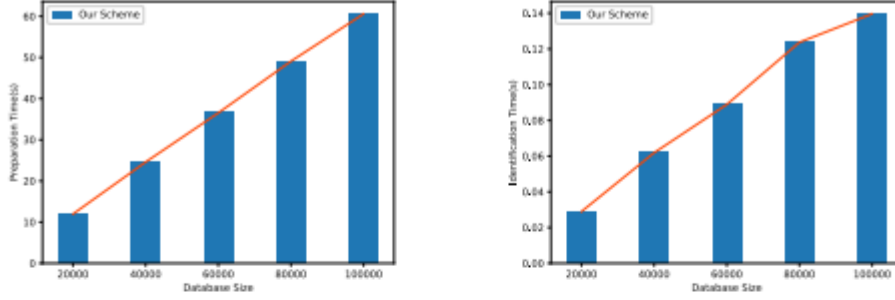
(a) Time costs in preparation stage of our scheme

(b) Time costs in identification stage of our scheme

FIGURE 3. Time cost for lager databases in different stages of our proposed scheme.
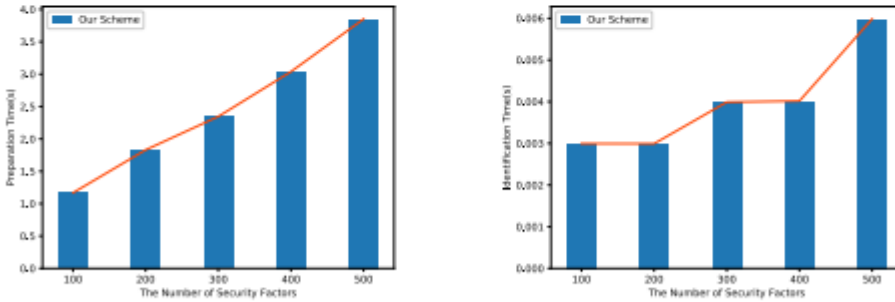
(a) Time costs in preparation stage of our scheme

(b) Time costs in identification stage of our scheme

FIGURE 4. Time cost for higher security in different stages of our proposed scheme.

RESULTS OVER SYNTHETIC DATA :

For a better performance evaluation, we first compare all schemes with the size of database m varying from 1000 records to 5000 records. And then we test the performance of our proposed scheme with larger databases. Further, we will show the impact of extending templates for higher security with more security factors From Fig. 2, we can see the efficiency of our scheme and Zhu et al.'s scheme [26] is much better than Hu et al.'s scheme [27]. When the database has 5000 records, Fig. 2(b) shows the time cost of preparation stage for [27] is 2.28 hours, much slower than our scheme's 2.94s and 2.43s of [26] shown in Fig. 2(a). Because template encryption of our scheme is more complicated than [26], our scheme has a tiny delay. Note that this stage is an one-off process, it also proves that our scheme has a comparable performance in this stage. Refer to Fig. 2(c) and Fig. 2(d), our scheme costs only 0.008s, much more efficient than 18.01 minutes of [27]. And compared to 0.76s of [26], our scheme saves 98.95% time. The

identification is the most process executed, so our scheme has the best performance and can be applied to a much larger size of biometric database

The communication cost is described in Table 4, where m is the number of database records. Moreover, we test our scheme using larger databases, with the size varying from 20000 to 100000. The result is described in Fig. 3(a) and Fig. 3(b). It shows the performance of our scheme is practical.

Further, we test the impact of extending templates with more security factors for higher security, with the size of the database set 1000. The result is given in Fig. 4(a) and Fig. 4(b). It is revealed that more security factors will affect the preparation stage but have less influence on the identification. So, the increasing overhead for higher security is tolerable.

Sample Codes:

Client Registration:

```
<h2>Client Registration Here</h2>
    <form action="ClientRegAction" method="post" enctype="multipart/form-data">
        <table>
            <%String pkey=PrivateKey.randomAlphaNumeric(20);


            %>
            <tr><th>Private Key</th><td><input type="text" name="pkey" value="<%=pkey%>" readonly=""></tD></tr>
            <tr><th>Name</th><td><input type="text" name="name" required=""></tD></tr>
            <tr><th>Email</th><td><input type="email" name="email" required=""></tD></tr>
            <tr><th>Mobile</th><td><input type="number" name="mobile" required=""></tD></tr>
            <tr><th>Address</th><td><input type="text" name="address" required=""></tD></tr>
            <tr><th>UserName</th><td><input type="text" name="uname" required=""></tD></tr>
            <tr><th>Password</th><td><input type="password" name="pass" required=""></tD></tr>
             <tr><th>Finger Print</th><td><input type="file" name="image" required=""></tD></tr>
            <tr><th></th><td><input type="submit" value="Register">

                    <a href="Client.jsp">Login</a>
                </tD></tr>
        </table>
    </form>
```
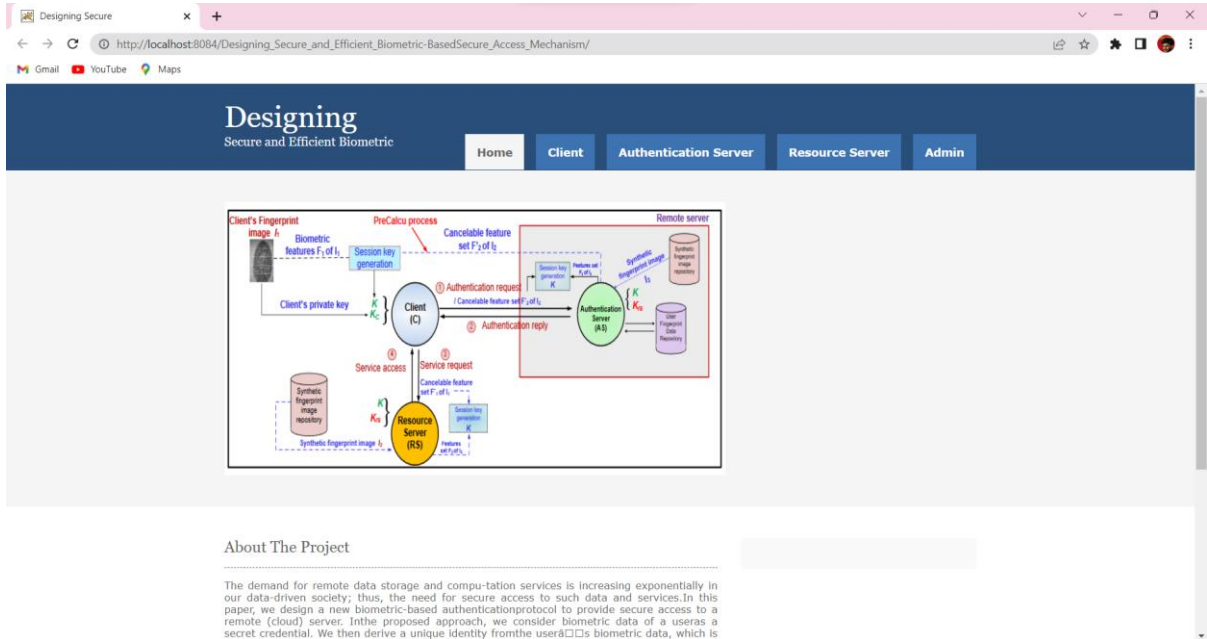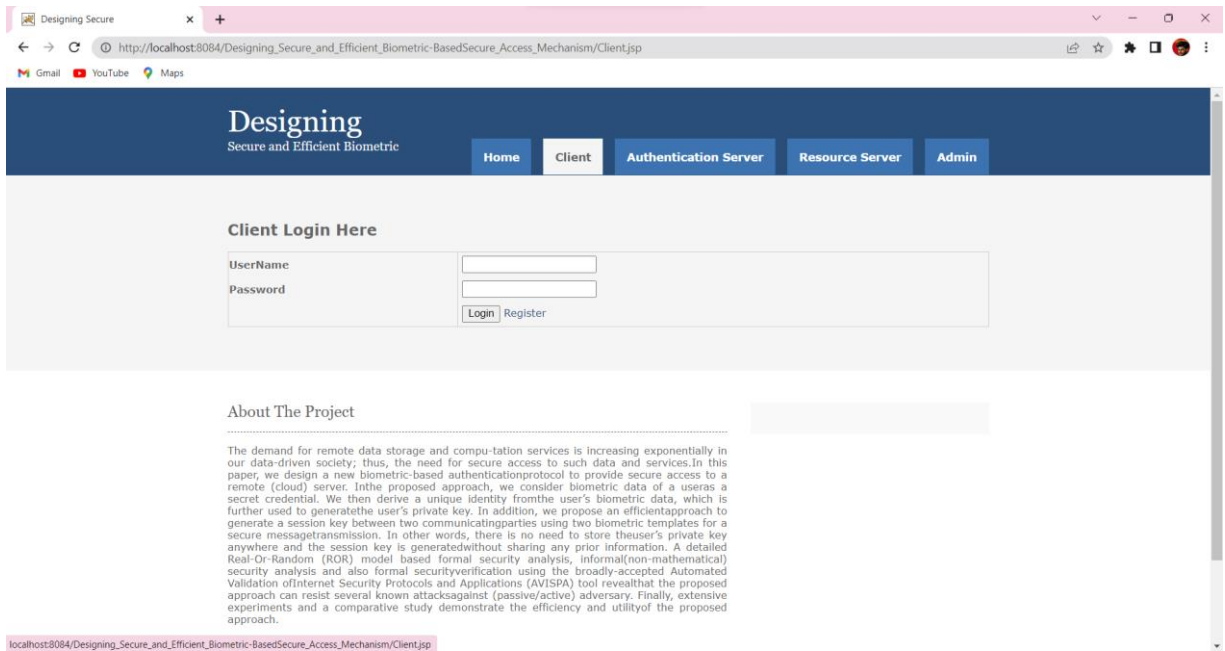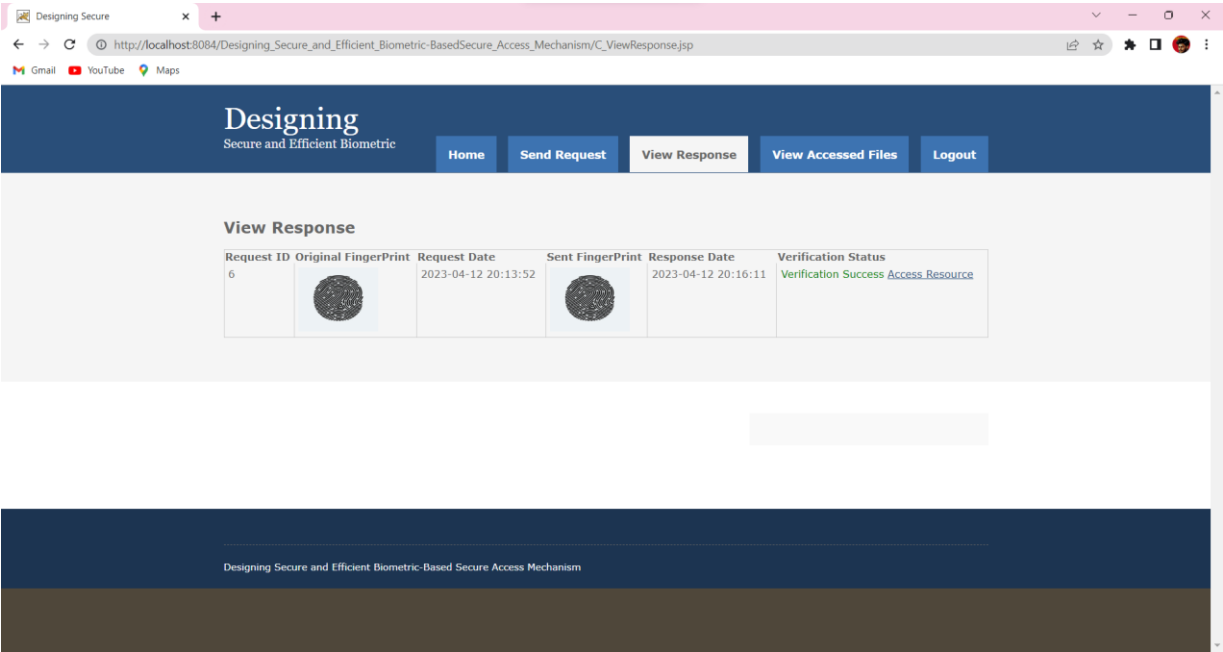
File Upload:

```
<% String msg=request.getParameter("msg");
    %>
    <%if(msg!=null){%>
    <h2><font color="red"><%=msg%></font></h2>
    <%}%>

    <form action="UploadFile" method="post" enctype="multipart/form-data">
        <table>
            <tr><th>File Domain</th><td><input type="text" name="domain" required=""></td></tr>
            <tr><th>Choose File</th><td><input type="file" name="file" required=""></td></tr>
            <tr><th></th><td><input type="submit" value="Upload"></td></tr>
        </table>

    </form>
```

Finger Print Matcher:

```java
package com.fingerprint.match;

import com.machinezoo.sourceafis.FingerprintMatcher;
import com.machinezoo.sourceafis.FingerprintTemplate;
import com.machinezoo.sourceafis.FingerprintImage;
import java.nio.file.Paths;
import java.nio.file.Files;


public class FingerprintMatch {

    public static boolean getImages(byte[] source,byte[] destination){
        boolean matches=false;
    try{
     FingerprintTemplate probe = new FingerprintTemplate(
    new FingerprintImage()
        .dpi(500)
        .decode(source));
FingerprintTemplate candidate = new FingerprintTemplate(
    new FingerprintImage()
        .dpi(500)
        .decode(destination));
double score = new FingerprintMatcher()
    .index(probe)
    .match(candidate);
matches = score >= 40;


    }catch(Exception e){
     System.out.println(e);
    }
    return matches;
}
}
}
```

CHAPTER 7

TESTING

7. TESTING

7.1 Test objectives

- All field entries must work properly.

- Pages must be activated from the identified link.

- The entry screen, messages and responses must not be delayed

7.2 Test Case Design

- The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring it.

- Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.3 Test strategy and approach

- Field testing will be performed manually, and functional tests will be written in detail.

7.3.1 Unit testing

- Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.

- It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.3.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.3.3 Functional testing

- Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

- Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.

- Invalid Input: identified classes of invalid input must be rejected.

- Functions: identified functions must be exercised.

- Output: identified classes of application outputs must be exercised.

- Systems/Procedures: interfacing systems or procedures must be invoked.

- Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.3.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, orat least its purpose. It is purpose. It is used to test areas that cannot be reachedfrom a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the innerworkings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software undertest is treated, as a black box. you cannot–see into it. The test provides inputs and responds to outputs without considering how the software works.

7.3.5 Acceptance Testing

- Verify that the entries are of the correct format
- No duplicate entries should bellow
- All links should take the user to the correct page.

7.4 Test Cases

The input of correct and incorrect should be given to check whether model working or not.

CHAPTER 8

SCREENSHOTS

8. SCREENSHOTS

Homepage:

Client Login:

Registration:

Client Send Request:

Client Responses:

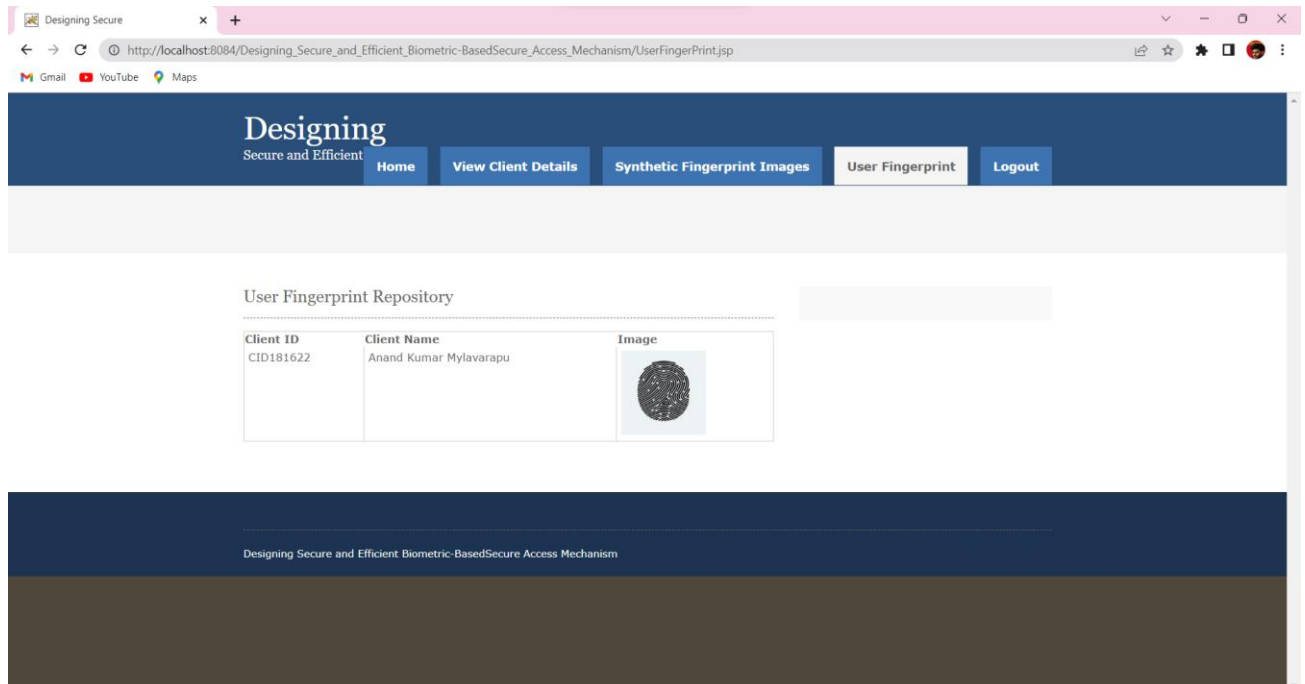Access Resource:
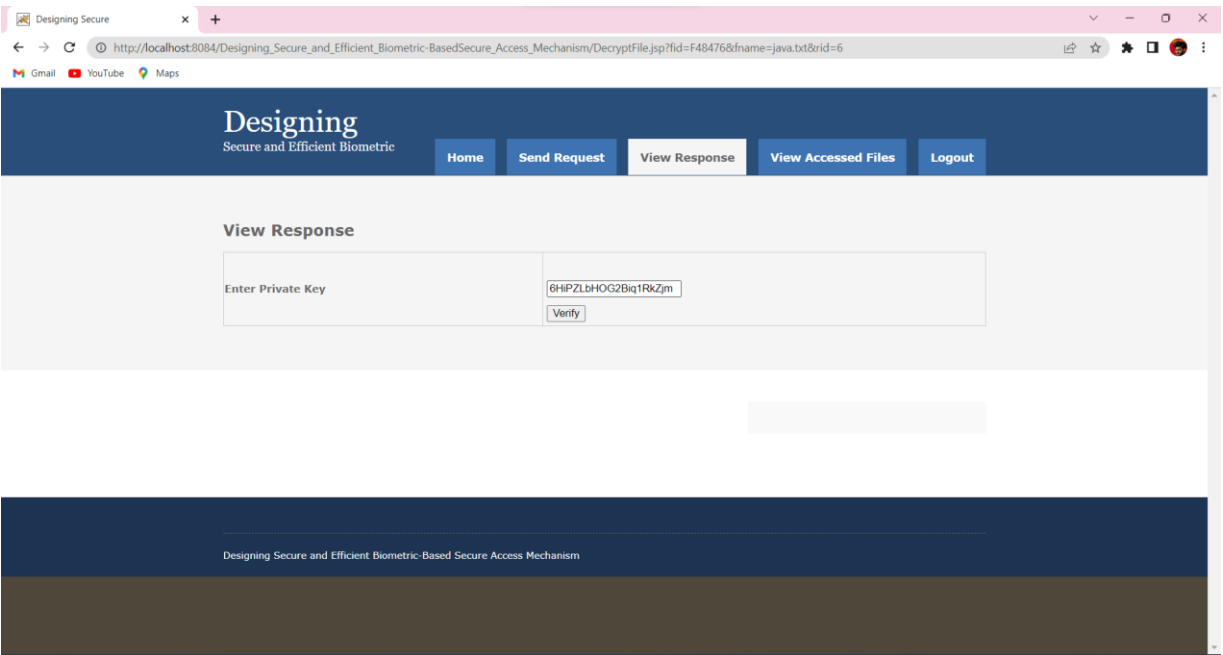
Download File:
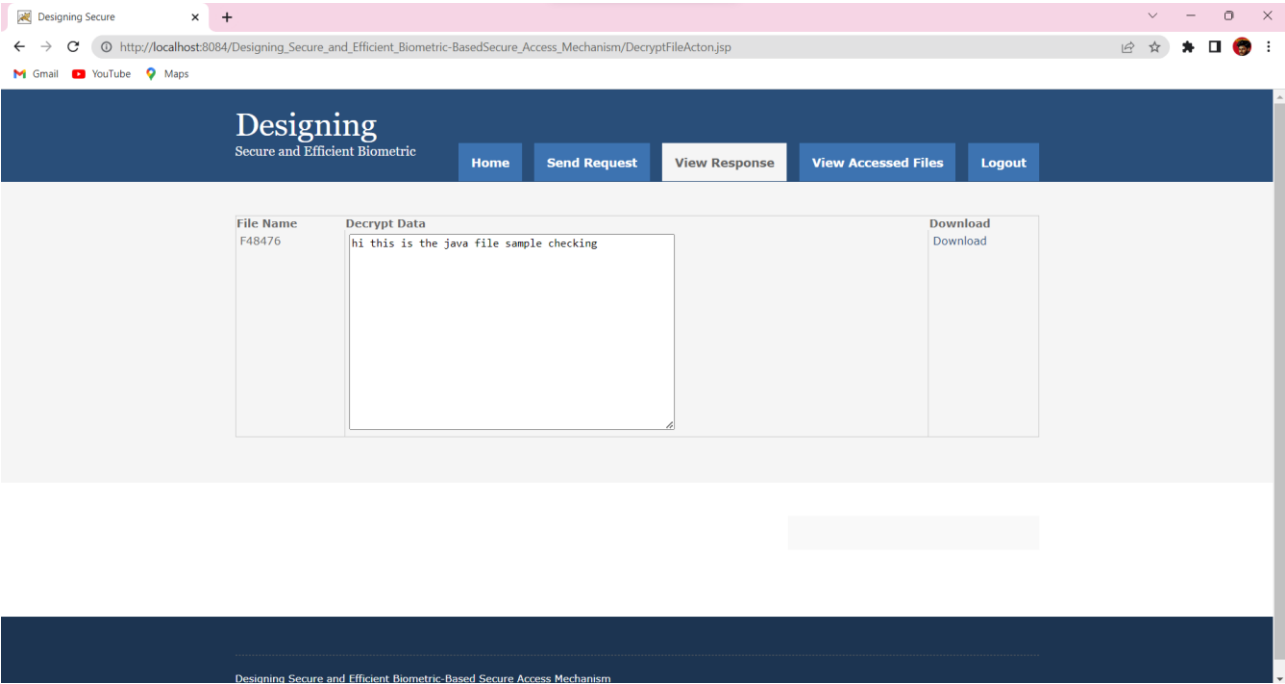
View Accessed Files

Authentication:

Authenticate User:

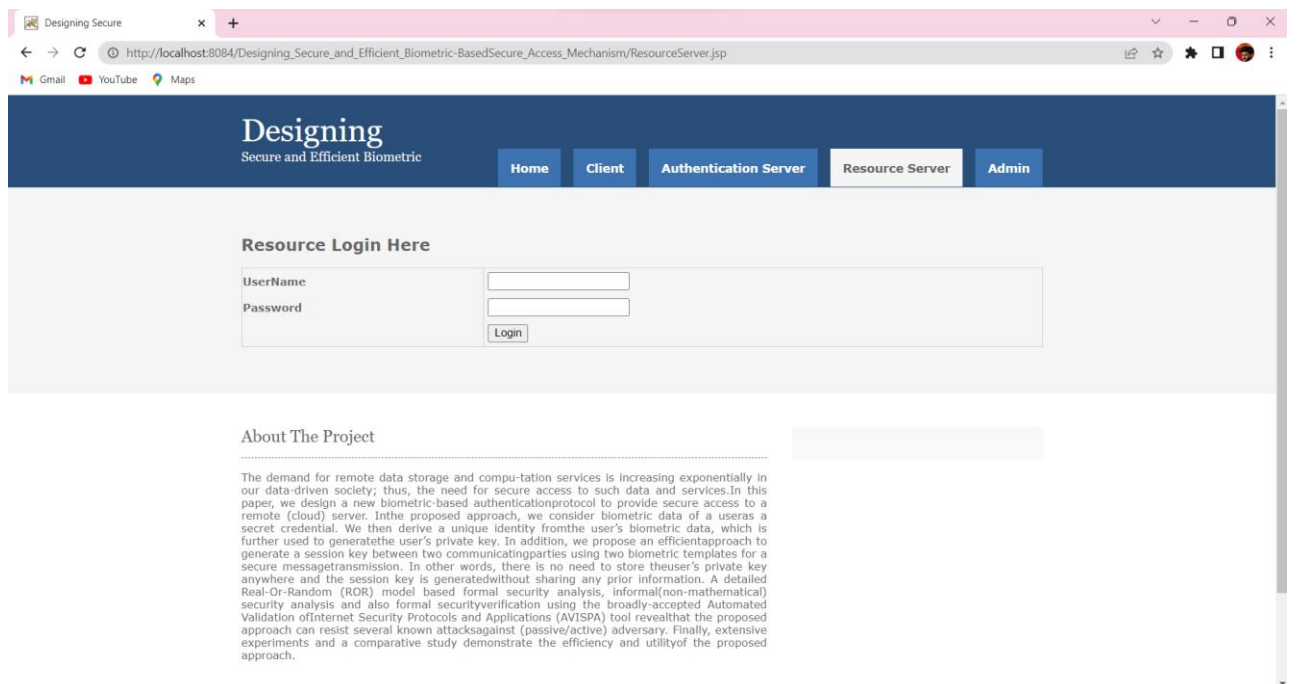Authenticated:
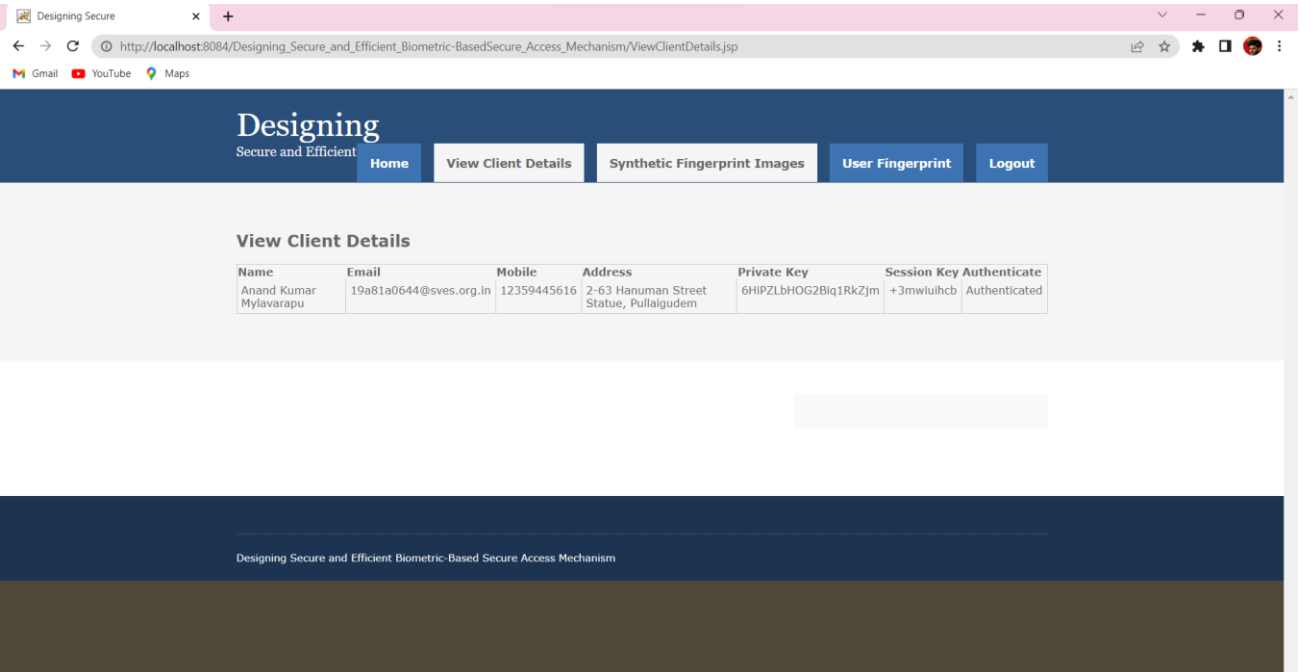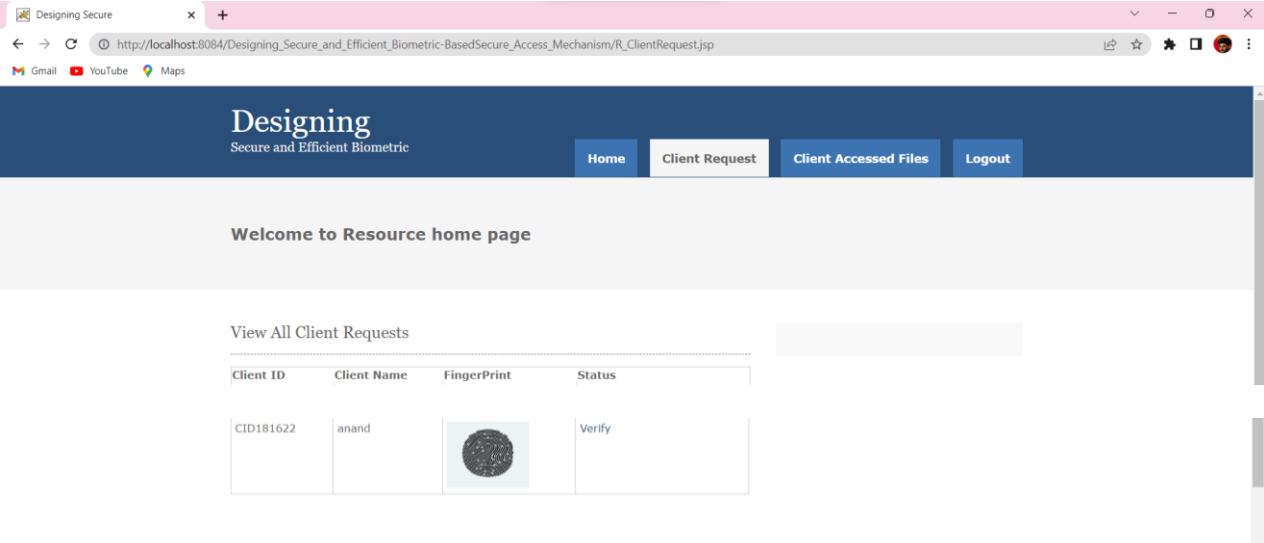
User Fingerprints:

Resource:

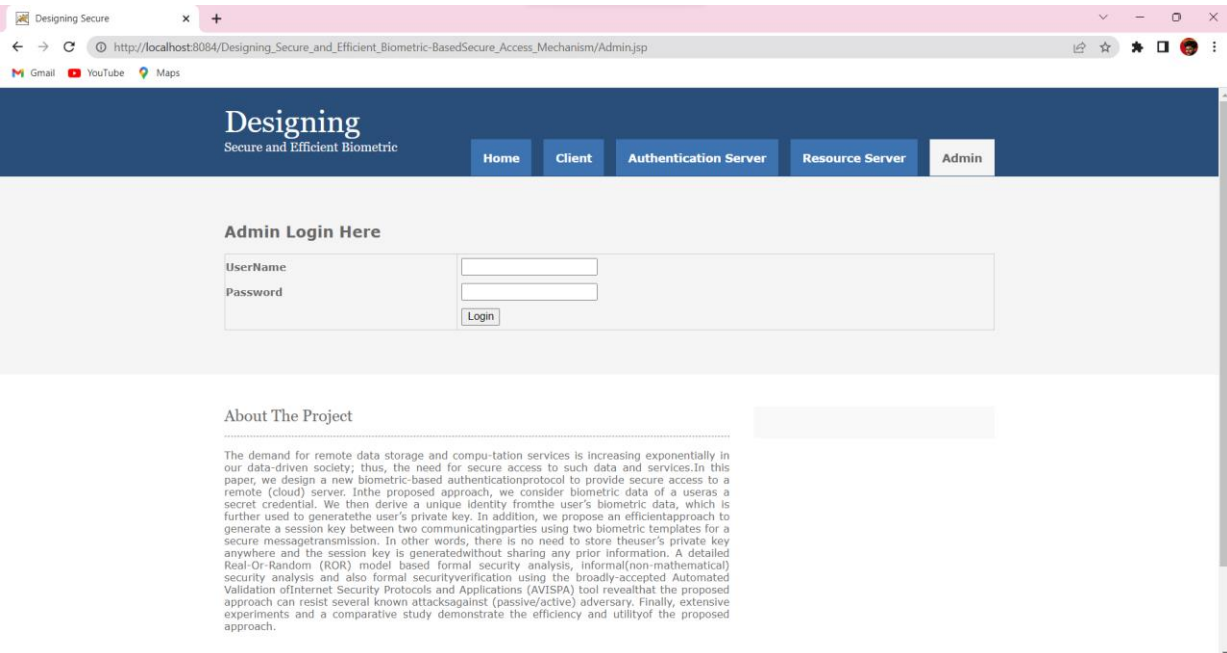## Requests from Client:

Admin:

File Upload:

All Files:

CHAPTER 9

NOMECLATURES

9. NOMECLATURES

- Figure.1: System Architecture

- Figure.2: Use Case Diagram

- Figure.3: Class Diagram

- Figure.4: Sequence Diagram

- Figure.5: Collaboration Diagram

- Figure 6: Activity Diagram

CHAPTER 10

CONCLUSION, FUTURE WORK, REFERENCES

10.a CONCLUSION

In this paper, we proposed an efficient privacy-preserving biometric identification scheme based on matrix transformation. Compared to the existing matrix-transformation-based scheme put forward by Zhu et al. recently, we improve the security of biometric identification by introducing additional randomness. Further, we reduce the computational complexity by exploiting orthogonal matrix, which means our scheme makes the biometric identification more practical for a large-scale database of templates in an actual situation. Our scheme may also benefit other areas, such as privacy-preserving cloud computing.

10.b FUTURE WORK

We can add different types of security features other than biometric, like we add voice, iris scan. To enhance the security from various threats. We can get the finger print live scan so that we can be able to get the authentication fast

10.c REFERENCES

[1] scheme with key distribution for mobile multi-server environment,'' Future Gener. Comput. Syst., vol. 84, pp. 239–251, Jul. 2018.

[2] E. Pagnin and A. Mitrokotsa, ''Privacy-preserving biometric authentication: Challenges and directions,'' Secur. Commun. Netw., vol. 2017, Sep. 2017, Art. no. 7129505.

[3] B. Schneier, ''Biometrics: Uses and abuses,'' Commun. ACM, vol. 42, no. 8, p. 58, 1999.

[4] Wells Fargo Bank. (2019). Convenient Access to Your Accounts. [Online]. Available: https://www.wellsfargo.com/online-banking/biometric/

[5] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, ''Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment,'' IEEE Internet Things J., vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[6] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, ''Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems,'' J. Ambient Intell. Humanized Comput., vol. 9, no. 4, pp. 1061–1073, 2018.

[7] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K. R. Choo, ''A robust biometrics based three-factor authentication scheme for global mobility networks in smart city,'' Future Gener. Comput. Syst., vol. 83, pp. 607–618, Jun. 2018.

[8] D. He, Y. Zhang, and J. Chen, ''Robust biometric-based user authentication scheme for wireless sensor networks,'' Adhoc Sensor Wireless Netw., vol. 25, no. 3, pp. 309–321, 2012.

[9] S. N. Syed, A. Z. Shaikh, and S. Naqvi, ''A novel hybrid biometric electronic voting system: Integrating finger print and face recognition,'' 2018, arXiv:1801.02430. [Online]. Available: https://arxiv.org/abs/1801.02430

[10] C.-A. Toli and B. Preneel, ''Privacy-preserving biometric authentication model for e-finance applications,'' in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 353–360.

[11] M. S. Obaidat, I. Traore, and I. Woungang, Biometric-Based Physical and Cybersecurity Systems. Springer, 2019.

[12] P. Tuyls and J. Goseling, ''Capacity and examples of template-protecting biometric authentication systems,'' in Proc. Int. Workshop Biometric Authentication. Berlin, Germany: Springer, 2004, pp. 158–170.

[13] P. Tuyls, E. Verbitskiy, J. Goseling, and D. Denteneer, ''Privacy protecting biometric authentication systems: An overview,'' in Proc. 12th Eur. Signal Process. Conf., Sep. 2004, pp. 1397–1400.

[14] A. T. B. Jin, D. N. C. Ling, and A. Goh, ''BioHashing: Two factor authentication featuring fingerprint data and tokenised random number,'' Pattern Recognit., vol. 37, no. 11, pp. 2245–2255, Apr. 2004.

[15] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, ''An analysis of biohashing and its variants,'' Pattern Recognit., vol. 39, no. 7, pp. 1359–1368, Jul. 2006.

[16] A. Juels and M. Sudan, ''A fuzzy vault scheme,'' Des., Codes Cryptogr., vol. 38, no. 2, pp. 237–257, Feb. 2006.

[17] W. J. Scheirer and T. E. Boult, ''Cracking fuzzy vaults and biometric encryption,'' in Proc. Biometrics Symp., Sep. 2007, pp. 1–6.

[18] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva, ''Privacy-preserving fingercode authentication,'' in Proc. 12th ACM Workshop Multimedia Secur., 2010, pp. 231–240.

[19] P. Paillier, ''Public-key cryptosystems based on composite degree residuosity classes,'' in Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 1999, pp. 223–238.

[20] D. Catalano and D. Fiore, ''Using linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data,'' in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., 2015, pp. 1518–1529.

[21] J.-H. Im, J. Choi, D. Nyang, and M.-K. Lee, ''Privacy-preserving palm print authentication using homomorphic encryption,'' in Proc. IEEE DASC/PICom/DataCom/CyberSciTec, Aug. 2016, pp. 878–881.

[22] H. Zhu, Q. Wei, X. Yang, R. Lu, and H. Li, ''Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data,'' IEEE Trans. Cloud Comput., to be published. doi: 10.1109/TCC.2018.2866405

[23] D. Boneh, E.-J. Goh, and K. Nissim, ''Evaluating 2-DNF formulas on ciphertexts,'' in Proc. Theory Cryptogr. Conf. Berlin, Germany: Springer, 2005, pp. 325–341.

[24] J. Yuan and S. Yu, ''Efficient privacy-preserving biometric identification in cloud computing,'' in Proc. IEEE INFOCOM, Apr. 2013, pp. 2652–2660.

[25] Y. Zhu, T. Takagi, and R. Hu, ''Security analysis of collusion-resistant nearest neighbor query scheme on encrypted cloud data,'' IEICE Trans. Inf. Syst., vol. 97, no. 2, pp. 326–330, 2014.

[26] L. Zhu, C. Zhang, C. Xu, X. Liu, and C. Huang, ''An efficient and privacypreserving biometric identification scheme in cloud computing,'' IEEE Access, vol. 6, pp. 19025–19033, 2018.

[27] S. Hu, M. Li, Q. Wang, S. S. M. Chow, and M. Du, ''Outsourced biometric identification with privacy,'' IEEE Trans. Inf. Forensics Security, vol. 13, no. 10, pp. 2448–2463, Oct. 2018.

[28] Z. Shan, K. Ren, M. Blanton, and C. Wang, ''Practical secure computation outsourcing: A survey,'' ACM Comput. Surv., vol. 51, no. 2, 2018, Art. no. 31.

[29] H. Delfs and H. Knebl, Introduction to Cryptography, vol. 2. Springer, 2002.

[30] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, ''Filterbankbased fingerprint matching,'' IEEE Trans. Image Process., vol. 9, no. 5, pp. 846–859, May 2000.

[31] T. Bhattasali, K. Saeed, N. Chaki, and R. Chaki, ''A survey of security and privacy issues for biometrics based remote authentication in cloud,'' in Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage. Berlin, Germany: Springer, 2015, pp. 112–121.

[32] K. Liu, C. Giannella, and H. Kargupta, ''An attacker's view of distance preserving maps for privacy preserving data mining,'' in Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery. Berlin, Germany: Springer, 2006, pp. 297–308.

[33] W. K. Wong, D. W.-I. Cheung, B. Kao, and N. Mamoulis, ''Secure KNN computation on encrypted databases,'' in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2009, pp. 139–152.

[34] S. Pan, S. Yan, and W.-T. Zhu, ''Security analysis on privacypreserving cloud aided biometric identification schemes,'' in Proc. Australas. Conf. Inf. Secur. Privacy. Cham, Switzerland: Springer, 2016, pp. 446–453.

[35] J. Katz and Y. Lindell, Introduction to Modern Cryptography. Boca Raton, FL, USA: CRC Press, 2014.