VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI-590018, KARNATAKA

A Project Report on

An Efficient and Privacy Preserving Biometric Identification Scheme in Cloud Computing with Blockchain

Submitted in partial fulfillment of the requirements for the VIII semester of degree of

Bachelor of Engineering

in

Information Science & Engineering

by

Anant Dubey 1RN18IS017 Dinky Asrani 1RN18IS039

Leena Chandra 1RN18IS063 Raashil Aadhyanth 1RN18IS081

Under the Guidance of

Mrs. Sudha V

Asst. Professor,

Dept of ISE, RNSIT

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvardhan Road, Channasandra Rajarajeshwari Nagar Post, Bengaluru-560 098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvardhan Road, Channasandra Rajarajeshwari Nagar Post,
Bengaluru-560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the project work phase 2 entitled ***An efficient and privacy preserving biometric identification scheme in cloud computing with blockchain*** has been successfully completed by **Anant Dubey (1RN18IS017), Dinky Asrani (1RN18IS039), Leena Chandra (1RN18IS063) and Raashil Aadhyanth (1RN18IS081),** bonafide students of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The project phase 2 report has been approved as it satisfies the academic requirements in respect of project phase1 work for the said degree.

Mrs. Sudha V
Project Guide

Dr. Prakasha S / Mrs. Kusuma S
Project Coordinator

Dr. Suresh L
Professor and HOD

Dr. M K Venkatesha
Principal

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

We, **ANANT DUBEY [USN: 1RN18IS017], DINKY ASRANI [USN: 1RN18IS039], LEENA CHANDRA [USN: 1RN18IS063], RAASHIL AADYANTH [USN: 1RN18IS081]** students of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project phase 2 work entitled ***An efficient and privacy preserving biometric identification scheme in cloud computing with blockchain*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belgaum* during academic year 2021-2022.

Place: Bengaluru

Date: 09.07.2022

Anant Dubey	**1RN18IS017**
Dinky Asrani	**1RN18IS039**
Leena Chandra	**1RN18IS063**
Raashil Aadhyanth	**1RN18IS081**

ABSTRACT

Block-chain is a distributed immutable ledger technology. It is consisting of blocks, and each block contains multiple transactions. Block-chain consists of a secure hash, timestamp, data of the current block, and the hash value of the previous block. Block- chain records all transactions across the network so that it cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network.

Biometric identification system usage has increased as it provides an auspicious way to identify users. While compared with traditional authentication methods Biometric is more reliable and convenient. Block-chain was designed initially to solve double spending problems in peer-to-peer payment system of Bitcoin.

But, since then its applications gone through the original intended use because of its properties, i.e., decentralization, immutability, no trusted authority, and auditability. In this paper it proposes how this Block-chain technology can have applied in biometric identification scheme like Aadhar in cloud server-to make Aadhar more transparent.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this mini-project work.

We would like to express my thanks to our Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

We wish to place on record my words of gratitude to **Dr. Suresh L,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my mini-project work.

We would also like to thank our project guide **Mrs. Sudha V,** Assistant Professor, Department of ISE, RNSIT, Bangalore, for her valuable suggestions and guidance.

We place our thanks to project coordinators **Dr. Prakasha S**, Associate Professor and **Mrs. Kusuma S**, Assistant Professor, ISE, RNSIT for their timely guidelines and suggestions for carrying out the project work successfully.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

Place: Bengaluru
Date: 09-07-2022

Anant Dubey	**1RN18IS07**
Dinky Asrani	**1RN18IS039**
Leena Chandra	**1RN18IS063**
Raashil Aadhyanth	**1RN18IS081**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CNN	Convolution Neural Network
GLCM	Grey Level Co-Occurrence Matrix
DCNN	Deep Convolution Neural Network
ReLu	Rectified Linear Unit
ANN	Artificial Neural Network
CAD	Computer Aided Diagnosis
DSB	Data Science Bowl
KNN	K nearest neighbor
AES	Advance Encryption Standard
SHA-1	Secure Hash Algorithm

Chapter 1

INTRODUCTION

Biometric identification has raised increasingly attention since it provides a promising way to identify users. Compared with traditional authentication methods based on passwords and identification cards, biometric identification is more reliable and convenient.

1.1 Introduction to Biometric

Biometric identification has been widely applied in many fields by using biometric traits such as fingerprint, iris, and facial patterns, which can be collected from various sensors. In a biometric identification system, the database owner such as the FBI who is responsible to manage the national fingerprints database, may desire to outsource the enormous biometric data to the cloud server to get rid of the expensive storage and computation costs.

However, to preserve the privacy of biometric data, the biometric data must be encrypted before outsourcing. Whenever a FBI's partner (e.g., the police station) wants to authenticate an individual's identity, user turns to the FBI and generates an identification query by using the individual's biometric traits (e.g., fingerprints, irises, voice patterns, facial patterns etc.). Then, the FBI encrypts the query and submits it to the cloud to find the close match. Thus, the challenging problem is how to design a protocol which enables efficient and privacy- preserving biometric identification in the cloud computing. Several privacy-preserving biometric identification solutions have been proposed.

1.2 Existing System

The existing system provides less security. During the identification process, the privacy of biometric data should not be protected. Attackers and the semi-honest cloud should learn all about the sensitive information, there is no security to protect the data. The main problem here is designing a protocol, which helps us to store the data into the cloud. Aadhar is a largest unique identification system. It records 1.19 billion Indian resident's records. Aadhar authentication is a procedure wherein Aadhar number along with the personnel attributes including biometrics are given to government. By using this Aadhar

government agency can track the login patterns and access the personnel data. Using Blockchain technology in biometric system like Aadhar to make it more transparent to the public.

Aadhar can broadcast all the modifications against each user record into the blockchain. In this paper the problem is solved by using Blockchain technology, everyone can know the changes against their Aadhar record.

1.2 Problem Statement

In July 2018, Telecom Regulatory Authority of India (TRAI) Chairman R.S Sharma posted his Aadhaar card number in twitter and challenged Aadhaar critics to hack if they can. Within just seven hours, the ethical hackers posted screenshot of sending re.1 to Sharma's bank account via the Aadhaar enabled service and also, they posted 14 details of his, which included Sharma's mobile no, DOB, residential address, phone no, PAN no, Bank details, etc.

As of now Aadhaar card data are not safely stored in cloud by the Govt of India, which creates a huge security problem in preserving privacy. In the present work the problem is the government data are directly sent to the cloud service provider & the assumption is cloud service provider is encrypting & storing the data is not used any block chain & other thing. But it is already proved that Aadhaar card details are hack able it is hacked &it is already released in the net. So that the old system is unsecured.

Biometric Aadhar system using block- chain every user no need to reveal the personnel data or Aadhar number. In this system data will be stored by using hash and this data recorded into the block-chain. So that any changes happening against the user Aadhar record and immediately question the authorities

1.4 Proposed System

In the proposed system the application of blockchain technology to Aadhar system will make the system robust and trustworthy blockchain based Aadhar system is much secured. How it is much secured means that Aadhaar card details are encrypted in a

government server by using AES algorithm alongwithSHA1hashingtechnique.Only and the encrypted data only sent to the cloud service provider and in this cloud service provider also is giving the second level of security by using the Block chain so that it is highly secure with the previous work.

The project proposes an efficient and privacy preserving biometric identification scheme which can resist the security attack done by the users and the cloud. Examine the biometric identification scheme and show its insufficiency and security weakness under the proposed level-3 attack. Specifically, it is demonstrated that the attacker can recover their secret keys by colluding with the cloud, and then decrypt the biometric traits of all users.

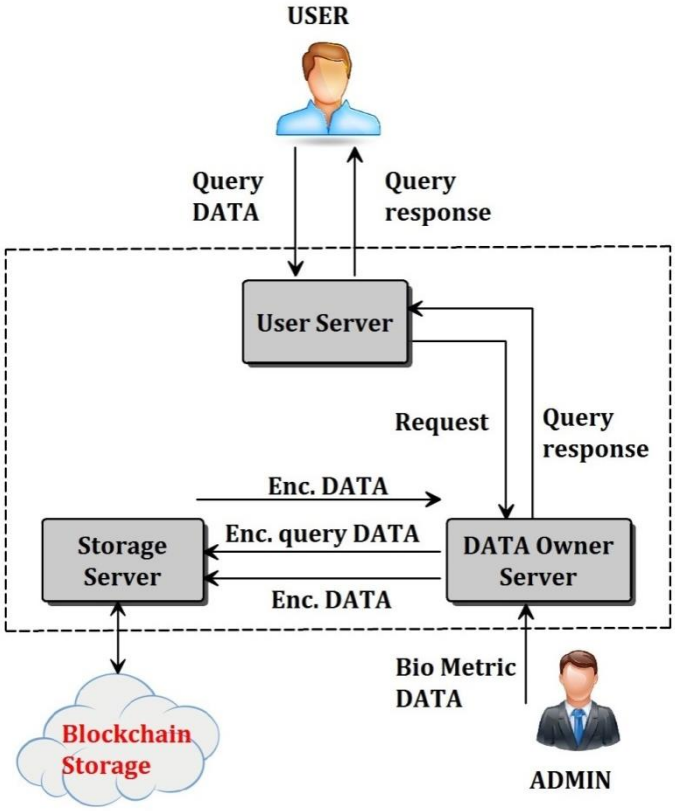

Figure 1.1 Proposed System Model

As shown in Figure 1.1, three types of entities are involved in the system such as the database owner, users, and the cloud. The database owner holds a large size of biometric data which is encrypted and transmitted to the cloud for storage and then stored in the blockchain. When a user wants to identify themself, a query request is sent to the database owner. After receiving the request, the database owner generates a ciphertext for the biometric trait and then transmits the ciphertext to the cloud for identification.

Finally, the database owner computes the similarity between the query data and the biometric data associated with the index and returns the query result to the user.

Advantage of Proposed System:

- More security with Block chain storage
- Reduce workload and enhance productivity
- Better flexibility and speed
- Data is secure for users
- Increasing trust as peer-to- peer network
- Data access by authenticated user
- Efficiency: Computational costs should be as low as possible at both the database owner side and the user side. To gain high efficiency, most biometric identification operations should be executed in the cloud.
- Security: During the identification process, the privacy of biometric data should be protected. Attackers and the semi-honest cloud should learn nothing about the sensitive information.

Chapter 2

LITERATURE REVIEW

A literature survey or a literature review in a project report shows the various analyses and research made in the field of interest and the results already published, considering the various parameters of the project and the extent of the project.

Literature survey is mainly carried out to analyze the background of the current project which helps to find out flaws in the existing system & guides on which unsolved problems to work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

A literature survey includes the following:

- Existing theories about the topic which are accepted universally.
- Books written on the topic, both generic and specific.
- Research done in the field usually in the order of oldest to latest.
- Challenges being faced and on-going work, if available.

Literature survey describes about the existing work on the given project. It deals with the problem associated with the existing system and also gives user a clear knowledge on how to deal with the existing problems and how to provide solution to the existing problems.

2.1 Objectives of Literature Survey

- Learning the definitions of the concepts.
- Access to latest approaches, methods and theories.
- Discovering research topics based on the existing research
- Concentrate on own field of expertise– Even if another field uses the same words, they usually mean complete

Biological characteristics [12, 20] under ideal conditions should have the following properties:

1. Universality: Ideally, all people have this biological characteristic.
2. Uniqueness: In all populations, each person's biological characteristics are different.
3. Stability: Ideally, the biological characteristics are immutable, or there is a smaller one within the class Change.
4. Collectability: Under certain conditions, the biological characteristics can be accurately obtained.

There are many studies on biometric identity authentication, mainly focusing on two directions [11]:

A. Based on the traditional biometric identification technology: Users submit their own biometric data, and compare them with the data stored in biometric template, the matching degree of the two determines whether they are legitimate users and whether they

B. Key generation technology based on biometrics: This model based on keystroke dynamics, extracts binary string from user keystroke mode.

Early privacy-preserving biometric identification schemes only focus on the privacy-preserving issue. In these schemes, the biometric identification scheme is considered to be a two-party system, where the data owner takes charge of biometric dataset management and template matching. Most of these schemes are designed based on the secure computation protocol [18–20] and homomorphic encryption [9,21,22] techniques.

Although the privacy-preserving is achieved in these schemes, the data owner is required to be equipped with powerful computing ability and remarkable storage capacity in these schemes, which can hardly be satisfied in most application scenarios and thus makes these schemes unpractical combines it with user password to form a stronger password; Hao and Chan design a key generation system

Few of the popular methodologies used over the years include:

Jain, L. Hong, and S. Pankanti [1], Multi biometric systems utilize the evidence presented by multiple biometric sources (e.g., face and fingerprint, multiple fingers of a user, multiple matchers, etc.) to determine or verify the identity of an individual. Information from multiple sources can be consolidated in several distinct levels, including the feature extraction level, match score level and decision level.

While fusion at the match score and decision levels have been extensively studied in the literature, fusion at the feature level is a relatively understudied problem. In this paper discussion of fusion at the feature level in 3 different scenarios: (i) fusion of PCA and LDA coefficients of face; (ii) fusion of LDA coefficients corresponding to the R,G,B channels of a face image; (iii) fusion of face and hand modalities. Preliminary results are encouraging and help in highlighting the pros and cons of performing fusion at this level. The primary motivation of this work is to demonstrate the viability of such a fusion and to underscore the importance of pursuing further research in this direction.

R. Allen, P. Sankar, and S. Prabhakar [2] A new method for biometric identification of human irises is proposed in this paper. The method is based on morphological image processing for the identification of unique skeletons of iris structures, which are then used for feature extraction. In this approach, local iris features are represented by the most stable nodes, branches and endpoints extracted from the identified skeletons. Assessment of the proposed method was done using subsets of images from the University of Bath Iris Image Database (1000 images) and the CASIA Iris Image Database (500 images). Compelling experimental results demonstrate the viability of using the proposed morphological approach for iris recognition when compared to a state-of-the-art algorithm that uses a global feature extraction approach.

The privacy preserving models for attack is introduced at first. An overview of several anonymity operations follow behind. The most important part is the coverage of anonymity algorithms and information metric which is essential ingredient of algorithms. The conclusion and perspective are proposed finally.

J. de Mira, Jr., H. V. Neto, E. B. Neves, and F. K. Schneider [3], this paper presents a novel algorithm aiming at analysis and identification of faces viewed from different poses and illumination conditions. Face analysis from a single image is performed by recovering the shape and textures parameters of a 3D Morphable Model in an analysis-by-synthesis fashion. The shape parameters are computed from a shape error estimated by optical flow and the texture parameters are obtained from a texture error.

The algorithm uses linear equations to recover the shape and texture parameters irrespective of pose and lighting conditions of the face image. Identification experiments are reported on more than 5000 images from the publicly available CMU-PIE database which includes faces viewed from 13 different poses and under 22 different illuminations. Extensive identification results are available on web page for future comparison with novel algorithms. S. Romdhani, V. Blanz, and T. Vetter [4]**,** Biometric identification is a reliable and convenient way of identifying individuals. The widespread adoption of biometric identification requires solid privacy protection against possible misuse, loss, or theft of biometric data. Existing techniques for privacy-preserving biometric identification primarily rely on conventional cryptographic primitives such as homomorphic encryption and oblivious transfer, which inevitably introduce tremendous cost to the system and are not applicable to practical large-scale applications.

In this paper, a novel privacy-preserving biometric identification scheme is proposed which achieves efficiency by exploiting the power of cloud computing. In proposed scheme, the biometric database is encrypted and outsourced to the cloud servers. To perform a biometric identification, the database owner generates a credential for the candidate biometric trait and submits it to the cloud. The cloud servers perform identification over the encrypted database using the credential and return the result to the owner. Personalized recommendation is crucial to help users find pertinent information. It often relies on a large collection of user data, in particular users' online activity (e.g., tagging/rating/checking-in) on social media, to mine user preference. However, releasing such user activity data makes users vulnerable to inference attacks, as (e.g., gender) private data can often be inferred from the users' activity data

During the identification, cloud learns nothing about the original private biometric data. Because the identification operations are securely outsourced to the cloud, the real time computational/communication costs at the owner side are minimal. Thorough analysis shows that proposed scheme is secure and offers a higher level of privacy protection than related solutions such as kNN search in encrypted databases. Real experiments on Amazon cloud, over databases of different sizes, show that computational/communication costs at the owner side are several magnitudes lower than the existing biometric identification schemes.

Y. Xiao et al. [5], with identity fraud in society reaching unprecedented proportions and with an increasing emphasis on the emerging automatic personal identification applications, biometrics-based verification, especially fingerprint-based identification, is receiving a lot of attention. There are two major shortcomings of the traditional approaches to fingerprint representation. For a considerable fraction of population, the representations based on explicit detection of complete ridge structures in the fingerprint are difficult to extract automatically.

The widely used minutiae-based representation does not utilize a significant component of the rich discriminatory information available in the fingerprints. Local ridge structures cannot be completely characterized by minutiae. Further, minutiae-based matching has difficulty in quickly matching two fingerprint images containing different number of unregistered minutiae points. The proposed filter-based algorithm uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode.

The fingerprint matching is based on the Euclidean distance between the two corresponding FingerCodes and hence is extremely fast. The method can achieve a verification accuracy which is only marginally inferior to the best results of minutiae-based algorithms published in the open literature [1]. system performs better than a state-of-the-art minutiae-based system when the performance requirement of the application system does not demand a very low false acceptance rate. Finally, it is shown that the matching performance can be improved by combining the decisions

of the matchers based on complementary (minutiae-based and filter-based) fingerprint information.

X. Du, Y. Xiao, M. Guizani, and H.-H. Chen [6], service providers like Google and Amazon are moving into the SaaS (Software as a Service) business. They turn their huge infrastructure into a cloud-computing environment and aggressively recruit businesses to run applications on their platforms.

To enforce security and privacy on such a service model, need to protect the data running on the platform. Unfortunately, traditional. Encryption methods that aim at providing "unbreakable" protection are often not adequate because they do not support the execution of applications such as database queries on the encrypted data. In this paper, discussion of the general problem of secure computation on an encrypted database and propose a SCONEDB (Secure Computation ON an Encrypted Database) model, which captures the execution and security requirements of the system currently at use and provides sufficient result.

As a case study, focus on the problem of k-nearest neighbor (kNN) computation on an encrypted database. Develop a new asymmetric scalar-product-preserving encryption (ASPE) that preserves a special type of scalar product. Usage pf APSE to construct two secure schemes that support kNN computation on encrypted data; each of these schemes is shown to resist practical attacks of a different background knowledge level, at a different overhead cost. Extensive performance studies are carried out to evaluate the efficiency of the system being used

X. Du and H. H. Chen **[7]** This paper introduces a SCiFI, system for Secure Computation of Face Identification. The system performs face identification which compares faces of subjects with a database of registered faces. The identification is done in a secure way which protects both the privacy of the subjects and the confidentiality of the database A specific application of SCiFI is reducing the privacy impact of camera-based surveillance. This paper provides an overview of the development of privacy preserving data publishing, which is restricted to the scope of anonymity algorithms using generalization and suppression.

In that scenario, SCiFI would be used in a setting which contains a server which has a set of faces of suspects, and client machines which might be cameras acquiring images in public places. The system runs a secure computation of a face recognition algorithm, which identifies if an image acquired by a client matches one of the suspects, but otherwise reveals no information to neither of the parties.

This paper presents a novel approach for face based biometric recognition. The proposed method is based on the sorted index numbers (SIN) of appearance based facial features. A new algorithm is introduced to measure the similarity between SIN vectors. Due to the non-invertibility of the transformation from the original features to the SIN vectors, the proposed method can preserve the privacy of the users.

The effectiveness of the proposed method is tested on a large generic data set, which contains images from several well-known face databases. Experimental results demonstrate that the proposed solution may improve the recognition accuracy in both identification and verification scenarios X. Hei, X. Du, J. Wu, and F. Hu [9] A novel algorithm aiming at analysis and identification of faces viewed from different poses and illumination conditions. Face analysis from a single image is performed by recovering the shape and textures parameters of a 3D Morphable Model in an analysis-by-synthesis fashion.

The shape parameters are computed from a shape error estimated by optical flow and the texture parameters are obtained from a texture error. The algorithm uses linear equations to recover the shape and texture parameters irrespective of pose and lighting conditions of the face image. Identification experiments are reported on more than 5000 images from the publicly available CMU-PIE database which includes faces viewed from 13 different poses and under 22 illuminations

M. Barni et al. [10], Examine the effectiveness of distance preserving transformations in privacy preserving data mining. These techniques are potentially very useful in that some important data mining algorithms can be efficiently applied to the transformed data and produce exactly the same results as if applied to the original data e.g., distance-based clustering, k-nearest neighbor classification.

However, the issue of how well the original data is hidden has, to knowledge, not been carefully studied. Take a step in this direction by assuming the role of an attacker armed with two types of prior information regarding the original data. Examine how well the attacker can recover the original data from the transformed data and prior information. Results offer insight into the vulnerabilities of distance preserving transformations.

In all the cloud-based schemes above, the searching process is not optimized, which means that the searching cost of the cloud server is linear with the size of the dataset. Even though the cloud server is equipped with strong computing power, it may still run into a bottleneck while simultaneously severing too many users. To address this issue, some researchers begin to focus on how to achieve sublinear searching efficiency in the biometric identification process, which will significantly ease the pressure of the cloud server.

Zhu et al. [11] proposed a cloud-assisted privacy-preserving biometric identification scheme. With the help of an asymmetric scalar-product preserving encryption scheme and R-tree, sublinear search efficiency is achieved in [11]. Nevertheless, the data owner also needs to keep online in [11]. And since R-tree is not constructed among the metric relation between the data objects, the cloud server needs to search the tree twice to find the closest biometric template in the dataset, which reduces the efficiency of the searching process. Yang et al. [26] proposed a privacy- preserving biometric identification scheme based on the M-tree to achieve a sublinear search efficiency.

Information sharing as an indispensable part appears in the vision, bringing about a mass of discussions about methods and techniques of privacy preserving data publishing which are regarded as strong guarantee to avoid information disclosure and protect individuals' privacy. Recent work focuses on proposing different anonymity algorithms for varying data publishing scenarios to satisfy privacy requirements and keep data utility at the same time. K-anonymity has been proposed for privacy preserving data publishing, which can prevent linkage attacks by the means of anonymity operation, such as generalization and suppression. Numerous anonymity algorithms have been utilized for achieving k-anonymity

In this paper, to protect the privacy of the biometric data and reduce the time consumption in the biometric searching process, introducing Fingerprint algorithm and SHA-1 to construct a privacy-preserving biometric identification scheme based on two non-colluded cloud servers. Compared with previous works, the service provider in proposed scheme does not need to keep online in the identification scheme, and higher efficiency in both computation and communication is achieved.

Chapter 3

ANALYSIS

Project Analysis means work done before the legislative appropriation for a project to develop a reliable estimate of the cost of the project to be used in the appropriations request.

3.1 Problem Identification

Problem identification is part of the scientific method, as it serves as the first step in a systematic process to identify, evaluate a problem and explore potential solutions. Problem identification consists of two steps: identifying and acknowledging that a discrepancy exists (i.e., identifying that there is a problem), and developing a problem identification statement.

3.1.1 Challenges of Cloud Computing

Since LIVING in a digital age, where data discovery and big data simply surpass the traditional storage and manual implementation and manipulation of business information, companies are searching for the best possible solution of handling data.

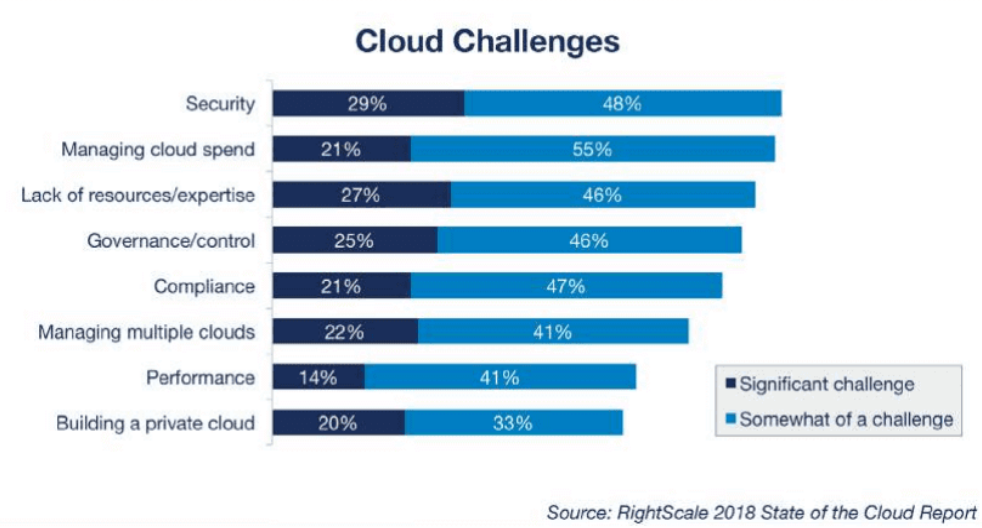

Figure 3.1 Challenges of cloud storage

Traditional spreadsheets no longer serve their purpose, there is just too much data to store, manage and analyze. Be it in the form of online BI tools, or an online data visualization system, a company must address where and how to store its data.

Even the most traditional sectors have to adjust. Though the opportunities are great, this explosion hasn't come without issues in cloud computing.

Security and spend are the top challenges. Security is a challenge for 77 percent of respondents, while 29 percent see it as a significant challenge. Managing cloud spend is a challenge for 76 percent of respondents, while a smaller 21 percent see it as a significant challenge. As companies become more experienced with cloud, the top challenge shifts. Security is the largest issue among cloud beginners, while cost becomes a bigger challenge for intermediate and advanced users.

3.1.2 Security Issues

For the longest time, the lack of resources/expertise was the number one voiced cloud challenge. In 2018 however, security inched ahead. Security has indeed been a primary, and valid, concern from the start of cloud computing technology. This increases the cloud computing risks that can arise during the implementation or management of the cloud.

Cybersecurity experts are even more concerned about cloud security than other IT staffers are. A 2018 Crowd Research Partners survey found that 90 percent of security professionals are concerned about cloud security. More specifically, they have fears about data loss and leakage (67 percent), data privacy (61 percent) and breaches of confidentiality (53 percent).

Headlines highlighting data breaches, compromised credentials, and broken authentication, hacked interfaces and APIs, account hijacking haven't helped alleviate concerns. All of this makes trusting sensitive and proprietary data to a third party hard to stomach for some and, indeed, highlighting the challenges of cloud computing.

Luckily as cloud providers and users, mature, security capabilities are constantly improving. In this paper, an efficient and privacy preserving biometric identification scheme is proposed which can resist the collusion attack launched by the users and the cloud.

3.2 Objectives

Several privacy-preserving biometric identification solutions have been proposed. However, most of them mainly concentrate on privacy preservation but ignore the efficiency, such as the schemes based on homomorphic encryption and oblivious transfer in for fingerprint and face image identification respectively. Suffering from performance problems of local devices, these schemes are not efficient once the size of the database is larger than 10 MB

Compared with other methods, using biometrics as identity can save a lot of space and time. Moreover, because of the uniqueness of individual biometrics, the key extracted from biometrics is also unique. Biological characteristics are remarkable characteristics that distinguish one from others, such as face shape, fingerprint, palm shape, voice, iris, infrared heat and other congenital characteristics are physiological characteristics. Biological characteristics also have the advantages of safety, confidentiality, convenience, good anti-counterfeiting performance, not easy to create or be stolen, and portable and readily available.

An efficient and privacy preserving biometric identification scheme is proposed which can resist the collusion attack launched by the users and the cloud. Specifically, main contributions can be summarized as follows:

- Presenting a novel efficient and privacy-preserving biometric identification scheme.
- The detailed security analysis shows that the proposed scheme can achieve a required level of privacy protection.
- Specifically, scheme is secure under the biometric identification outsourcing model and can also resist the attack proposed by an external hacker.
- Compared with the existing biometric identification schemes, the performance analysis shows that the proposed scheme provides a lower computational cost in both preparation and identification procedure

3.3 Methodology

It is a set of ideas or guidelines about how to proceed in gathering and data for

validating knowledge of a subject matter. Different areas of science have developed very different bodies of methodology based on which to conduct their research.

1) AES: stands for Advanced Encryption Standard and is a majorly used symmetric encryption algorithm. It is mainly used for encryption and protection of electronic data. It was used as the replacement of DES (Data encryption standard) as it is much faster and better than DES. AES consists of three block ciphers and these ciphers are used to provide encryption of data.

Points to remember

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

AES data encryption is a more mathematically efficient and elegant cryptographic algorithm, but its main strength rests in the option for various key lengths. AES allows you to choose a 128-bit, 192-bit or 256-bit key, making it exponentially stronger than the 56-bit key of DES.

So due to obvious reasons, AES is more advanced than 3DES. Its 128-bit keys provide ample strength. And these keys can be implemented in both software and hardware without any hassle

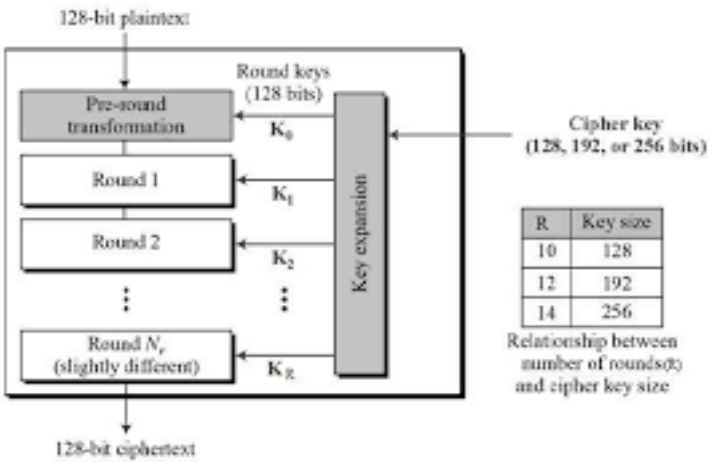

Figure 3.2 be Cipher text working

.

2) Secure Hash Algorithms: also known as SHA, are a family of cryptographic functions designed to keep data secured. It works by transforming the data using a hash function:
an algorithm that consists of bitwise operations, modular additions, and compression functions.

- SHA-1 works by feeding a message as a bit string of length less than 2^26 bits, and producing
- A 160-bit hash value known as a *message digest*.
- The hash function then produces a fixed-size string that looks nothing like the original.

These algorithms are designed to be one-way functions, meaning that once they're transformed into their respective hash values, it's virtually impossible to transform them back into the original data.

A common application of SHA is to encrypting passwords, as the server side only needs to keep track of a specific user's hash value, rather than the actual password. This is helpful in case an attacker hacks the database.

They will only find the hashed functions and not the actual passwords, so if they were to input the hashed value as a password, the hash function will convert it into another string and subsequently deny access.

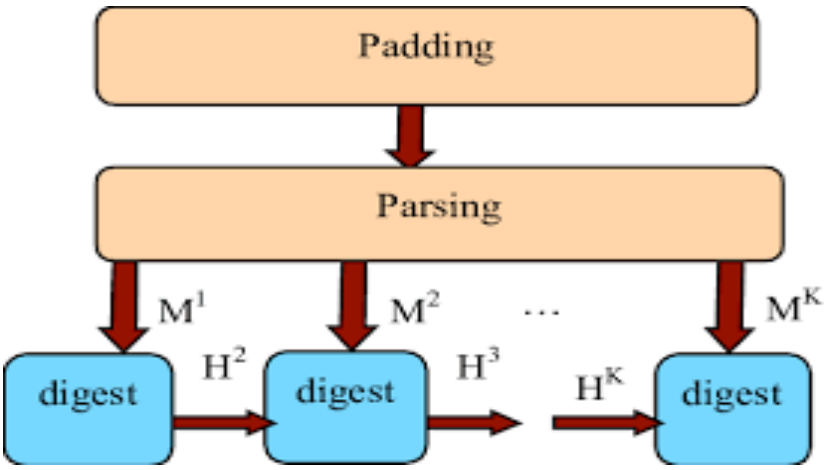

Figure 3.3 Flow diagram of Sha-1 algorithm

3) Fingerprint algorithm: Human fingerprints are defined by tiny ridges, spirals and valley patterns on the tip of each finger. They are unique: no two people have the same ones.

Similarly, a fingerprinting algorithm in computer science is one that maps large data (such as documents or images) to a much shorter sequence of bytes. Such a sequence may be called the data's fingerprint.

This fingerprint may be used for data deduplication(data deduplication is a technique for eliminating duplicate copies of repeating data. Successful implementation of the technique can improve storage utilization,) purposes.

Fingerprint functions may be seen as high-performance hash functions used to uniquely identify substantial blocks of data where cryptographic hash functions may be unnecessary

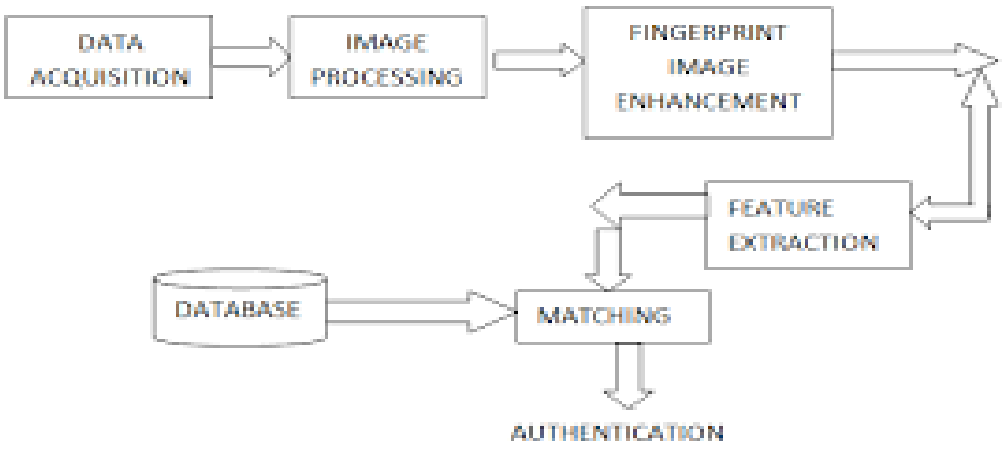

Figure 3.4 Architecture of matching in fingerprint algorithm

What are some use cases of fingerprinting algorithms? When large filesystems or databases need to be searched, fingerprints are used to speed up data retrieval. Rather than compare large amounts of content, it's sufficient to compare their fingerprints. When files are uploaded, cloud providers use fingerprints to do deduplication so that multiple copies of the same file are avoided.

4) Cloud storage: is a cloud computing model that stores data on the Internet through a cloud computing provider who manages and operates data storage as a service. It’s.

delivered on demand with just-in-time capacity and costs, and eliminates buying and managing own data storage infrastructure. This gives you agility, global scale and durability, with “anytime, anywhere” data access.

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offering.

There are three types of cloud data storage: object storage, file storage, and block storage. Each offers their own advantages and own specific functions

1. Object Storage - Applications developed in the cloud often take advantage of object storage's vast scalability and metadata characteristics. Object storage solutions like Amazon Simple Storage Service (S3) are ideal for building modern applications from scratch

2. File Storage - Some applications need to access shared files and require a file system. This type of storage is often supported with a Network Attached Storage (NAS) server. File storage solutions like Amazon Elastic File System (EFS) are ideal for use cases like large content repositories,

3. Block Storage - Other enterprise applications like databases or ERP systems often require dedicated, low latency storage for each host. This is analogous to direct-attached storage (DAS) or a Storage Area Network (SAN).

Figure 3.5 Cloud Storage

3.4 Functional Requirements

- In this project, there are two types of actors such as an admin and the user.
- Admin has to login using the adminid and admin password, so that admin can login and go to the admin home page.
- Admin has to upload the datasheet like Aadhaarid, Aadhaar fingerprint image and some basic information regarding the user, have to be filled in this datasheet.
- While uploading this, system has to generate the signature of Aadhaarid using SHA algorithm then for fingerprint image this system has to extract the features using fingerprint algorithm.
- The extracted fingerprint feature and signature of Aadhaarid both has to concatenate and the concatenated value has to be encrypted using AES algorithm then store it in the cloud platform with the generated block id.
- User has to login using their username name and password sent to their email id, so that user can login and visit the home page.
- Once the valid user has logged in, then user will be able to send Aadhaar request to the Admin along with user's Aadhaar no. sent to the email id of the user, Aadhaar image and Aadhaar fingerprint image.
- So that the admin person can receive the request from users.
- Admin has to download the Aadhaarid from the cloud, which done in the backend. The downloaded data has to be decrypted using AES algorithm.
- Then admin has to verify the downloaded Aadhaarid & Fingerprint with the requested Aadhaarid & fingerprint, only if these are matching then only admin will provide the details of Aadhaar.
- So User can retrieve the full details of Aadhaar.

3.5 Non-Functional Requirements

Usability

Simple is the key here. The system must be simple that people like to use it, but not so complex that people avoid using it. The user must be familiar with the user interfaces and should not have problems in migrating to a new system with a new environment. The menus, buttons and dialog boxes should be named in a manner that they provide

clear understanding of the functionality. Several users are going to use the system simultaneously, so the usability of the system should not get affected with respect to individual users.

Flexibility

The system should be flexible enough to allow modifications at any point of time.

Reliability

The system should be trustworthy and reliable in providing the functionalities. Once a user has made some changes, the changes must be made visible by the system. The changes made by the Programmer should be visible both to the Project leader as well as the Test engineer.

Performance

The system is going to be used by many employees simultaneously. Since the system will be hosted on a single web server with a single database server in the background, performance becomes a major concern. The system should not succumb when many users would be using it simultaneously. It should allow fast accessibility to all of its users.

Scalability

The system should be scalable enough to add new functionalities at a later stage. There should be a common channel, which can accommodate the new functionalities.

Maintainability

The system monitoring and maintenance should be simple and objective in its approach. There should not be too many jobs running on different machines such that it gets difficult to monitor whether the jobs are running without errors.

Portability

The system should be easily portable to another system. This is required when the web server, which s hosting the system gets stuck due to some problems, which requires the system to be taken to another system.

Reusability

The system should be divided into such modules that it could be used as a part of another system without requiring much of work.

3.6 HARDWARE REQUIREMENTS:

System	:	i3 core processor or above
Hard Disk	:	500 GB
RAM	:	8 GB

3.7 SOFTWARE REQUIREMENTS:

Operating system	:	Windows 8 or above
Coding Language	:	Java (Jdk 1.7)
Web Technology	:	Servlet, JSP
Web Server	:	TomCAT 6.0
IDE	:	Eclipse Indigo
Database	:	My-SQL 5.0
UGI for DB	:	SQLyog
JDBC Connection	:	Type 4 Driver

Chapter 4

SYSTEM DESIGN

System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. The purpose is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture

4.1 System Architecture

System Architecture identifies the overall hypermedia structure for the WebApp. Architecture design is tied to the goals establish for a WebApp, the content to be presented, the users who will visit, and the navigation philosophy that has been established. Content architecture focuses on the manner in which content objects and structured for presentation and navigation. WebApp architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. WebApp architecture is defined within the context of the development environment in which the application is to be implemented.

4.2. Flow Chart Diagram

It is important to complete all tasks and meet deadlines. There are many project management tools that are available to help project managers manage their tasks and schedule and one of them is the flowchart.

A flowchart is one of the seven basic quality tools used in project management and it displays the actions that are necessary to meet the goals of a particular task in the most practical sequence. Also called as process maps, this type of tool displays a series of steps with branching possibilities that depict one or more inputs and transforms them to outputs.

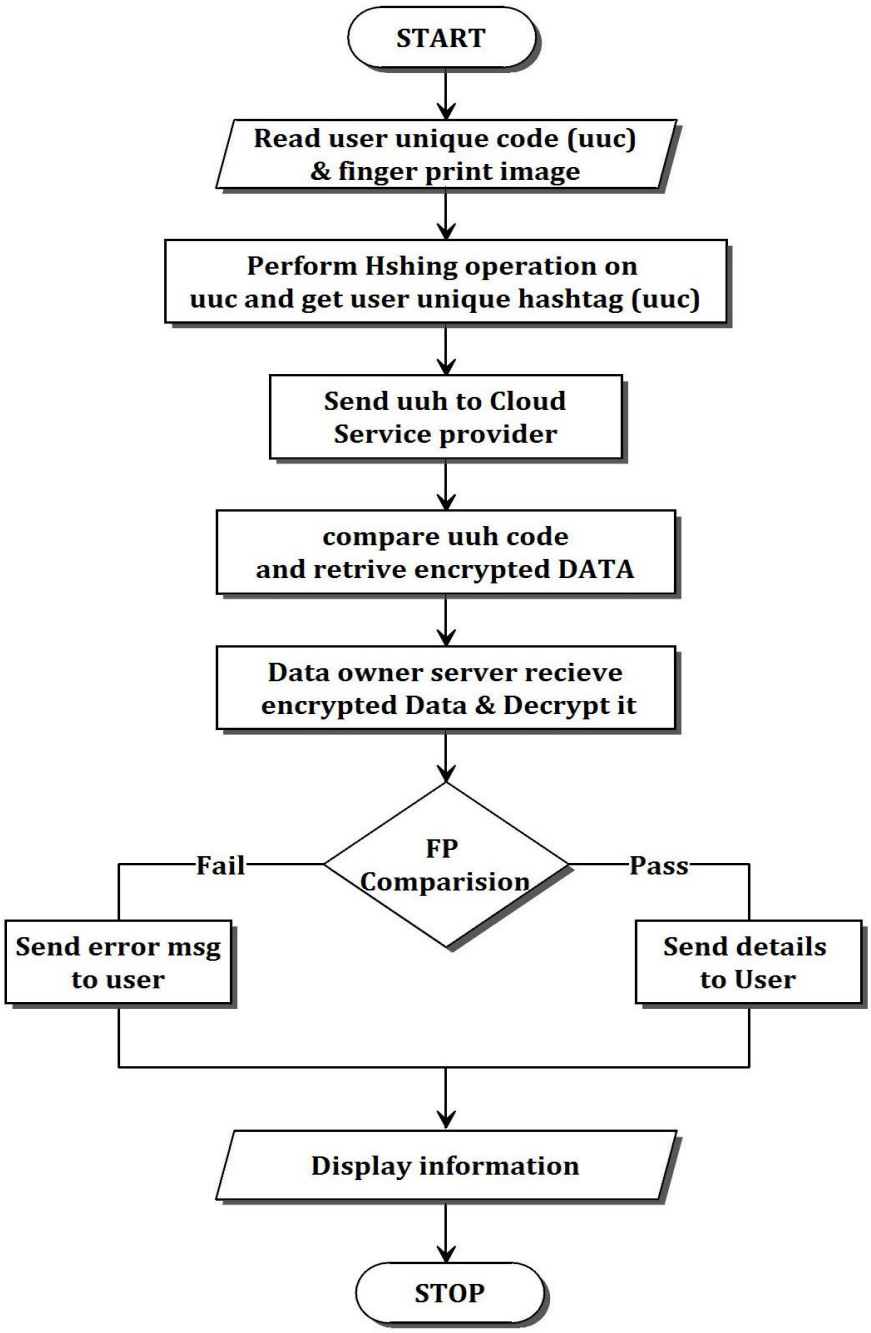

Figure 4.1 Flow Chart Diagram

4.3 Use Case Diagrams

A use case is a set of scenarios that describing an interaction between a source and a destination. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. shows the use case diagram

4.3.1 Use Case Diagram of Data owner

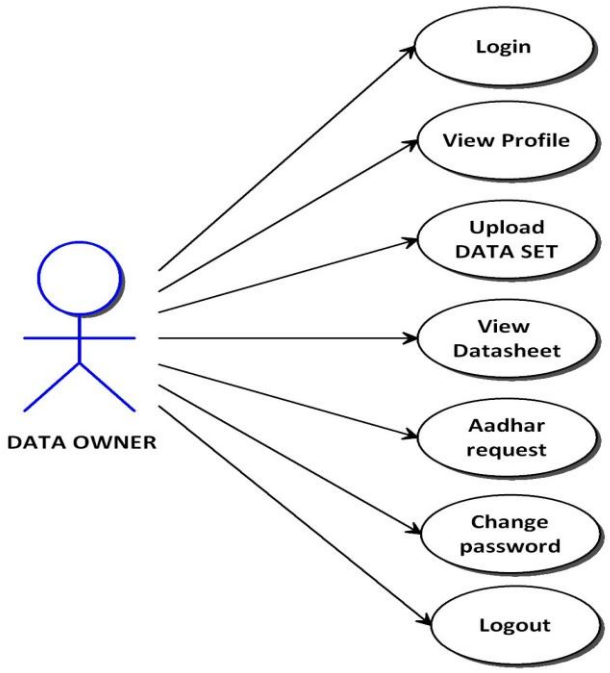

Figure 4.2 Use Case Diagram for Data owner

4.3.2 Use Case Diagram of User

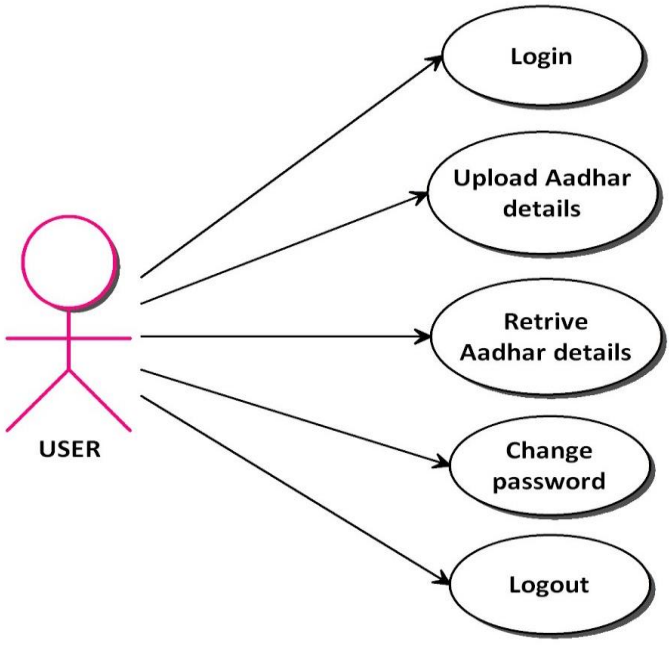

Figure 4.3 Use Case Diagram for User

4.4 Data Flow Diagram

1. A data flow diagram (DFD) is graphic representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context level DFD first which shows the interaction between the system and outside entities. DFD's show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols:
2. Squares representing external entities, which are sources and destinations of information entering and leaving the system.
3. Rounded rectangles representing processes, in other methodologies, may be called 'Activities', 'Actions', 'Procedures', 'Subsystems' etc. which take data as input, do processing to it, and output it.
4. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store except via a process, and external entities are not allowed to access data stores directly.
5. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.

4.4.1 Level 0 Data Flow Diagram

The Level0 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. Figure shows the level 0 DFD.

The process will do by the Intermediate nodes. Finally the requested data will be delivered to the requesting node. It represents the overall process in the simple and short procedure. Here there are only to nodes Source which used transmit the data packet to the respective node and destination which are being used to receive the packet and gain the required dat

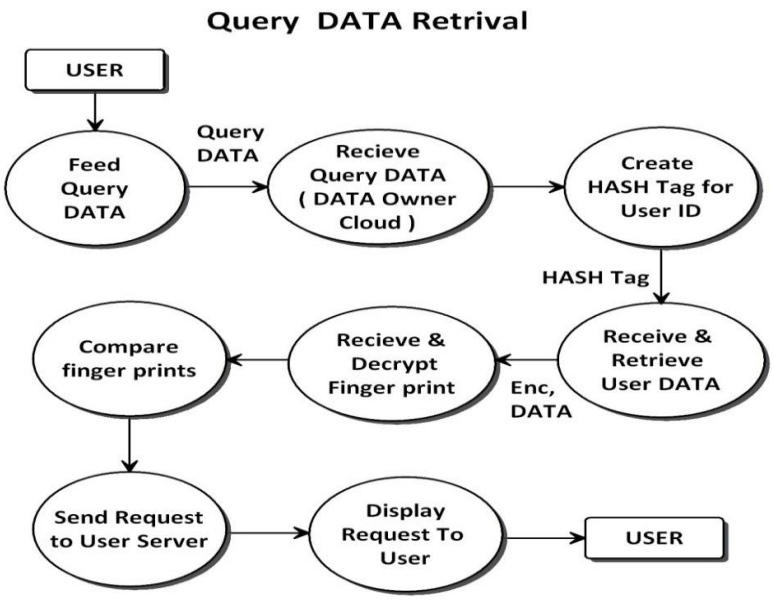

Figure 4.4 Level 0 Data Flow diagram

4.4.2 Level 1 Data Flow Diagram

In the level1Data Flow Diagram, select a file and transfer that file to server. Server receives all the details and generates a Message Digest, once MD file is generated it retrieves all the public key belongs to the user group i.e.(MD+Public key) generates a secure MD , Encrypt secure MD with user Private keys and generates Ring-Signature and send a mail to all users.

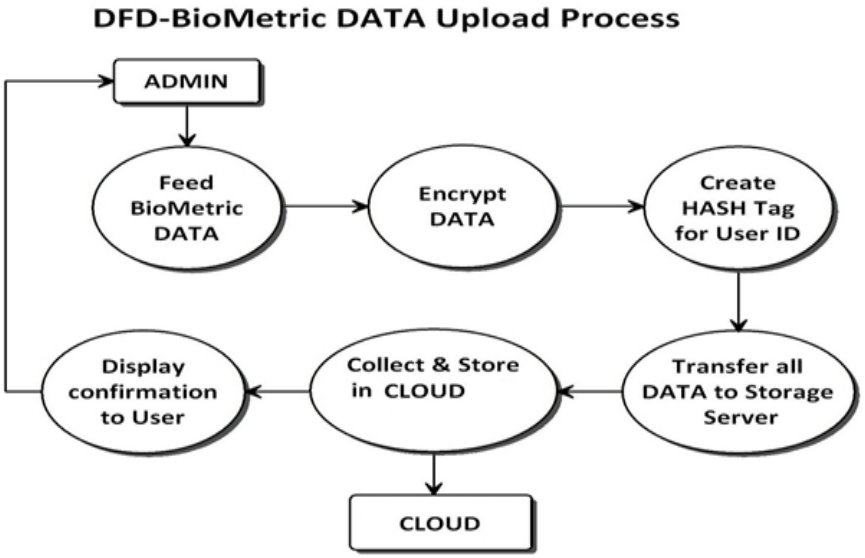

Figure 4.5 Level 1 Data Flow Diagram

4.5 Sequence Diagram

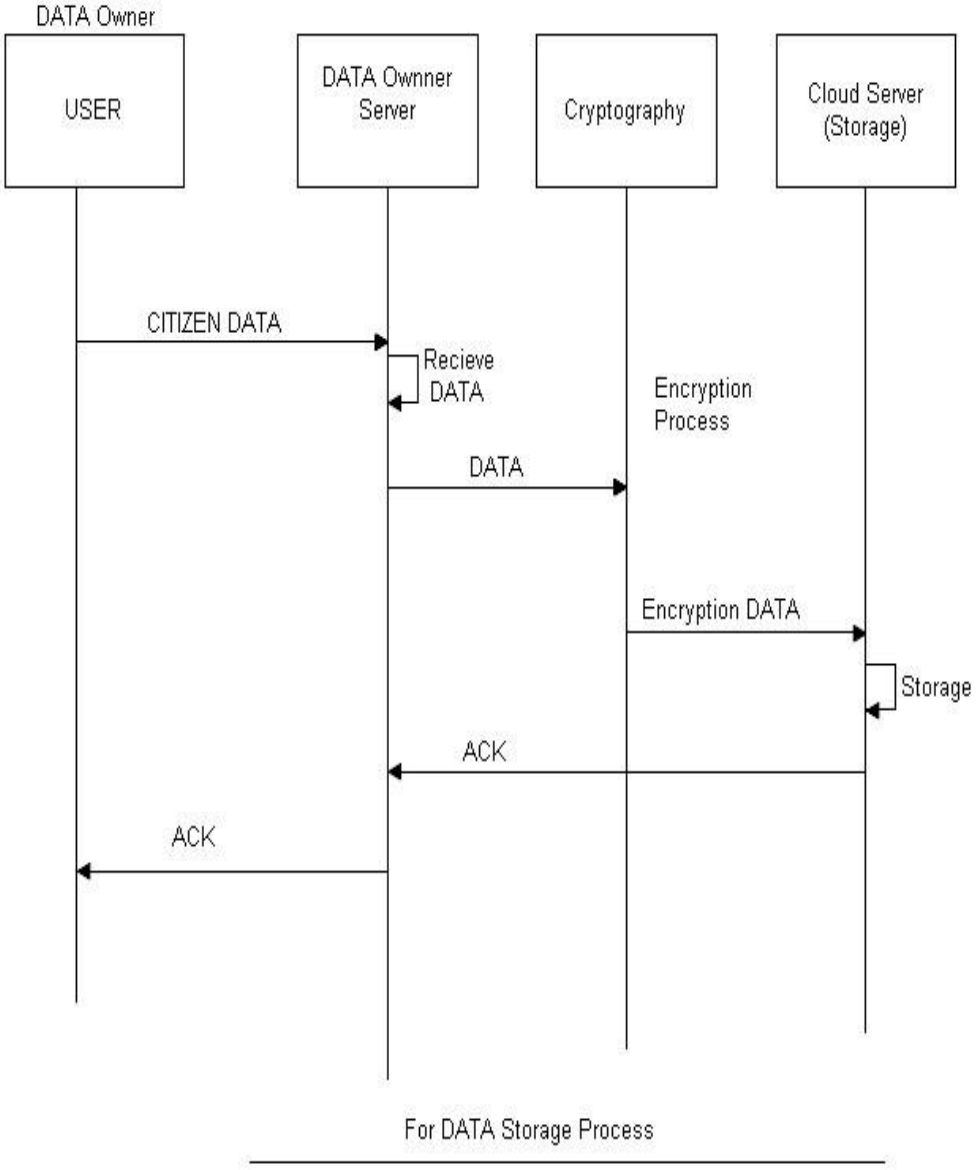

Figure 4.6 Sequence diagram of Data storage process

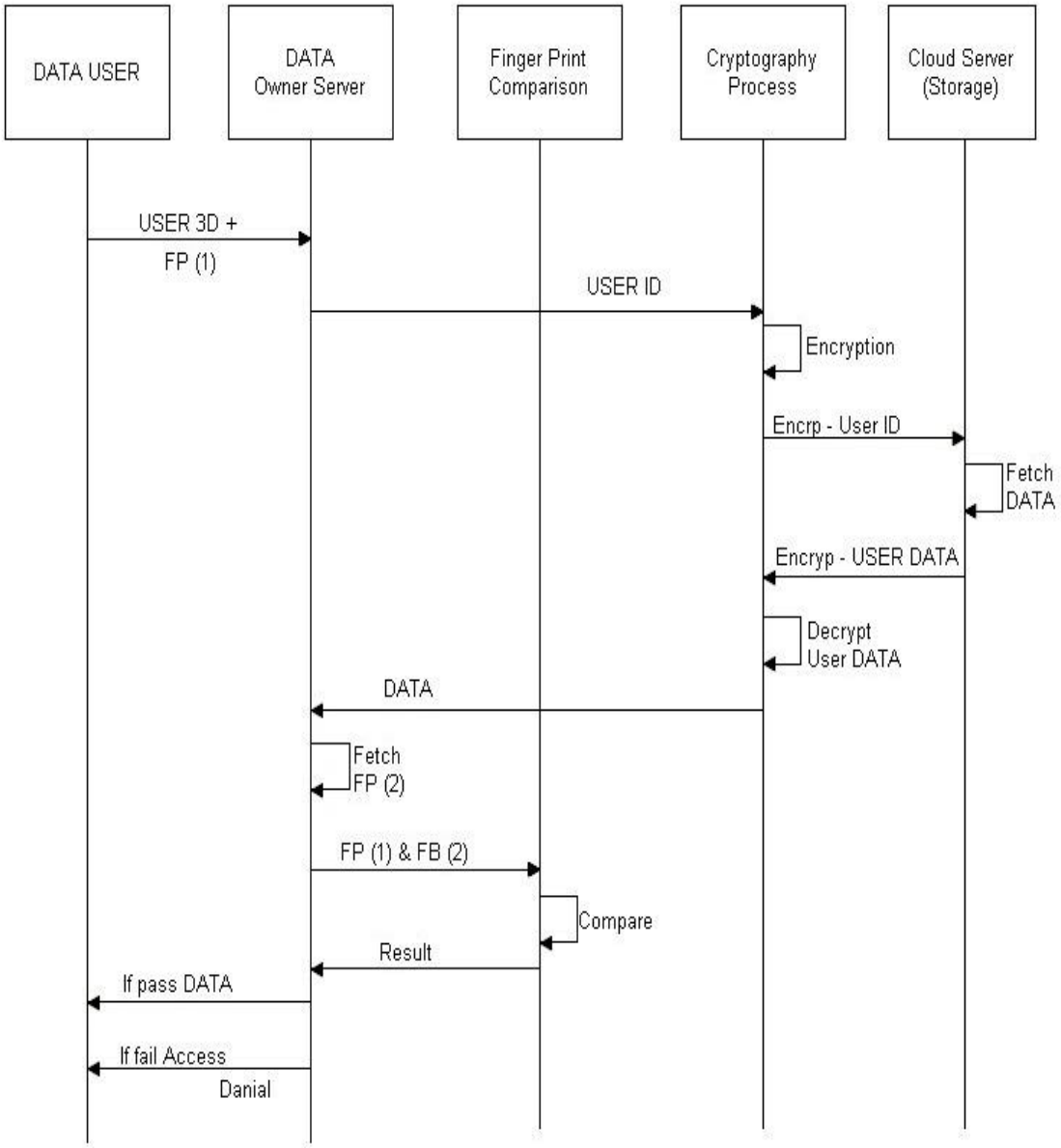

Figure 4.7 Sequence diagram Data Retrieval

4.6 Class Diagram

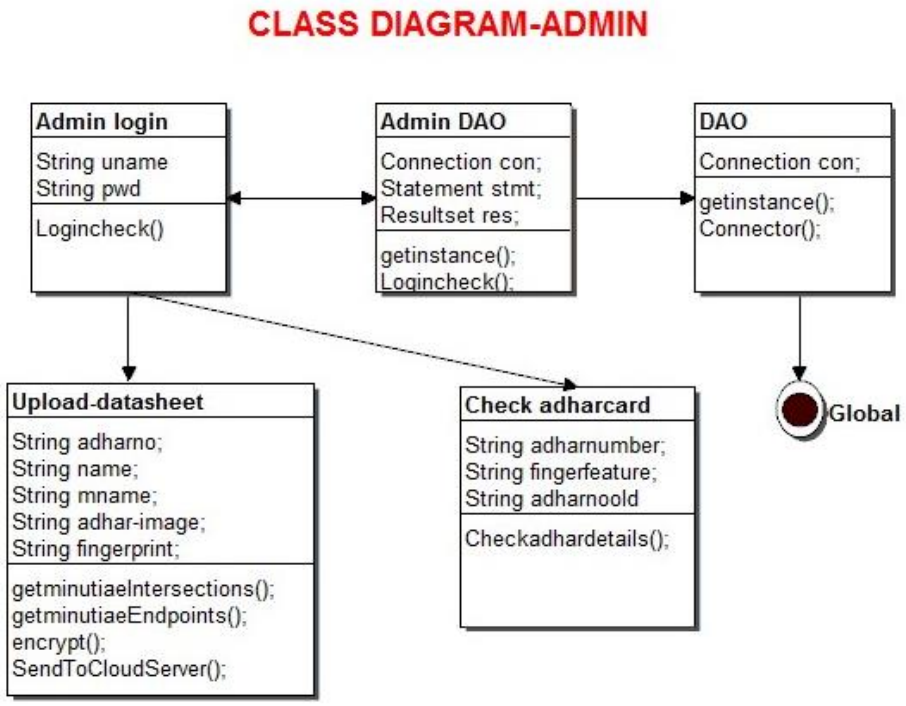

Fig 4.8 Class diagram for Admin

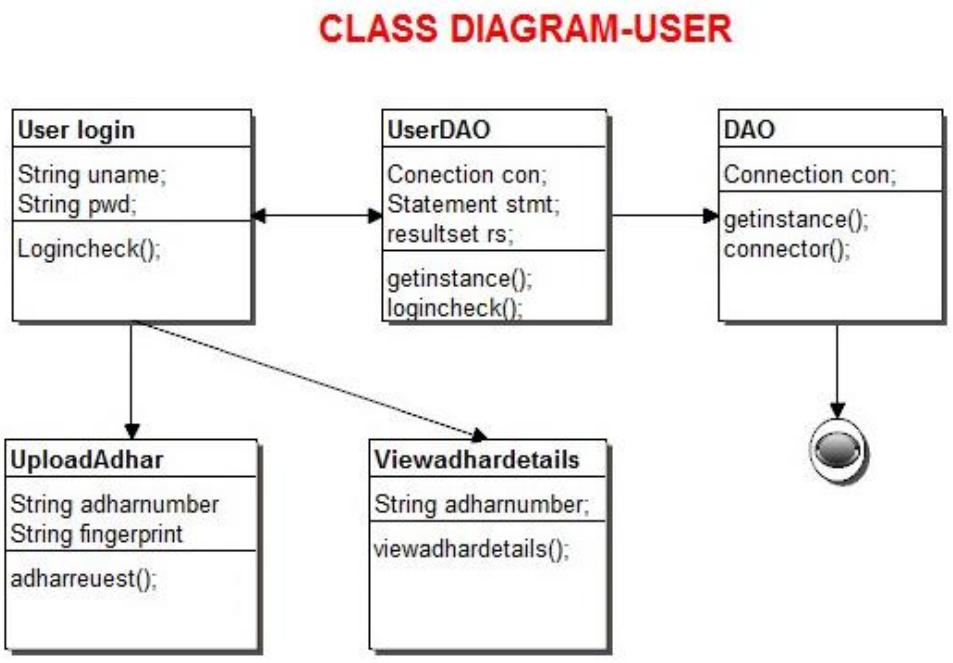

Figure 4.9 Class diagram for User

4.7 Modules Description

1. Admin (Data Owner)

- The Admin is responsible for uploading the Aadhaar card details into the cloud in a secure manner. The Aadhaar id and the fingerprint is encrypted and then stored in the cloud server in the blockchain format.
- The Aadhaar card number and the password would be sent to the registered user's email id, once the datasheet is uploaded successfully.
- Whenever dataowner is uploading the query data, the Aadhaar no. and the fingerprint's encrypted features will get encrypted by using the AES algorithm. So, query data will be in a secure manner.
- To secure the encrypted file having the encrypted data such as extracted features, this system is generating the hash tag by using SHA1 algorithm. Which is helpful to know the data whether it's been modified or corrupted.
- From data owner server to cloud storage server communication usage of the web service concept. So Data owner has to feed the query data, those data has to get encrypted and that encrypted data such as the SHA file and the confidential key file generated having the previous hash code, current hash code, timestamp and nonce have to be stored in the blockchain zip file with the blockid in the cloud storage.

2. User

- The User has to login with his username and password sent via the email-id. The User has to give their Aadhaar number sent through the email and upload the fingerprint so the request will be sent to the dataowner for retrieving the Aadhaar card details.
- The Dataowner has to verify if the adhar id is correct or not with the help of cloud server data, after the verification dataowner is sending the query data to the User. So according to the adharid the user will receive the query data.
- There will be a comparison happening with the Dataowner finger print feature and user uploaded fingerprint's encrypted features considering the old hash code and the new hash code, only if its matching the query data retrieval process would be successful.

Chapter 5

SYSTEM IMPLEMENTATION

The implementation phase involves more than just writing code. Code also needs to be tested and debugged as well as compiled and built into a complete executable product. Usually need to utilize configuration management in order to keep track of different version of code. This is the stage of the project where the theoretical design is turned into a working system.

If the implementation is not carefully planned and controlled, it can cause chaos and confusions. It is always a good idea to keep in mind that some characteristics that should be found in a good implementation like Readability- code is written in MVC Architecture, JAVA to achieve the objective of the project that is to introduce a novel scheme of mechanism design for balancing the resource consumptions .
Implementation stage requires the following tasks:

- Careful planning
- Investigation of system and constraints
- Design of methods to achieve the changeover
- Evaluation of the changeover method
- Correct decisions regarding selection of the platform
- Appropriate selection of the language for application development
- Java Technology is both a programming language and a platform.

5.1 JAVA

Java is a computer-programming language that is concurrent ,class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA),

Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

In 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

5.2 J2EE

The Java EE stands for Java Enterprise Edition, which was earlier known as J2EE and is currently known as Jakarta EE. It is a set of specifications wrapping around Java SE (Standard Edition). The Java EE provides a platform for developers with enterprise features such as distributed computing and web services. Java EE applications are usually run on reference run times such as microservers or application servers. Examples of some contexts where Java EE is used are e-commerce, accounting, banking information systems. The J2EE technologies consists of Servlets, Java Server Pages (JSP), Java Database Connectivity (JDBC), Enterprise Java Bean (EJB), Java Message Service (JMS) etc. In the project, using the MVC architecture containing Servlets, JSP and JDBC technologies.

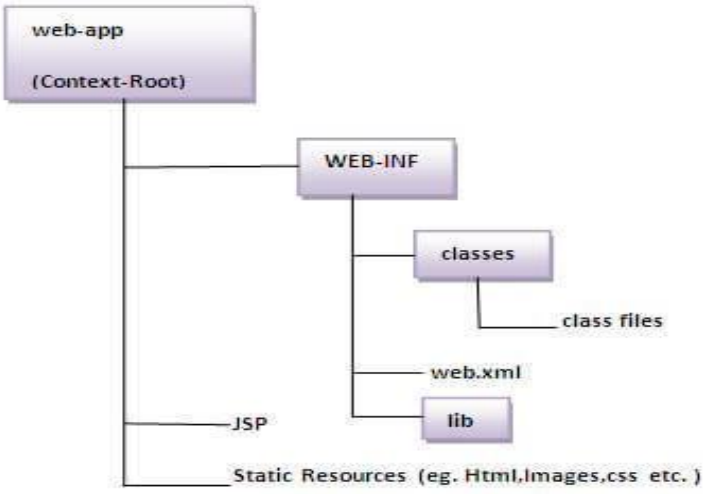

Figure 5.1 Directory structure of the web application

5.2.1 MVC Architecture

The Model-View-Controller (MVC) is a well-known design pattern in the web development field. It is way to organize code. It specifies that a program or application shall consist of data model, presentation information and control information. The MVC pattern needs all these components to be separated as different

objects. This project developed as web based application which is having model view controller.

The model designs based on the MVC architecture follow MVC design pattern. The application logic is separated from the user interface while designing the software using model designs. The MVC pattern architecture consists of three layers:

- Model: It represents the business layer of application. It is an object to carry the data that can also contain the logic to update controller if data is changed. In the project, Model is the .java files.
- View: It represents the presentation layer of application. It is used to visualize the data that the model contains. In the project, View is the .jsp files which giving the user interface of the application
- Controller: It works on both the model and view. It is used to manage the flow of application, i.e. data flow in the model object and to update the view whenever data is changed. In the project, Controller is the web.xml file which is controlling the web application

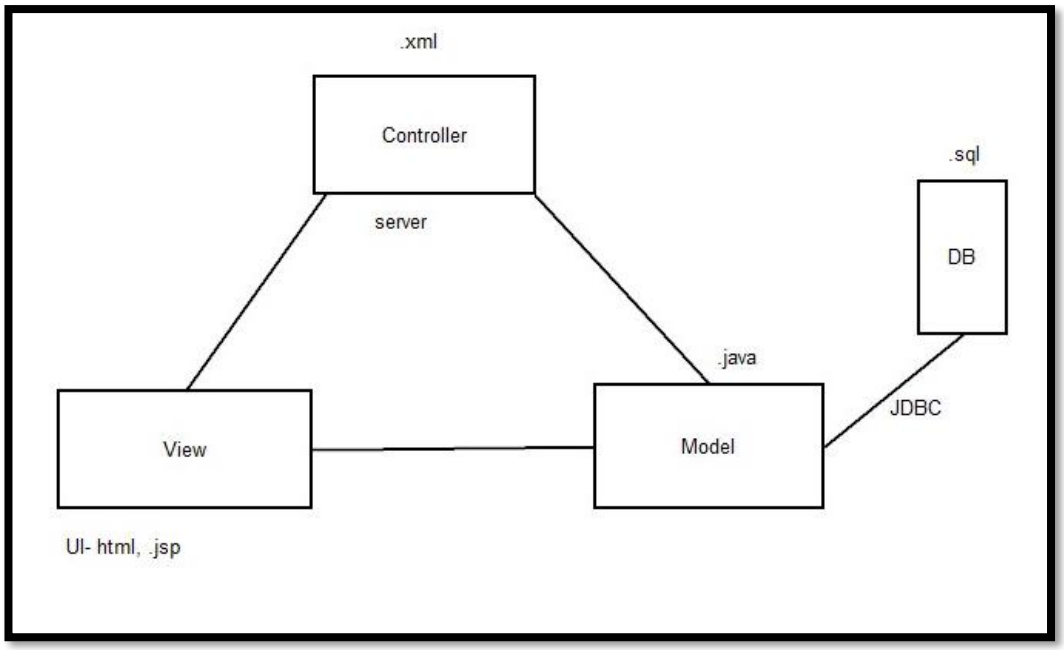

Figure 5.2 MVC Architecture

5.2.2 Servlets

Servlets technology is used to create a web application (resides at server side and generates a dynamic web page) Servlet technology is robust and scalable because of java language. There are many interfaces and classes in the Servlet API such as Servlet,

GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

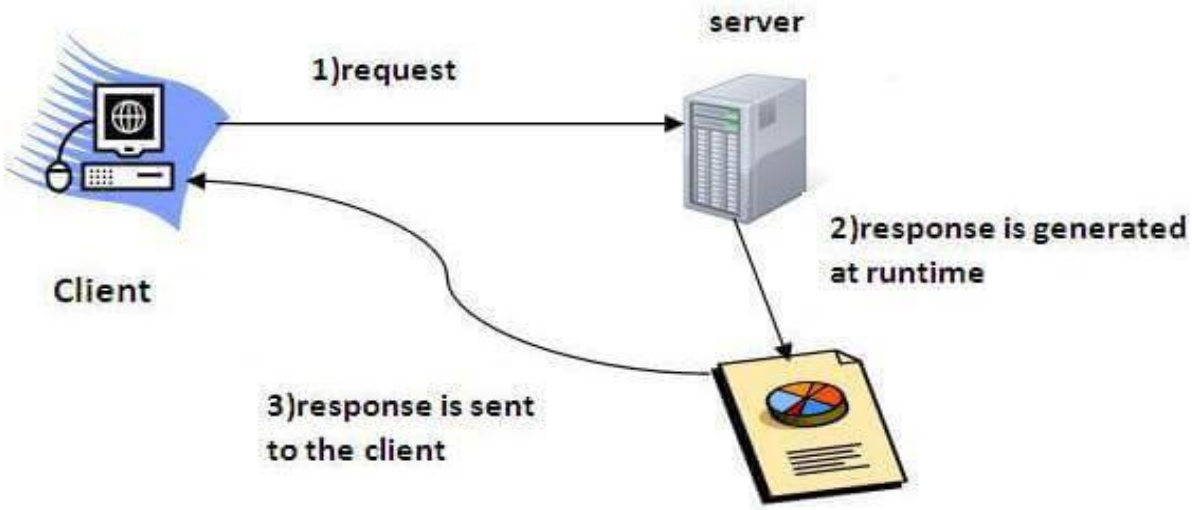

Figure 5.3 Servlets Technology working

- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.

The Hypertext Transfer Protocol (HTTP) is application-level protocol for collaborative, distributed, hypermedia information systems. It is the data communication protocol used to establish communication between client and server.

There are given 6 steps to create a servlet example.

1. Create a directory structure
2. Create a Servlet
3. Compile the Servlet
4. Start the server and deploy the project
5. Access the servlet

These steps are required for all the servers. The servlet example can be created by three ways:

- By implementing Servlet interface,
- By inheriting GenericServlet class, (or)
- By inheriting HttpServlet class
- The mostly used approach is by extending HttpServlet because it provides http request specific method such as doGet(), doPost(), doHead() etc.

The deployment descriptor is an xml file, from which Web Container gets the information about the servlet to be invoked. Web.xml file is the deployment descriptor in the project.

```
<web-app>
<servlet>
<servlet-name>Login</servlet-name>
<servlet-class>com.admin.Login</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Login</servlet-name>
<url-pattern>/Login</url-pattern>
</servlet-mapping>
</web-app>
```

- Sample web.xml file structure is given above. The explanation for each line is given below<web-app> represents the whole application.
- <servlet> is sub element of <web-app> and represents the servlet.
- <servlet-name> is sub element of <servlet> represents the name of the servlet.
- <servlet-class> is sub element of <servlet> represents the class of the servlet.

:

- <servlet-mapping> is sub element of <web-app>. It is used to map the servlet.
- <url-pattern> is sub element of <servlet-mapping>. This pattern is used at client side to invoke the servlet.

The Request Dispatcher interface provides the facility of dispatching the request to another resource it may be html, Servlet or JSP. The Forward method: Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.

5.2.3 JSP

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than Servlet.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because it can separate designing and development.

The scripting elements provide the ability to insert java code inside the jsp. There are three types of scripting elements:

1. scriptlet tag
2. expression tag
3. declaration tag

5.2.4 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs.

This consistent interface is achieved through the use of “plug-in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, user must provide the driver for each platform that the database and Java run on.

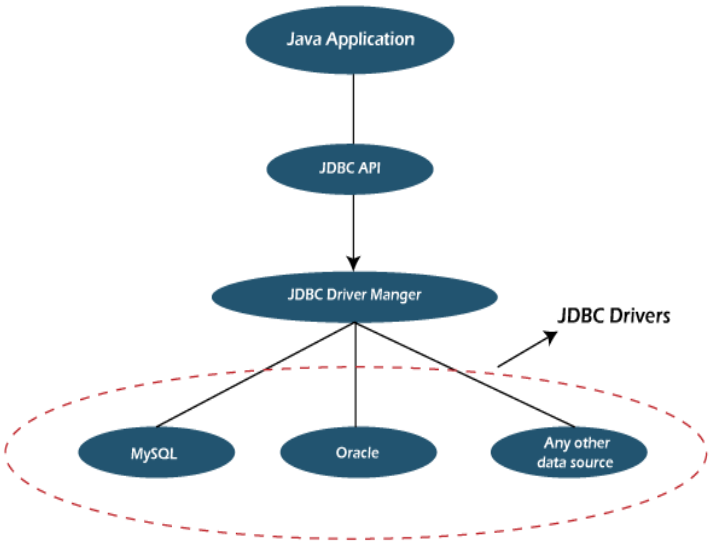

Figure 5.4 JDBC Architecture

Here the DriverManager plays an important role. It uses some database specific drivers to communicate with the J2EE application to database. The DriverManager class is the component of JDBC API and also a member of the java.sql package. The DriverManager class acts as an interface between users and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. In the project, using the com.mysql.jar file as the JDBC Driver.

5 Steps involved in connecting a Java Application with Database using a JDBC:

1. **Register the Driver.**
   - To begin with, you first need load the driver or register it before using it in the program . Registration is to be done once in the program. This step is mandatory.
   - Class.forName("com.mysql.jdbc.Driver");
2. **Create a Connection.**
   - After loading the driver, establish connections using :
   - Connection con = DriverManager.getConnection(url,user,password)
3. **Create SQL Statement.**
   - Once a connection is established you can interact with the database.
     Use of JDBC Statement is as follows:

- Statement st = con.createStatement();

4. Execute SQL Statement.

- Now comes the most important part i.e executing the query. Query here is an SQL Query. Now, can have multiple types of queries. Some of them are as follows:
- Query for updating / inserting table in a database.
- Query for retrieving data.

5. Closing the connection.

- The close() method of Connection interface is used to close the connection. This step is optional. Example :
- con.close();
- The sample code for the JDBC establishment is given below.

```
import java.sql.*;
 class AdminDAO
    {
      public static void main(String args[])
      {
        try
          {   Class.forName("com.mysql.jdbc.Driver");    //Loading the driver

            //Creating the comnnection
            Connection con=DriverManager.getConnection(  "jdbc:mysql://localhost:3306/db_name" , "root", "admin");

            //Creating the statement
            Statement stmt=con.createStatement();
                        ResultSet rs=stmt.executeQuery("select * from  m_admin");  //Executing the query
                        while(rs.next()) {
                                    System.out.println(rs.getString(1)+"  "+rs.getString(2));
                                     }
              con.close();    //Closing the connection
       }
        catch(Exception e)
            { System.out.println(e);
            }
    }
 }
```

5.3 ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also

be used to develop applications in other programming languages via plug, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, Ja vaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

5.4 TOMCAT

Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, Java Server Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

5.5 MY SQL

MySQL ("My Sequel") is (as of 2008) the world's most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python."

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records.

5.6 METHODOLOGY

A methodology is a set of ideas or guidelines about how to proceed in gathering and validating knowledge of a subject matter. Different areas of science have developed very different bodies of methodology on the basis of which to conduct their research.

[1] Using cryptography technique, going to secure outsourcing data. To secure the data can use asymmetric or symmetric algorithms. In project using the AES algorithm and using SHA1 algorithm for the hash function generation.

[2] AES stands for Advanced Encryption Standard and is a majorly used symmetric encryption algorithm. It is mainly used for encryption and protection of electronic data. It was used as the replacement of DES(Data encryption standard) as it is much faster and better than DES. AES consists of three block ciphers and these ciphers are used to provide encryption of data

[3] SHA-1 or Secure Hash Algorithm 1 is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency.

[4] In computer science, a fingerprinting algorithm is a procedure that maps an arbitrarily large data item (such as a computer file) to a much shorter bit string, its fingerprint, that uniquely identifies the original data for all practical purposes just as human fingerprints uniquely identify people for practical purposes. This fingerprint may be used for data deduplication purposes.

[5] fiFingerprinting. Fingerprints are typically used to avoid the comparison and transmission of bulky data. For instance, a web browser or proxy server can efficiently check whether a remote file has been modified, by fetching only its fingerprint and

comparing it with that of the previously fetched copy.

Fingerprint Algorithm:

Step1: Initialize colors

Step2: Open and create the buffered image

Step3: Create the binary picture

Step4: Generate Greymap

Step5: Binary local Result

Step6: Remove Noise (Remove noise from the binary picture using mean algorithm)

Step7: Skeletonization (skeletonize until there are no changes between two iterations)

Step8: Direction (Convert the direction matrix to direction buffered image)

Step9: Minutiaes (intersections-Extract intersection points from the binary image, which is a kind of minutiae)

Step10: Minutiaes (endpoints - Extract end points from the binary image, which is a kind of minutiae)

[6] FTP (File Transfer Protocol) is built on a client-server model architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. FTP is the commonly used protocol for exchanging files over the Internet. FTP uses the Internet's TCP/IP protocols to enable data transfer. FTP uses a client-server architecture, often secured with SSL/TLS. FTP promotes sharing of files via remote computers with reliable and efficient data transfer

[7] Cloud computing is an information technology (IT) paradigm that enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility. Third-party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance

[8] A block chain, originally block chain, is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a merkle tree root hash).

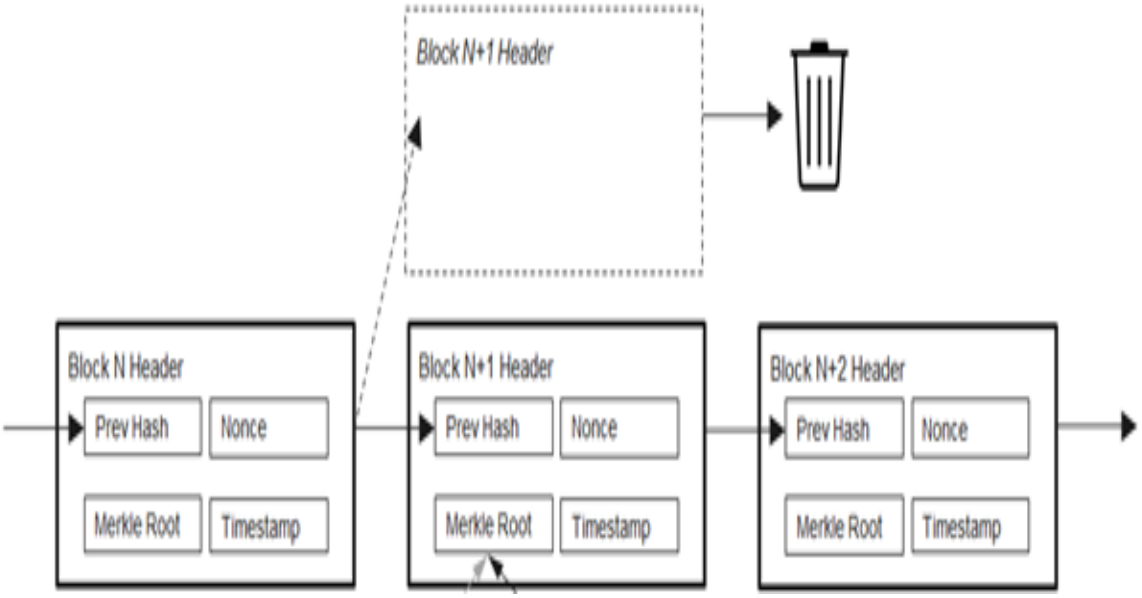

Figure 5.5 Block Chain Technology

A Blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network. This allows the participants to verify and audit transactions in expensively.

A Blockchain database is managed autonomously using a peer-to-peer network and a distributed time stamping server over the data. They are authenticated by mass collaboration powered by collective self-interests.

The result is a robust workflow where participants' uncertainty regarding data security is marginal. The use of a Blockchain removes the characteristic of infinite problems from a digital asset. It confirms that each unit of value was transferred only once, solving the long-standing problem of double spending. Blockchain have been described as a value-exchange protocol. This Blockchain-based exchange of value can be completed quicker, safer and cheaper than with traditional systems A Blockchain

can assign title rights because, when properly set up to detail the exchange agreement, it provides a record that compels offer and acceptance.

5.7 Code Implementation

```
package com.priya.admin;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


import com.priya.DAO.AdminDAO;

import com.symmetric_AES.AESCrypt;

import com.util.biometric.Utilityjava;

public class sendAadhaarcardrequest extends HttpServlet

{

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp)

                throws ServletException, IOException

try

        {

        {

            String defaultverification="unverified";
```

```
                System.out.println("its came inside the checkAadhaarcard");

                String Aadhaarno=req.getParameter("Aadhaarno");

                System.out.println("////////////////Aadhaarno//////////////////"+Aadhaarno);

                String verify=AdminDAO.checkverificationstatus(Aadhaarno);

                boolean flag=defaultverification.equals(verify);

                if(!flag)

                {

                        boolean request=AdminDAO.trancaterequest(Aadhaarno);

                        RequestDispatcher rd= req.getRequestDispatcher
                    ("/jsp/admin/Aadhaarrequestdetails.jsp?no=2");

                        rd.forward(req, resp);

                }

                else

                {

                                        RequestDispatcher
            rd=req.getRequestDispatcher("/jsp/admin/home.jsp?no=4");

                                    rd.forward(req, resp);

                }

        }

catch (Exception e)

        {

                System.out.println(e)

}

  }

};
```

```
package com.priya.admin;

import java.awt.Point;

import java.awt.image.BufferedImage;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.io.PrintWriter;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.List;

import java.util.Random;

import java.util.zip.ZipOutputStream;


import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.fileupload.FileItem;

import org.apache.commons.fileupload.FileItemFactory;

import org.apache.commons.fileupload.FileUploadException;

import org.apache.commons.fileupload.disk.DiskFileItemFactory;

import org.apache.commons.fileupload.servlet.ServletFileUpload;
```

```
import sun.text.normalizer.Utility;
import com.priya.DAO.AdminDAO;
import com.symmetric_AES.AESCrypt;
import com.util.biometric.FingerPrint;
import com.util.biometric.FingerPrint.direction;
import com.util.biometric.Utilityjava;
/**
 * @author
 *Leena
 */
public class Upload_DataSheet extends HttpServlet
{
        public void doPost(HttpServletRequest request, HttpServletResponse response)throws IOException
        {

        FileItemFactory fileItemFactory = new DiskFileItemFactory();
                ServletFileUpload servletFileUpload = new ServletFileUpload(fileItemFactory);
                List fileItems = null;
String Aadhaarno="",name="", fname="", mname="",dob="",sex="",hosueno="",village="",district="",state="",pincode="",phoneno="",emailid="",Aadhaar_image="",finger_print="";

                String featureToEncrypt="",encrypted = "";

                try
                { fileItems = servletFileUpload.parseRequest(request);
```

```
FileItem file1 = (FileItem) fileItems.get(0);
FileItem file2 = (FileItem) fileItems.get(1);
FileItem file3 = (FileItem) fileItems.get(2);
FileItem file4 = (FileItem) fileItems.get(3);
FileItem file5 = (FileItem) fileItems.get(4);
FileItem file6 = (FileItem) fileItems.get(5);
FileItem file7 = (FileItem) fileItems.get(6);
FileItem file8 = (FileItem) fileItems.get(7);
FileItem file9 = (FileItem) fileItems.get(8);
FileItem file10 = (FileItem) fileItems.get(9);
FileItem file11 = (FileItem) fileItems.get(10);
FileItem file12 = (FileItem) fileItems.get(11);
FileItem file13 = (FileItem) fileItems.get(12);
FileItem file14 = (FileItem) fileItems.get(13);
FileItem file15 = (FileItem) fileItems.get(14);

Aadhaarno=file1.getString();
name=file2.getString();
fname=file3.getString();

mname=file4.getString();
dob=file5.getString();
sex=state=file6.getString();
hosueno=file7.getString();
village=file8.getString();
district=file9.getString();
fname=file3.getString();
 mname=file4.getString()
```

```
dob=file5.getString();
 sex=state=file6.getString();
 hosueno=file7.getString();
 village=file8.getString();
 district=file9.getString();
 state=file10.getString();
 pincode=file11.getString();
 phoneno=file12.getString();
 emailid=file13.getString();
 Aadhaar_image=file14.getName();
 finger_print = file15.getString();
 String fingerprintfname = file15.getName();

 System.out.println("name:"+name);
 System.out.println("mother:"+mname);
 System.out.println("father :"+fname);
 System.out.println(" Dob:"+dob);
 System.out.println(">>>>>>>sex>>>>>>>>>"+sex);
 System.out.println(">>>>>>>hosueno>>>>>>>>>"+hosueno);
 System.out.println(">>>>>>>village>>>>>>>>>"+village);
 System.out.println(">>>>>>>>district>>>>>>>>"+district);
 System.out.println(">>>>>>>state>>>>>>>>>"+state);
 System.out.println(">>>>>>>>pincode>>>>>>>>"+pincode);
 System.out.println(">>>>>>>>phoneno>>>>>>>>"+phoneno);
System.out.println(">>>>>>>>emailid>>>>>>>>"+emailid);

System.out.println(">>>>>>>>>Aadhaar_image>>>>>>>"+Aadhaar_image);
```

```
/*
System.out.println(">>>>>>>>finger_print>>>>>>>>"+finger_print);*/

System.out.println(">>>>>>>>fingerprintfname>>>>>>>>"+fingerprintfnam
e);
AdminDAO adminDAO=AdminDAO.initialize();
Random r=new Random();

String fileName = request.getRealPath("") +
"/uploadedimages/"+name+"_"+Aadhaar_image;
OutputStream outputStream = new FileOutputStream(fileName);
InputStream inputStream = file14.getInputStream();
int readBytes = 0;
byte[] buffer = new byte[10000];
while ((readBytes = inputStream.read(buffer, 0, 10000)) != -1)
{
outputStream.write(buffer, 0, readBytes);
}
outputStream.close();
inputStream.close();

String fingerprintfilename = request.getRealPath("") +
"/uploadfingerprint/"+name+"_"+fingerprintfname;

OutputStream outputStream1 = new
FileOutputStream(fingerprintfilename);
InputStream inputStream1 = file15.getInputStream();
int readBytes1 = 0;
byte[] buffer1 = new byte[10000];
```

```
while ((readBytes1 = inputStream1.read(buffer1, 0, 10000)) != -1)
{
        outputStream1.write(buffer1, 0, readBytes1);
}
outputStream1.close();
inputStream1.close();
String pass=new Integer(r.nextInt(8253)).toString();
System.out.println("aAadhaar no ====> " + Aadhaarno);

//====================Extract the Features of the Finger Print=starts================
FingerPrint fingerprint1 = new FingerPrint(fingerprintfilename);
        System.out.println("its cazme inside feature extraction>>>>>>>>>>>>>>>>>>"+fingerprint1);

BufferedImage buffer11;

// Print original image
buffer11 = fingerprint1.getOriginalImage();

                // Print binary result
                fingerprint1.binarizeMean();
                buffer11 = fingerprint1.toBufferedImage();
    // Print binary local result
                fingerprint1.binarizeLocalMean();
                buffer11 = fingerprint1.toBufferedImage();
                fingerprint1.removeNoise();
```

```
                    fingerprint1.removeNoise();

                    buffer11 = fingerprint1.toBufferedImage();

                    // Skeletonization
                    fingerprint1.skeletonize();

                    // Direction
                    direction [][] dirMatrix = fingerprint1.getDirections();
                    buffer11 =
fingerprint1.directionToBufferedImage(dirMatrix);

                    // Core
                    buffer11 =
fingerprint1.directionToBufferedImage(dirMatrix);
                    Point core = fingerprint1.getCore(dirMatrix);
                    System.out.println("  Core....."+core);

                    int coreRadius = buffer11.getWidth() / 3;
                    System.out.println(" Core Radius....."+coreRadius);

                    // Minutiaes
                    buffer11 =
fingerprint1.directionToBufferedImage(dirMatrix);
                    ArrayList<Point> intersections =
fingerprint1.getMinutiaeIntersections(core, coreRadius);

        ArrayList<Point> endPoints = fingerprint1.getMinutiaeEndpoints(core,
coreRadius);
```

```
System.out.println(" MinutiaeIntersections......."+intersections);
System.out.println(" MinutiaeEndpoints....."+endPoints);

int count_intersection=intersections.size();
int count_endpoints=endPoints.size();

featureToEncrypt=count_intersection+"~"+count_endpoints;
System.out.println("priya finger print featureToEncrypt>>>>>>>>>>>>>>>>>>"+featureToEncrypt);
String value=Aadhaarno+"/"+featureToEncrypt;
System.out.println("Input value is>>>>>>>>>>>>>>>>>> :"+value);
AESCrypt  encrypter = new AESCrypt();
encrypted = encrypter.encrypt(value);

System.out.println("Encrypted File is>>>>>>>>>>>>>>>>>>>>>>>>>>"+encrypted);
String hashtagpath =
request.getRealPath("")+"\\Fingerprint_SHA1\\"+"SHA_"+name+".txt";
System.out.println("File Path :"+hashtagpath);
result=adminDAO.upload_datasheet(Aadhaarno,name,pass, fname, mname, dob, sex, hosueno,village,district,state, pincode, phoneno, emailid, Aadhaar_image, fingerprintfname,featureToEncrypt,previousblkvale,sha1,date,randomNumber,zipblkame);

SendMailAttachment.sendPersonalizedMail(emailid, "Information Related User-ID & Password.....", "Dear "+name+"\n\n\n User id-"+Aadhaarno+" ....\n Password-"+pass+" .....\n\n\n\n\n\n\nThank You...\nE_AADHAAR India\nE_AADHAAR@govr.com  ", "no");
```

```
                    if(result)
            {
                    System.out.println("it came inside the result is>>>>>>>>>>>>>>
");
                    RequestDispatcher rd=null;
rd=request.getRequestDispatcher("/jsp/admin/upload_datasheets.jsp?no=1");
                            rd.forward(request, response);

                    }
            }
            else
            {
                    RequestDispatcher rd=null;

        rd=request.getRequestDispatcher("/jsp/admin/ViewDatasheet.jsp?no=1");
                    rd.forward(request, response);
            }
            }

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

        }
```

Chapter 6
TESTING AND IMPLEMENTATION

Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected.

6.1 Software Testing Introduction

Software testing is a process used to help identify the correctness, completeness, and quality of developed computer software. Software testing is the process used to measure the quality of developed software. Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification & validation

6.2 Explanation for SDLC & STLC

SDLC: The software development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

6.3 Phases of Software Development

- Requirement Analysis
- Software design
- Development or Coding
- Testing
- Maintenance

The requirements of a desired software product are extracted. Based the business scenario the SRS (Software Requirement Specification) document is prepared in this phase.Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues for the software. Design phase is as described in the below stages :

There are 2 stages in design,

HLD – High Level Design

LLD – Low Level Design

HLD – gives the architecture of the software product to be developed and is done by architects and senior developers.

LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

Testing: Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding defects.

Maintenance: Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures

6.4 SDLC Models

Water Fall Model: It will be executing one by one of the SDLC process. The design Starts after completing the requirements analysis coding begins after design. It is a traditional model It is a sequential design process, often used in SDLC, in which the progress is seen as flowing steadily downwards (like a waterfall), through the different phases.

Prototype Model**:** Developed from the sample after getting good feedback from the customer. This is the Valuable mechanism for gaining better understanding of the customer needs.

Rapid Application Development Model (RAD): This mechanism will develop from already existing one . If The New requirement is matching in already existing requirement, will develop from that**.**

Spiral Model: This mechanism is updating the application version by version. All the SDLC process will update version by version**.**

V-Modelv: V model is a process where the development and testing phases can do parallelly. For every development phase there is a testing phase. Development phases are called as verification whereas testing phases are called as validation

6.5 STLC (**Software Testing Life Cycle)**

Testing itself has many phases i.e. is called as STLC. STLC is part of SDLC

- Test Plan
- Test Development
- Test Execution
- Analyze Results
- Defect Tracking
- Summaries Report

6.6 Types of Testing:

- White Box Testing
- Black Box Testing
- Grey box testing

White Box Testing**:** White box testing as the name suggests gives the internal view of the software. This type of testing is also known as structural testing or glass box testing as well, as the interest lies in what lies inside the box.

Black Box Testing: It's also called as behavioral testing. It focuses on the functional requirements of the software. Testing either functional or nonfunctional without reference to the internal structure of the component or system is called black box testing.

Grey Box Testing: Grey box testing is the combination n of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.

6.7 Level of Testing Used in Project

A. Unit Testing: Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable**.**

B. Integration Testing: All module which makes application are tested. Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirement**.**

C. System Testing: To test the complete system in terms of functionality and non functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

D. Functional Testing: The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms.

E. Alpha Testing: Alpha testing is final testing before the software is released to the public. This testing is conducted at the developer site and in a controlled environment by the end user of the software.

F. Beta Testing: The beta test is conducted at one or more customer sites by the end user of the software. The beta test is conducted at one or more customer sites by the end.

6.8. Unit Testing Cases:

Initialization testing is the first level of dynamic testing and is first the of the test engineers. Unit testing is performed after the expected test results are met or are

differences explainable/acceptable**.**

Table – 6.1 Unit test case for login

Sl # Test Case : -	UNTC-1
Name of Test: -	Login as Admin
Items being tested: -	Admin model
Sample Input: -	Correct Username & Password is given as inputs
Expected output: -	Depending on the correct inputs, it must login as Admin
Actual output: -	Login successful
Remarks: -	Pass.

Table- 6.2 unit test case for failed login

Sl # Test Case : -	UNTC-2
Name of Test: -	Login as User
Items being tested: -	User model
Sample Input: -	Incorrect Username & Password is given as inputs
Expected output: -	Depending on the incorrect inputs, it shouldn’t login as User
Actual output: -	Login Failed.
Remarks: -	Pass

Do the changes of wrong values of username & password in data base with correct username & correct password and save the database.

6.9 System Testing

To test the complete system in terms of functionality and non functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

Table 6.3 system testing

Sl # Test Case : -	STC-1
Name of Test: -	System testing in various versions of OS
Sample Input: -	Execute the program in windows server 2003/XP/ Windows-7
Expected output: -	Performance is better in windows-8
Actual output: -	Same as expected output, performance is better in windows-8
Remarks: -	Pass

6.10 Functional Testing:

It is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing).

Table-6.4 Functional Testing

Test Case Id	Test Input	Expected Result	Actual Result	Remarks
F_01	Enter valid admin username and password click on login button	Admin should be navigated to homepage with the following details: User creation, user details, cloud config, key setting, upload files, keyword rank, user request change password and logout	Admin is navigated to homepage with the following details: User creation, user details, cloud config, key setting, upload files, keyword rank, user request change password and logout	Pass
F_02	Enter valid username and password in the user login page click on login button	User should be navigated to the homepage with the following: user profile, send request search keyword, change password file request and logout	User is navigated to the homepage with the following details user profile, send request search keyword, change password file request and logout	Pass
F-03	Click on change password on admin module to change password	Admin should be navigated to change password page there the admin change password by providing the new password.	Admin is navigated to change password page there the admin change password by providing the new password.	Pass
F-04	Click on change password on user module to change password	User should be navigated to change password page there the admin change password by providing the new password.	User is navigated to change password page there the admin change password by providing the new password.	Pass

6.11. Integration Testing:

The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms.

Table 6.5- integration testing

Sl # Test Case : -	ITC-1
Name of Test: -	Change Password for Admin
Item being tested: -	Admin & User Model
Sample Input: -	Incorrect Old password, give new password.
Expected output: -	Failed to update new password.
Actual output: -	Password Update failed
Remarks: -	Pass.

Chapter 7

RESULT AND ANALYSIS

A result is the final consequence of actions or events expressed qualitatively or quantitatively. Performance analysis is an operational analysis, is a set of basic quantitative relationship between the performance quantities.

The SHA(Secured Hash Algorithm) technique has been used to ensure the privacy of identification requests and the biometric dataset. The security of proposed scheme has been analyzed, and the result shows that the privacy of both the biometric dataset and biometric identification can be preserved.

To evaluate the computational and communication cost of the proposed scheme, implement it and test it over a synthetic dataset. Experimental results demonstrate that the proposed scheme is efficient in terms of computational and communication costs when identifying a biometric template in a large dataset.

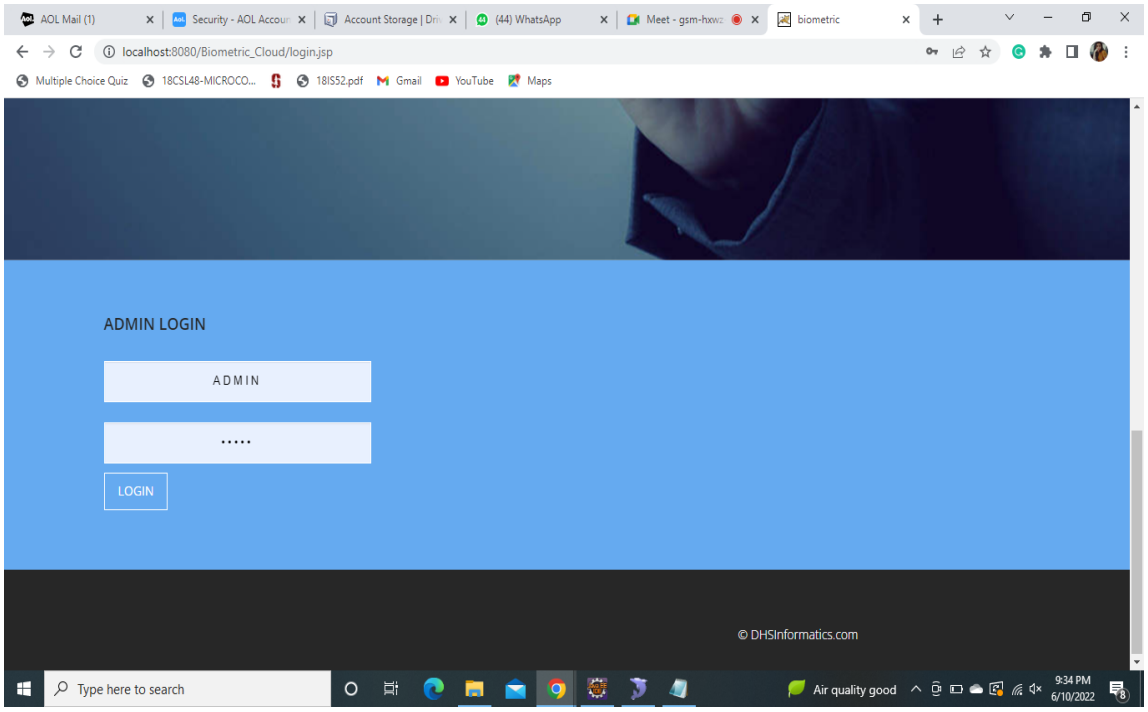

Figure 7.1: Admin Login

Admin login page, is the home page of the project and consists of the admin login portal where the username and password has been set to: ADMIN. The admin can log into the admin account by accessing this page.

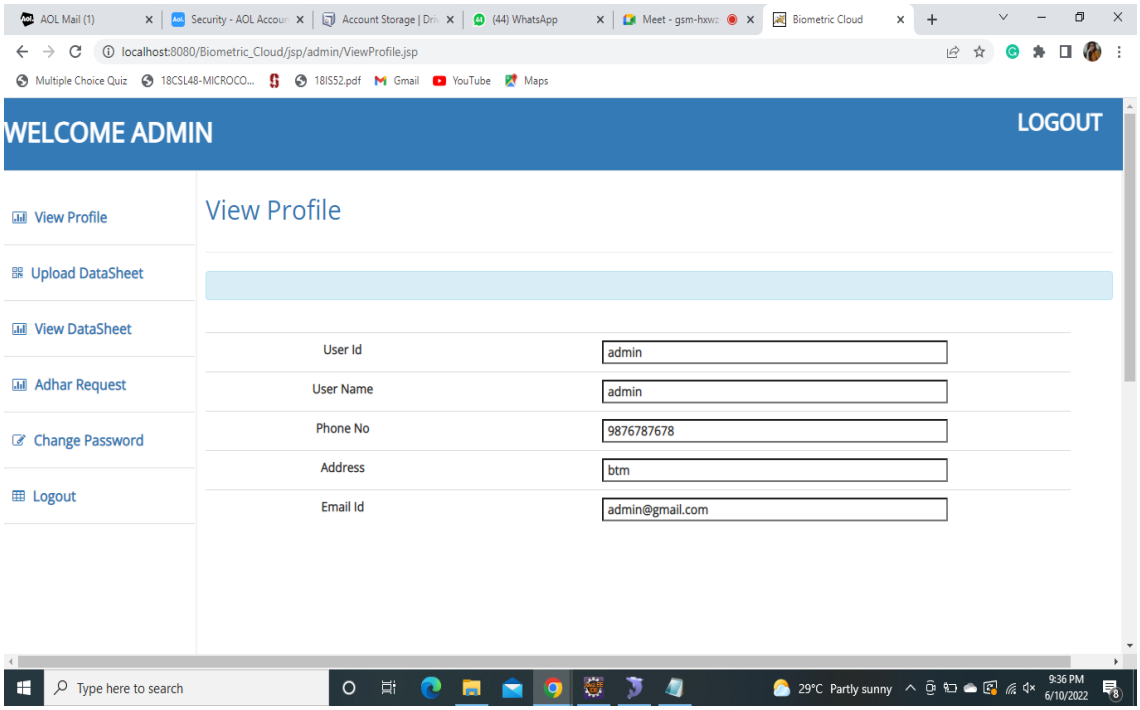

Figure 7.2: Admin profile

The Admin database consists of View Profile, Upload datasheet, View Datasheet, Adhar Request, Change Password, Logout, the following interfaces.

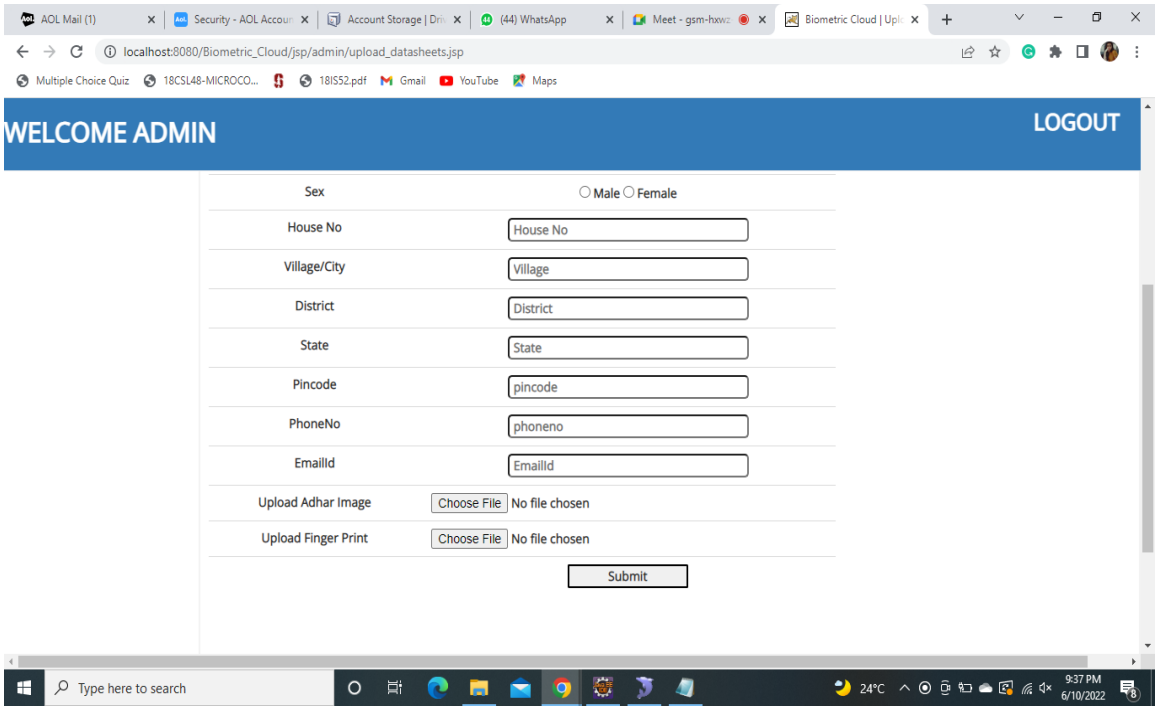

Figure 7.3: Upload data Sheet

The Upload datasheet interface is used by the admin to register a new user, where the admin is provided the option to upload the user's Adhar card and scanned fingerprint image..

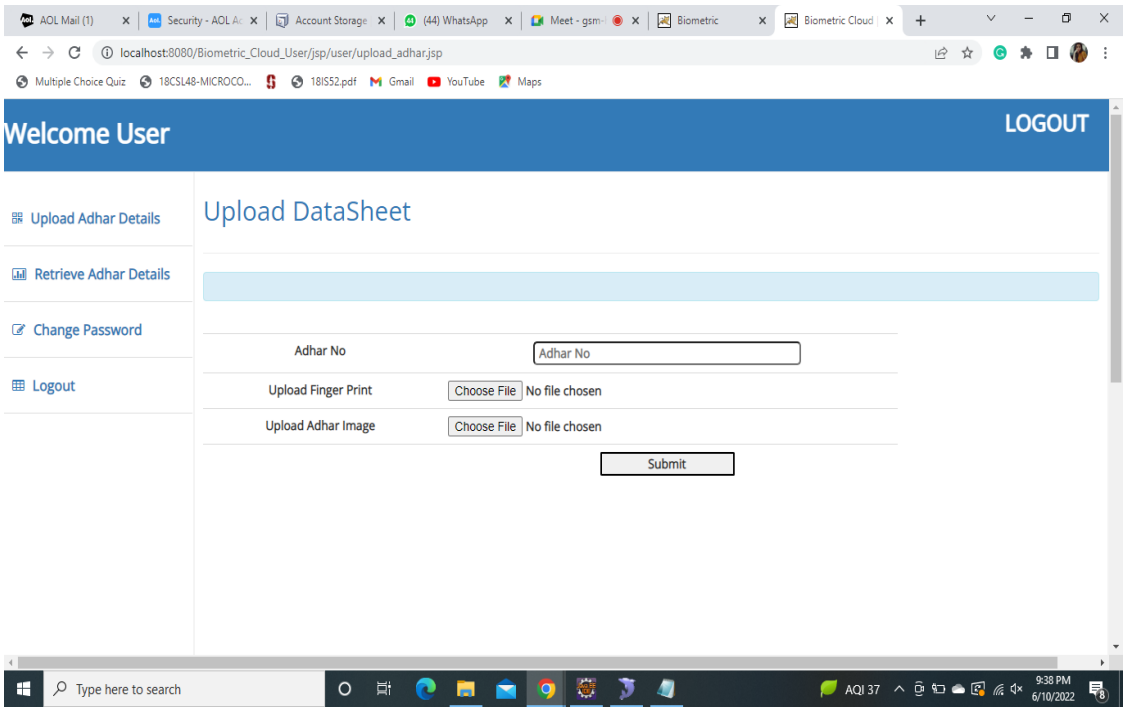

Figure 7.4: Upload datasheet by user

The upload datasheet interface of the user account can be accessed by the user by logging into the user account. The email ID is the user Adhar number and the password in a random number that is generated and sent to the user email ID

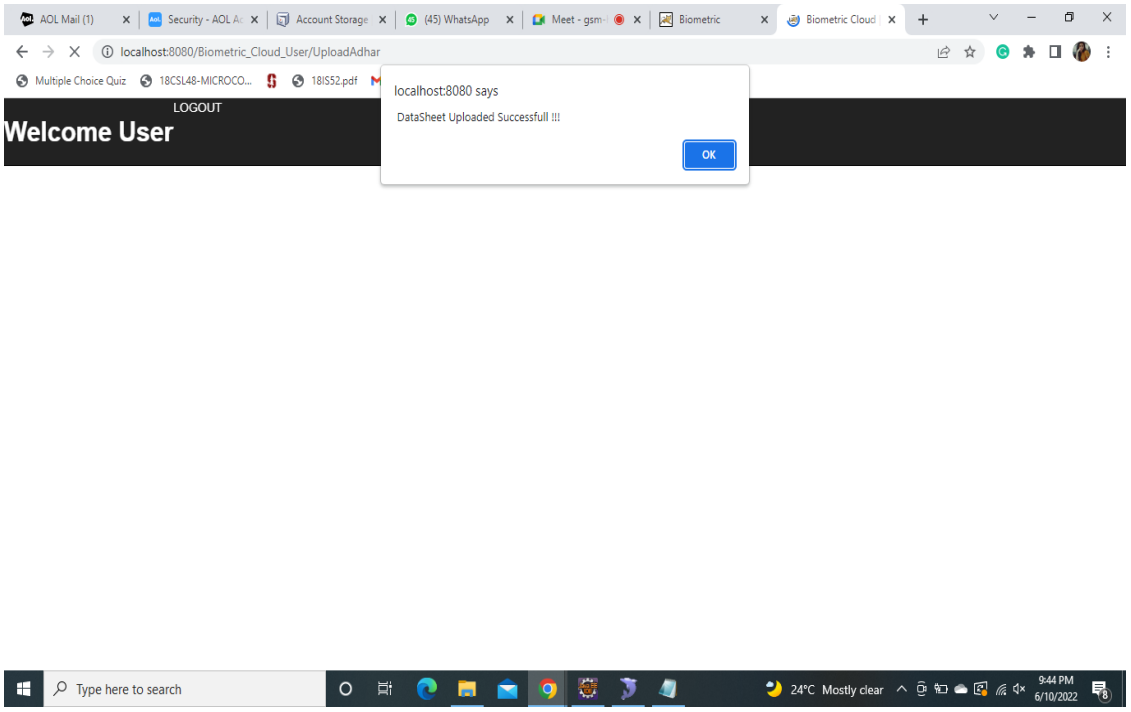

Figure 7.5: Data sheet upload successfully message

The user can request their respected data by uploading their Adhar image and fingerprint scanning image and sending a data request message to the Admin.

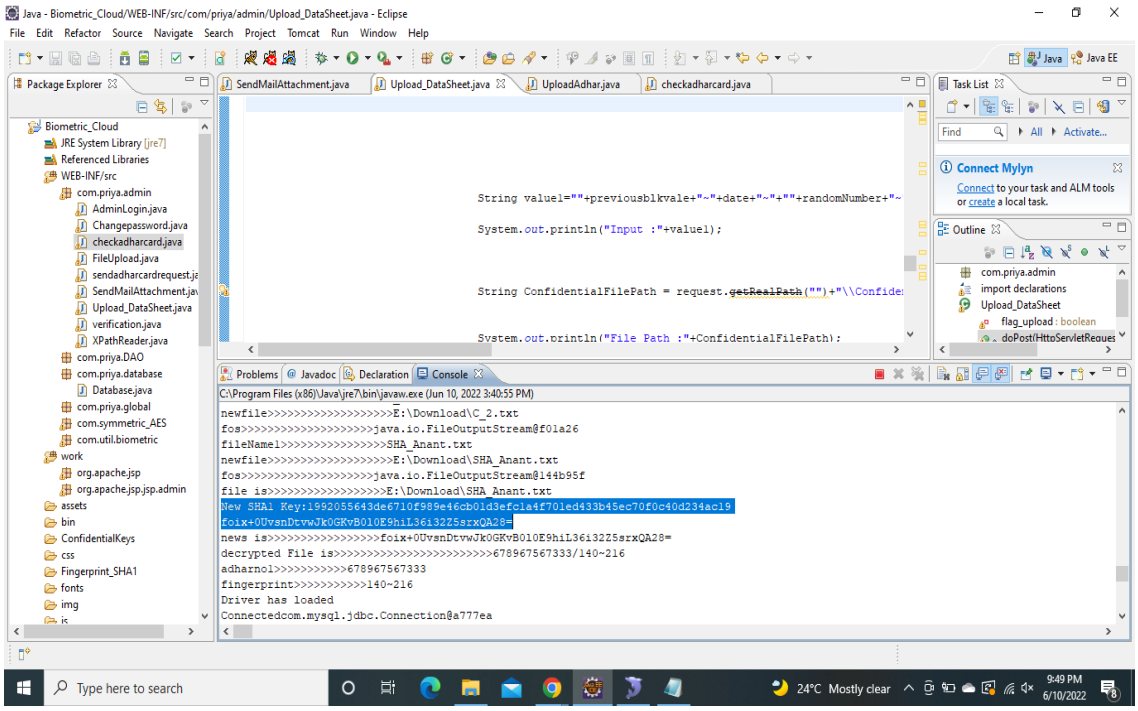

Figure 7.6: Encryption and decryption happening in backend

At the backend of the running project it can be seen that fingerprint matching is taking place by comparing the extraction feature of the 2 fingerprints uploaded, and also the AES algorithm is encrypting the data submitted by the admin.

Chapter 8

CONCLUSION AND FUTURE ENHANCEMENT

Proposing a novel privacy-preserving biometric identification scheme in cloud computing. To realize the efficiency and secure requirements, that have designed a new encryption algorithm and cloud authentication certification.

The detailed analysis shows it can resist the potential attacks. Besides, through performance evaluations, further demonstrated the proposed scheme meets the efficiency need well. The encryption algorithm used is extremely secure and not penetrable at all. It works like a secure network.

Suggesting a number of future research directions intending to focus on the implementation of a usable and secure authentication system with privacy preservation.

- First, research on a secure and privacy-preserving biometric authentication system is urgently needed. A number of open issues need to be solved as soon as possible facing such a complex and risky cyberspace. At present, the widely used biometric authentication system based on static characteristics, such as touchID and faceID needs to provide a means of liveliness detection.

- Second, usability enhancement and accuracy insurance are worth particular exploration for achieving high level user acceptance and wide adoption. A series of factors could affect the usability of a biometric authentication system, including UI design, user-device interaction design, data collection method, authentication protocol design, and so on. How to design a usable biometric authentication system is a significant topic, especially when security and privacy should be considered.

- Third, cost of authentication in a source limited mobile device should be considered. Most mobile devices (such as mobile phones and smart bracelets) have limited resources of electricity, computing capability, storage space, and

- so on. Therefore, it becomes essential to study biometric authentication methods and algorithms that can be implemented in wearable devices or even low-end devices with low computational requirements.

- Fourth, the systems based on dynamic features have a potential for further study due to its advantages regarding aliveness detection. After the comprehensive comparison on the existing works, coupled with discussion, believe that in the field of biometric authentication, the systems based on dynamic features have a potential. It does not require user distraction to make input data on the screen, nor does it require users to fix their body positions. From the user point of view, they provide more convenience to users than the authentication system based on iris, face and so on, since when collecting iris or face images, users have to look at the camera and hold their position for a while.

REFERENCES

[1] A. Jain, L. Hong, and S. Pankanti, ‘‘Biometric identification,’’ Commun. ACM, vol. 43, no. 2, pp. 90–98, 2000.

[2] R. Allen, P. Sankar, and S. Prabhakar, ‘‘Fingerprint identification technology,’’ in Biometric Systems. London, U.K.: Springer, 2005, pp. 22–61.

[3] J. de Mira, Jr., H. V. Neto, E. B. Neves, and F. K. Schneider, ‘‘Biometric oriented iris identification based on mathematical morphology,’’ J. Signal Process. Syst., vol. 80, no. 2, pp. 181–195, 2015.

[4] S. Romdhani, V. Blanz, and T. Vetter, ‘‘Face identification by fitting a 3D morphable model using linear shape and texture error functions,’’ in Proc. Eur. Conf. Comput. Vis., 2002, pp. 3–19.

[5] Y. Xiao et al., ‘‘A survey of key management schemes in wireless sensor networks,’’ Comput. Commun., vol. 30, nos. 11–12, pp. 2314–2341, Sep. 2007.

[6] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, ‘‘An effective key management scheme for heterogeneous sensor networks,’’ Ad Hoc Netw., vol. 5, no. 1, pp. 24–34, 2007.

[7] X. Du and H. H. Chen, ‘‘Security in wireless sensor networks,’’ IEEE Wireless Commun. Mag., vol. 15, no. 4, pp. 60–66, Aug. 2008.

[8] X. Hei and X. Du, ‘‘Biometric-based two-level secure access control for implantable medical devices during emergencies,’’ in Proc. IEEE INFOCOM, Apr. 2011, pp. 346–350.

[9] X. Hei, X. Du, J. Wu, and F. Hu, ‘‘Defending resource depletion attacks on implantable medical devices,’’ in Proc. IEEE GLOBECOM, Dec. 2010, pp. 1–5.

[10] M. Barni et al., ‘‘Privacy-preserving fingercode authentication,’’ in Proc. 12th ACM Workshop Multimedia Secur., 2010, pp. 231–240.

[11] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, ‘‘SCiFI—A system for secure face identification,’’ in Proc. IEEE Symp. Secur. Privacy (SP), May 2010, pp. 239–254.

[12] D. Evans et al., ‘‘Efficient privacy-preserving biometric identification,’’ in Proc. 17th Conf. Netw. Distrib. Syst. Secur. Symp. (NDSS), 2011, pp. 1–4

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

[14] Q. Wang, S. Hu, K. Ren, M. He, M. Du, and Z. Wang, ‘‘CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud,’’ in Proc. Eur. Symp. Res. Comput. Secur., 2015, pp. 186–205.

[15] Y. Zhu, Z. Wang, and J. Wang, ‘‘Collusion-resisting secure nearest neighbor query over encrypted data in cloud,’’ in Proc. IEEE/ACM 24th Int. Symp. Quality Ser. (IWQoS), Jun. 2016, pp. 1–6.

[16] S. Pan, S. Yan, and W. Zhu, ‘‘Security analysis on privacy-preserving cloud aided biometric identification schemes,’’ in Proc. Australasian Conf. Inf. Secur. Privacy, 2016, pp. 446–453.

[17] C. Zhang, L. Zhu, and C. Xu, ‘‘PTBI: An efficient privacy-preserving biometric identification based on perturbed term in the cloud,’’ Inf. Sci., vols. 409–410, pp. 56–67, Oct. 2017.

[18] 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. E97.D, no. 2, pp. 326–330, 2014.

[19] 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.

[20] H. Delfs, H. Knebl, and H. Knebl, Introduction to Cryptography. Berlin, Germany: Springer, 2002.

[21] K. Liu, C. Giannella, and H. Kargupta, ‘‘An attacker’s view of distance preserving maps for privacy preserving data mining,’’ in Knowledge Discovery in Databases: PKDD (Lecture Notes in Computer Science). Heidelberg, Germany: Springer, 2006, pp. 2