SECURING DATA WITH ENCRYPTED K-NN DEFENDING AGAINST STATISTICAL ANALYSIS

PROJECT REPORT

21AD1513- INNOVATION PRACTICES LAB

Submitted by

HARRISH KUMAR S	**211422243097**
HARRISH SEBASTIN A	**211422243098**
HARISHWARAN N	**211422243096**

in partial fulfillment of the requirements for the award of degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123

ANNA UNIVERSITY: CHENNAI-600 025

November, 2024

BONAFIDE CERTIFICATE

Certified that this project report titled “**SECURING DATA WITH ENCRYPTED K-NN DEFENDING AGAINST STATISTICAL ANALYSIS**” is the bonafide work of **HARRISH KUMAR S (211422243097)** , **HARRISH SEBASTIN A (211422243098) , HARISHWARAN N (211422243096)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

HEAD OF THE DEPARTMENT
Dr. S. MALATHI M.E., Ph.D .,
Professor and Head,
Department of AI & DS,
Panimalar Engineering college,
Chennai – 600123.

INTERNAL GUIDE
Ms.K.CHARULATHA M.E.,
Assistant Professor,
Department of AI & DS
Panimalar Engineering college,
Chennai – 600123.

Certified that the candidate was examined in the Viva-Voce Examination for the course **21AD1513- INNOVATION PRACTICES LAB** held on …………..

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

I also take this opportunity to thank all the Faculty and Non-Teaching Staff Members of Department of Computer Science and Engineering for their constant support. Finally I thank each and every one who helped me to complete this project. At the outset we would like to express our gratitude to our beloved respected Chairman, **Dr.Jeppiaar M.A.,Ph.D,** Our beloved correspondent and Secretary **Mr.P.Chinnadurai M.A., M.Phil., Ph.D.,** and our esteemed director for their support.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr.S.Malathi M,E.,Ph.D.,** of Artificial Intelligence and Data Science for her encouragement.

Personally we thank **Ms.K.Charulatha M.E, Assistant Professor**, Department of Artificial Intelligence and Data Science for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our thanks to the project coordinators **Mrs.V.Rekha M.E., Assistant Professor** in Department of Artificial Intelligence and Data Science for their Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, friends, and well-wishers who have helped us for the successful completion of our project.

We also take the opportunity to thank all faculty and non-teaching staff members in our department for their timely guidance in completing our project.

HARRISH KUMAR.S **HARRISH SEBASTIN.A** **HARISHWARAN.N**

(211422243097) **(211422243098)** **(211422243096)**

TABLE OF CONTENTS

.

ABSTRACT

In the article we proposed the Encryption and decryption using cryptographic algorithm AES(Advance Encryption Standard) is used to secure the data on the educational details and for important educational certificate in an organization or educational department, AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data others involve shuffling bits around (permutations). The implementation of cryptographic algorithms is used to achieve the prevention of significant data loss and to avoid malevolent situations, which will provide an optimal learning environment. Cryptography is used to limit data leakage and ensure data security. The Key generator has been using to generate different keys randomly with help of RNG(Random Number Generator) at each time while encrypting the data, It has the symmetric key chipper block is used to generate the key as range in 128 or 192 or 256 bit of key value. A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption. The mode of operation will handles the process in series of sequential message in an blocks, there are various encryption modes such as ECB, CBC, CFB, OFB, CTR etc., This application has contains the CBC(Cipher Blocking Chain) mode of to improve the vulnerability, through this multiple blocks has been encrypted as parallel process. It also avoids the bit flipping attacks from the foreign object

Keywords : Encryption, Decryption, Cryptographic, Algorithm, AES (Advanced Encryption Standard), Data Security, Educational Data, Educational Certificates, Substitution-Permutation Network, Bit Shuffling.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATIONS	MEANING
AES	Advanced Encryption Standard
RNG	Random Number Generator
ECB	Electronic Codebook
CBC	Cipher Block Chaining
CFB	Cipher Feedback
OFB	Output Feedback
CTR	Counter Mode

CHAPTER 1

INTRODUCTION

1.1 Introduction to Data Security in Education:

Cryptography is a foundational element of modern information security, providing techniques to protect sensitive data from unauthorized access, tampering, and exposure. It employs mathematical algorithms and protocols to ensure the confidentiality, integrity, and authenticity of data during storage and transmission. In an era where vast amounts of data are generated and shared digitally, cryptographic methods are essential for safeguarding personal, financial, and organizational information from malicious attacks and leaks.

In the context of data protection, one of the key applications of cryptography is **data encryption**, which transforms plaintext data into an unreadable format that can only be decrypted with the correct cryptographic key. This ensures that sensitive data—such as personal identifiers, medical records, financial transactions, and educational information—remains secure, even if intercepted by unauthorized entities.

1.2 Overview of Cryptography and Its Applications:

Cryptography is a foundational element of modern information security, providing techniques to protect sensitive data from unauthorized access, tampering, and exposure. It employs mathematical algorithms and protocols to ensure the confidentiality, integrity, and authenticity of data during storage and transmission. In an era where vast amounts of data are generated and shared digitally, cryptographic methods are essential for safeguarding personal, financial, and organizational information from malicious attacks and leaks.In the context of data protection, one of the key applications of cryptography is data encryption, which transforms plaintext data into an unreadable format that can only be decrypted with the correct cryptographic key. This ensures that sensitive data—such as personal identifiers, medical records, financial transactions, and educational information—remains secure, even if intercepted by unauthorized entities.

1.2 AES (Advanced Encryption Standard) and Its Role in Data Protection:

The Advanced Encryption Standard (AES) is one of the most widely used and trusted cryptographic algorithms in the world today. AES provides robust encryption techniques that ensure the confidentiality and integrity of sensitive data, making it a cornerstone of modern cybersecurity. Whether for protecting personal information, financial transactions, or institutional data, AES plays a critical role in securing information in an increasingly interconnected world.

AES was established by the U.S. National Institute of Standards and Technology (NIST) in 2001 as the official encryption standard after a rigorous selection process. It replaced the older Data Encryption Standard (DES), which had become vulnerable to modern computational power. AES was designed to address the growing need for a more secure and efficient encryption standard that could handle large amounts of data while providing robust security**.**

Key Features of AES

AES is a **symmetric-key** encryption algorithm, meaning that the same key is used for both encryption and decryption. This is in contrast to asymmetric encryption, which uses a pair of keys (public and private) for these operations. The symmetric nature of AES makes it computationally efficient, which is why it is widely adopted in both hardware and software implementations.

Key features of AES include:

1. **Block Cipher**: AES operates on fixed-size blocks of data (128 bits per block). This fixed block size ensures consistent processing of data and is a key design element that contributes to AES's security and efficiency.
2. **Key Lengths**: AES supports three key lengths—128, 192, and 256 bits. The key length directly impacts the strength of encryption:
   - **AES-128** provides a high level of security and is computationally faster than its longer counterparts.
   - **AES-192** offers an intermediate level of security and performance.
   - **AES-256** provides the highest level of security, making it suitable for extremely sensitive data but may come with slightly increased processing requirements.
3. **Rounds**: AES operates through multiple rounds of processing the data, where each round involves several operations, including substitution, permutation, and mixing. The number of rounds varies depending on the key size:
   - 10 rounds for AES-128
   - 12 rounds for AES-192

- 14 rounds for AES-256

4. **Substitution-Permutation Network**: AES employs a combination of substitution (replacing bytes) and permutation (shuffling bits) operations in its rounds. This approach significantly strengthens the cipher against attacks by making the relationship between the ciphertext and plaintext highly non-linear.
5. **S-Boxes**: One of the key components of AES's substitution step is the use of **S-Boxes** (substitution boxes), which are lookup tables that transform bytes of data. This makes it more difficult for attackers to reverse-engineer the encryption.

AES in Data Protection

AES is extensively used to protect data in various contexts, from securing communications on the internet to encrypting stored files. Its reliability, speed, and security have made it the algorithm of choice in numerous security protocols, including:

1. **File Encryption:** AES is commonly used to encrypt files on disk, ensuring that even if an attacker gains access to a device, the data remains unreadable without the decryption key. For example, applications like BitLocker and FileVault use AES to encrypt data stored on hard drives and solid-state drives (SSDs) in computers.
2. **Secure Communications:** AES is fundamental in securing communication protocols, such as SSL/TLS (used for HTTPS in web browsing) and IPSec (used for virtual private networks, or VPNs). These protocols ensure that sensitive data—like login credentials, credit card numbers, and personal communications—remain encrypted during transmission.
3. **Database Encryption:** Organizations often use AES to encrypt sensitive database records. This ensures that even if an attacker manages to access the database, the stored data remains unreadable unless they have the decryption key.
4. **Cloud Storage Encryption:** With the increasing use of cloud storage, AES has become a critical tool for securing data stored in the cloud. Services such as Google Drive, Dropbox, and iCloud implement AES encryption to ensure that user data is protected both in transit and at rest.
5. **Regulatory Compliance:** Many industries are subject to regulations and compliance standards, such as HIPAA (Health Insurance Portability and Accountability Act) for healthcare and GDPR (General Data Protection Regulation) for European data subjects, that mandate the use of encryption to protect sensitive data. AES meets these requirements and is widely recognized as the standard encryption method for compliance**.**

ARCHITECTURE DIAGRAM :

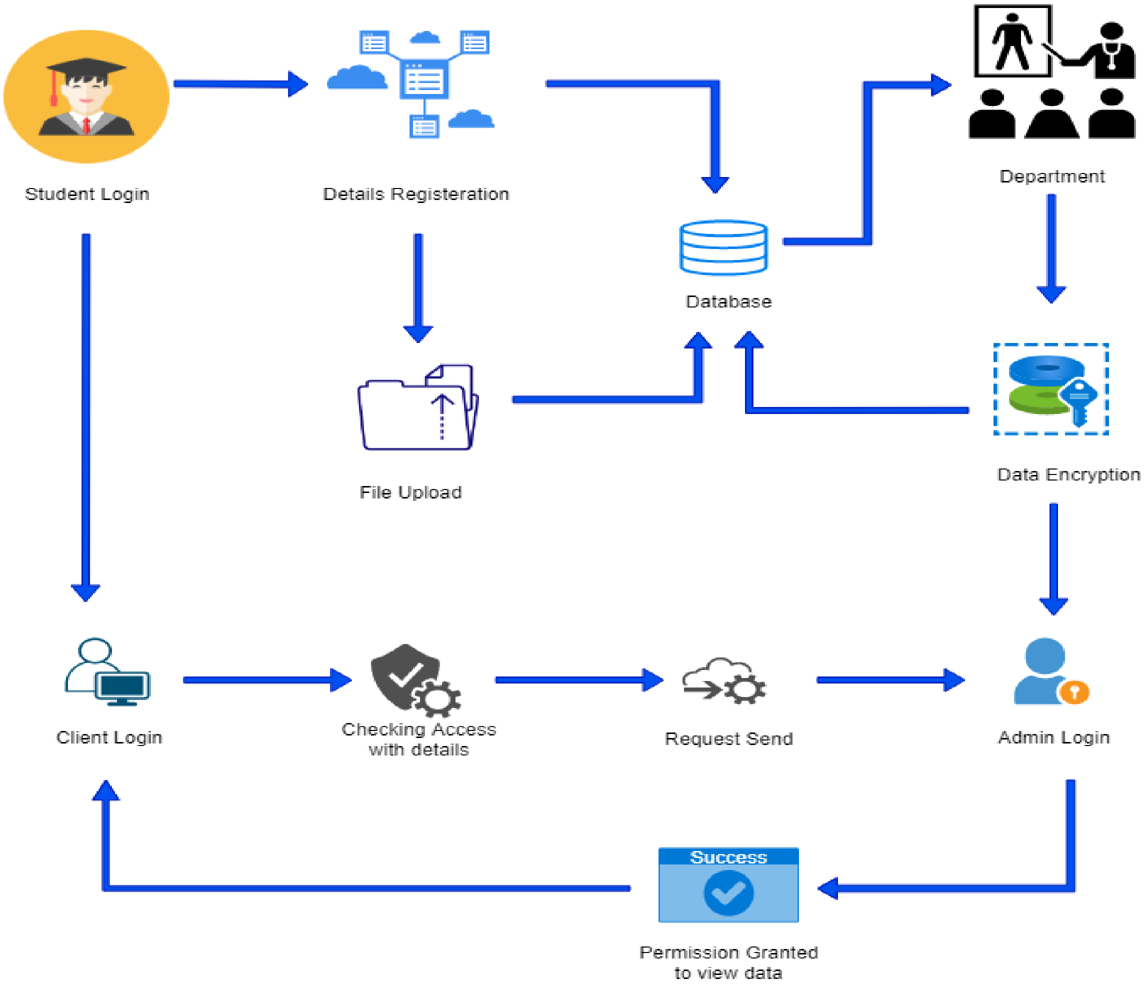

Fig 1.4: Educational organization secure computing architecture

The educational organization is run as manual certificate distribution and it have more chance to malpractice, frauds or it will destroyed in any global disasters, so that student and organization will may suffer to handle the situation, the overall organization will may face the critical risk situation through these things. In Future also the student has a difficulty to face the problems for them to recover the difficult situation and problems will happens on joining to any interested jobs. The educational department also has responsibilities by providing the students certificate and to securing the details of the student

1.3 Importance of Secure Educational Certificates and Sensitive Data:

In the digital age, educational institutions are not only custodians of knowledge but also stewards of sensitive personal data. As the reliance on electronic records and digital credentials grows, ensuring the security of educational certificates and other sensitive data has become a paramount concern. From academic transcripts to diplomas, these records are central to a student's academic history and often carry significant personal, professional, and legal weight.

The importance of securing educational certificates and sensitive data cannot be overstated, as data breaches and unauthorized access can have serious consequences not just for individuals, but for the institutions responsible for safeguarding that information. This section will explore the reasons why securing educational certificates and sensitive data is critical and the risks posed by inadequate protection.

CHAPTER 2

LITERATURE REVIEW

2.1 Cryptographic Algorithms in Data Security

Cryptographic algorithms are mathematical formulas used to secure data through encryption and decryption. In data security, these algorithms ensure the confidentiality, integrity, and authenticity of information, protecting it from unauthorized access, tampering, or impersonation. There are two primary types of cryptographic algorithms:

- **Symmetric-key algorithms**, where the same key is used for both encryption and decryption (e.g., AES).
- **Asymmetric-key algorithms**, where two keys are used: a public key for encryption and a private key for decryption (e.g., RSA).

These algorithms form the backbone of most modern security systems, including secure communications, data storage, and authentication processes.

2.2 AES (Advanced Encryption Standard) and Its Evolution

AES (Advanced Encryption Standard) is a symmetric-key encryption algorithm developed to replace the outdated Data Encryption Standard (DES). AES was selected by the U.S. National Institute of Standards and Technology (NIST) in 2001 after a rigorous selection process. It operates on blocks of data (128 bits) and supports three key sizes: 128, 192, and 256 bits. AES evolved to address the increasing need for stronger encryption due to advances in computational power and cryptanalysis techniques. Today, AES is the encryption standard for a wide range of applications, from government communications to commercial data security.

2.3 Applications of AES in Different Sectors

AES is used across various sectors for securing data, from protecting personal information to safeguarding national security. Common applications include:

- **Financial Sector**: Encrypting sensitive financial transactions and protecting customer data in banking systems.
- **Healthcare**: Safeguarding patient records and ensuring compliance with health data protection regulations (e.g., HIPAA).
- **Government**: Securing classified communications, documents, and national security data.
- **Cloud Computing**: Ensuring the privacy of data stored and transmitted in cloud services.
- **Consumer Technology**: Securing personal devices, mobile apps, and online communications (e.g., HTTPS).

The wide adoption of AES is a testament to its reliability and versatility in securing various types of data.

2.4 Symmetric Encryption: Principles and Practices

Symmetric encryption is a cryptographic technique where the same key is used for both encryption and decryption. The principle behind symmetric encryption is that both the sender and receiver must share the secret key in a secure manner before communication. This key is used to transform plaintext into ciphertext and vice versa. The primary advantage of symmetric encryption, including AES, is its computational efficiency, making it suitable for encrypting large volumes of data. However, the main challenge is secure key distribution, as the key must be protected from exposure during transmission.

2.5 Key Generation and Management in Cryptography

Key generation and management are essential components of cryptographic systems, as the security of encryption depends on the strength and secrecy of the cryptographic key. In symmetric encryption (such as AES), the same key is used for both encryption and decryption, meaning that the key must be securely shared and stored. Key management includes processes like key generation (usually using random number generators), distribution, storage, and revocation. Secure key management practices are crucial to prevent unauthorized access and to ensure the ongoing confidentiality of encrypted data. Techniques like **Key Escrow**, **Key Rotation**, and **Public Key Infrastructure (PKI)** are used in managing and securing cryptographic keys.

2.6 Modes of Operation in AES Encryption

AES, as a block cipher, operates on fixed-size blocks of data (128 bits). However, since real-world data often exceeds this block size, AES can be used in different **modes of operation**, which define how blocks of data are encrypted and chained together. Common AES modes include:

- **ECB (Electronic Codebook)**: The simplest mode, where each block is encrypted independently. However, it is vulnerable to patterns and is not recommended for large datasets.
- **CBC (Cipher Block Chaining)**: Each plaintext block is XORed with the previous ciphertext block before encryption, providing more security but requiring initialization vectors (IVs).
- **CFB (Cipher Feedback)**: Encrypts data in smaller units than the block size, offering error propagation and stream cipher-like functionality.
- **OFB (Output Feedback)**: Similar to CFB, but the encryption of the previous block generates the next keystream, helping prevent error propagation.
- **CTR (Counter Mode)**: Converts a block cipher into a stream cipher by encrypting a counter value that is incremented for each block of data.

Each mode has its strengths and weaknesses, and the choice of mode depends on the specific use case and security requirements.

2.7 AES in the Context of Educational Data Security

In the educational sector, AES encryption plays a crucial role in protecting sensitive data such as student records, academic transcripts, diplomas, and certifications. Educational institutions are increasingly adopting AES encryption to ensure data confidentiality, prevent unauthorized access, and safeguard students' personal information. With the shift towards online learning and digital platforms, securing educational data becomes even more critical to avoid data breaches, identity theft, and fraud. AES ensures that digital certificates, student IDs, grades, and other personal information are protected both during storage and transmission, helping institutions comply with data protection regulations like **FERPA** (Family Educational Rights and Privacy Act) and **GDPR**.

2.8 Counteracting Security Threats and Vulnerabilities

Despite its robustness, AES encryption is not immune to potential threats and vulnerabilities. Common attacks against AES include **brute-force attacks**, where an attacker tries all possible keys to decrypt data, and **side-channel attacks**, which exploit weaknesses in the system's physical implementation, such as timing information or power consumption. To counteract these threats, various countermeasures are employed, including:

- **Key length**: Using longer keys (AES-192 or AES-256) increases security against brute-force attacks.
- **Secure key management**: Protecting encryption keys through hardware security modules (HSMs) or other secure storage methods.
- **Implementation techniques**: Ensuring that AES implementations are resistant to side-channel attacks through measures like constant-time algorithms.
- **Regular security audits**: Periodically reviewing cryptographic systems to identify potential vulnerabilities and weaknesses.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

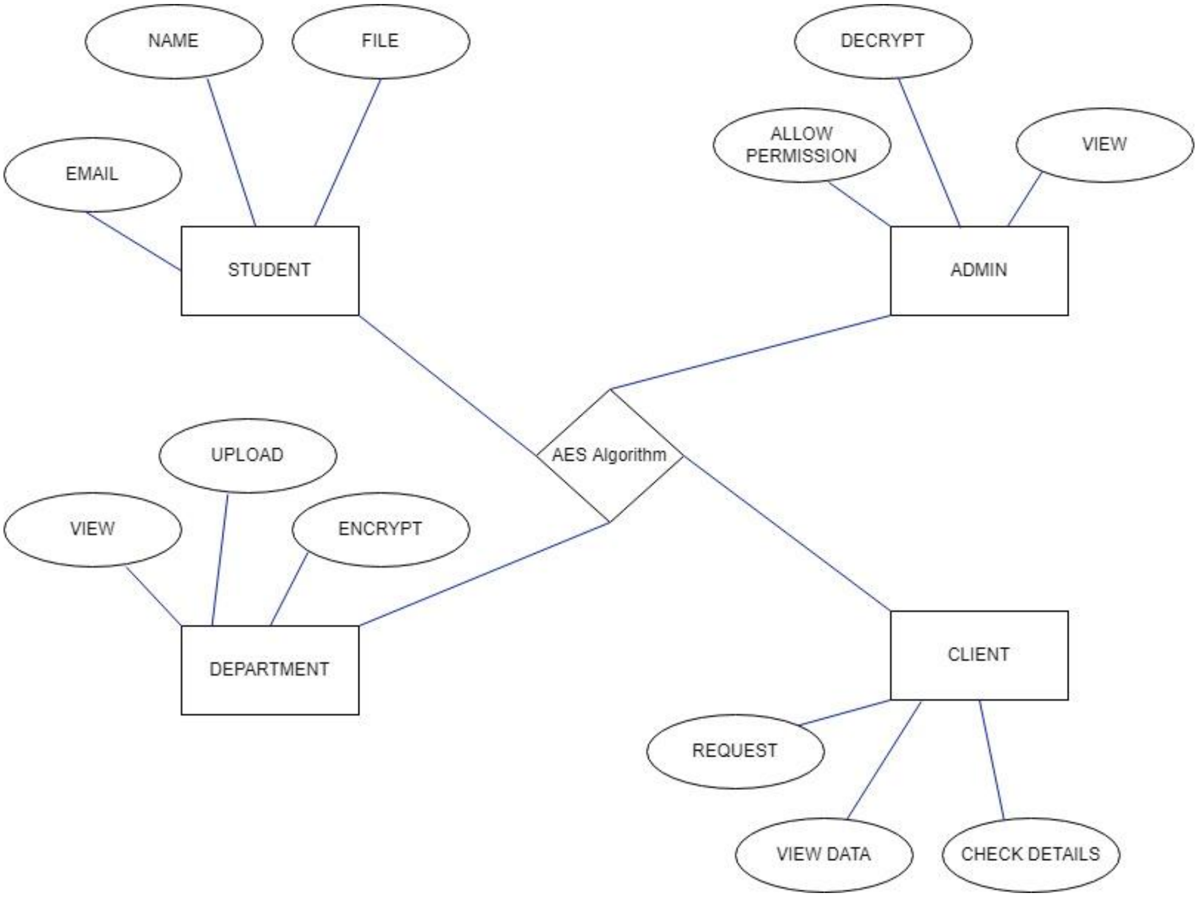

fig 3.1 : system architecture

- **End-to-End Encryption**: The entire pipeline from data collection to processing and classification remains encrypted, ensuring privacy.

- **No Exposure of Raw Data**: The raw data never leaves the encrypted form, preventing any statistical analysis from being performed on the unencrypted data.
- **Statistical Defense Mechanisms**: Techniques like adding noise and differential privacy ensure that adversaries cannot infer useful information from the encrypted dataset.
- **Access Control and Key Management**: Only authorized parties can decrypt the final results, maintaining strict control over sensitive information.

3.2 CLASS DIAGRAM

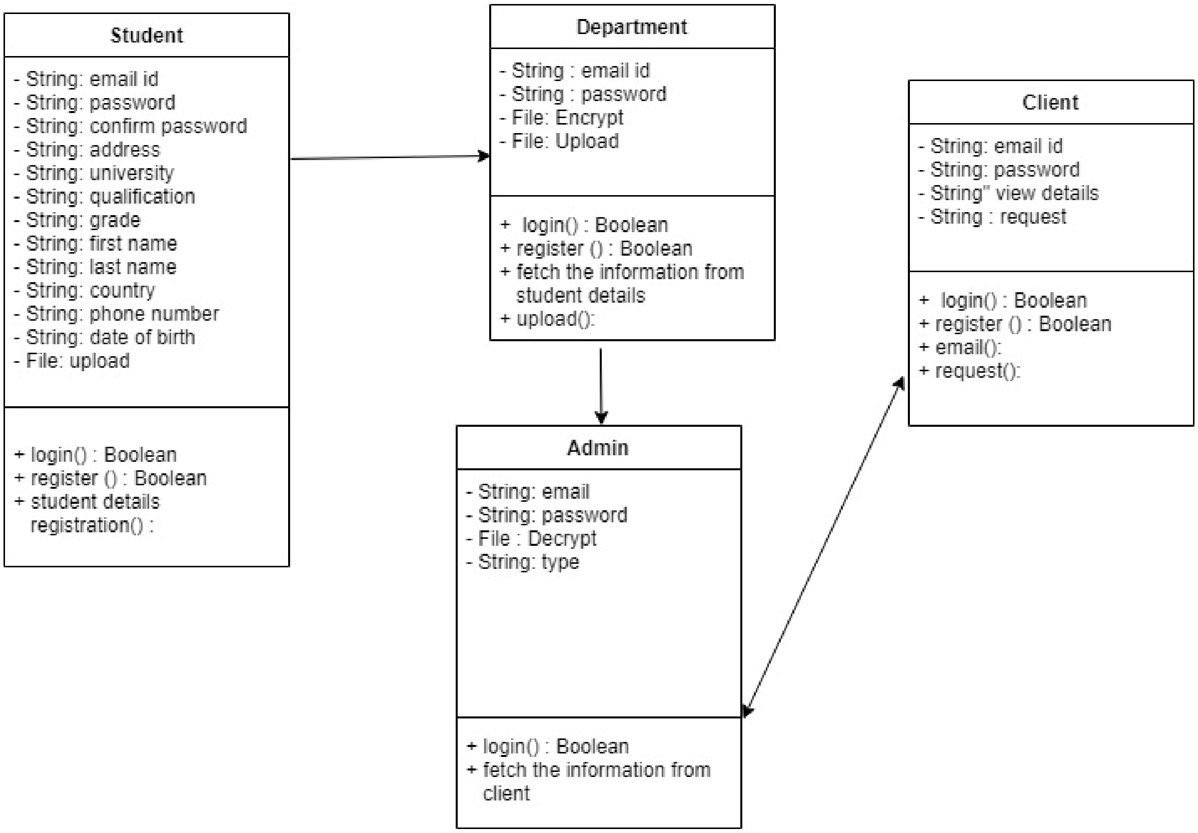

Fig 3.2 : class diagram

3.3 ACTIVITY DIAGRAM

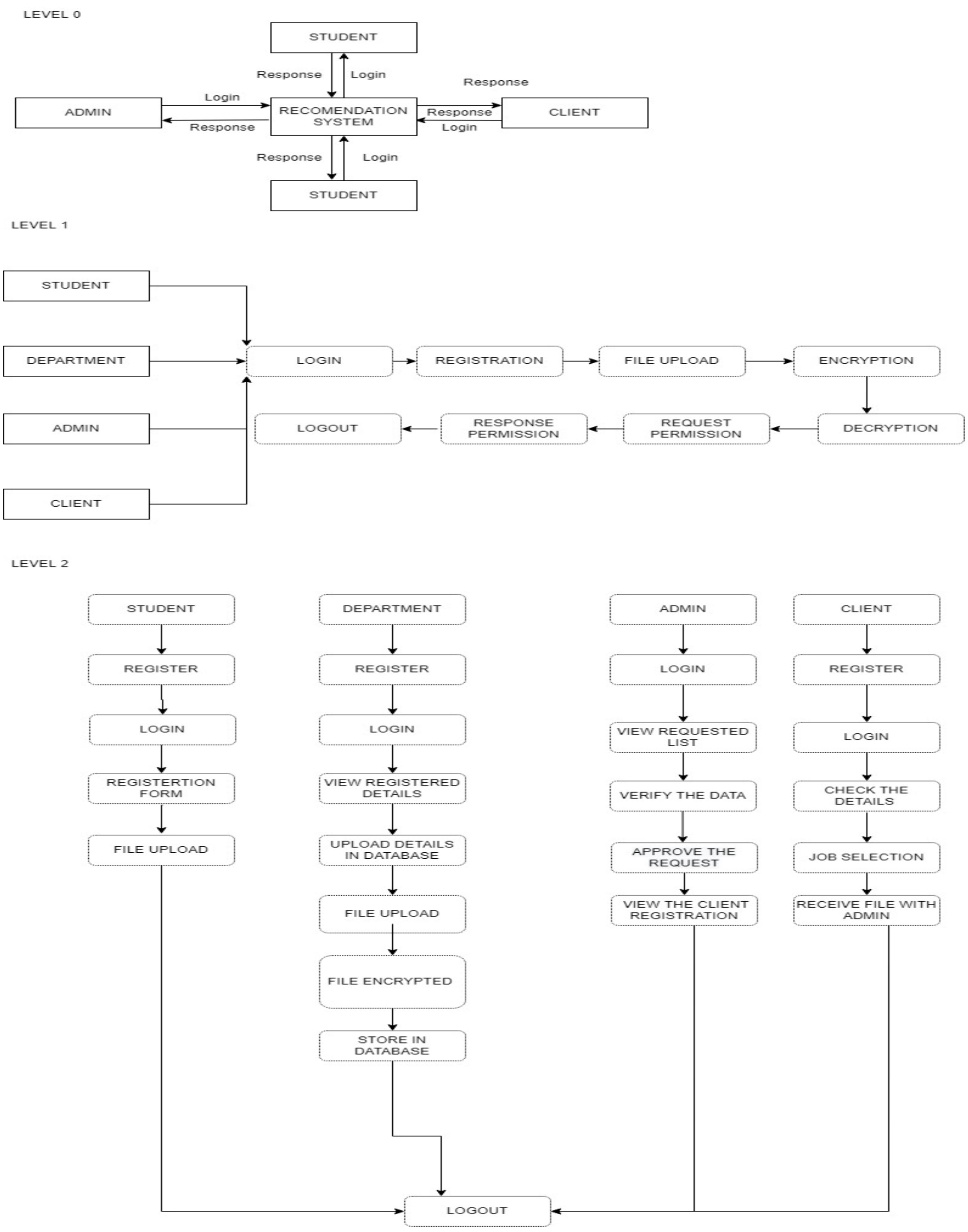

Fig 3.3 : activity diagram

An activity diagram for "Securing Data with Encrypted K-NN Defending Against Statistical Analysis" would typically map out the sequence of operations needed to protect data confidentiality during a k-nearest neighbor (K-NN) search, especially under the threat of statistical attacks.

3.4 SEQUENCE DIAGRAM

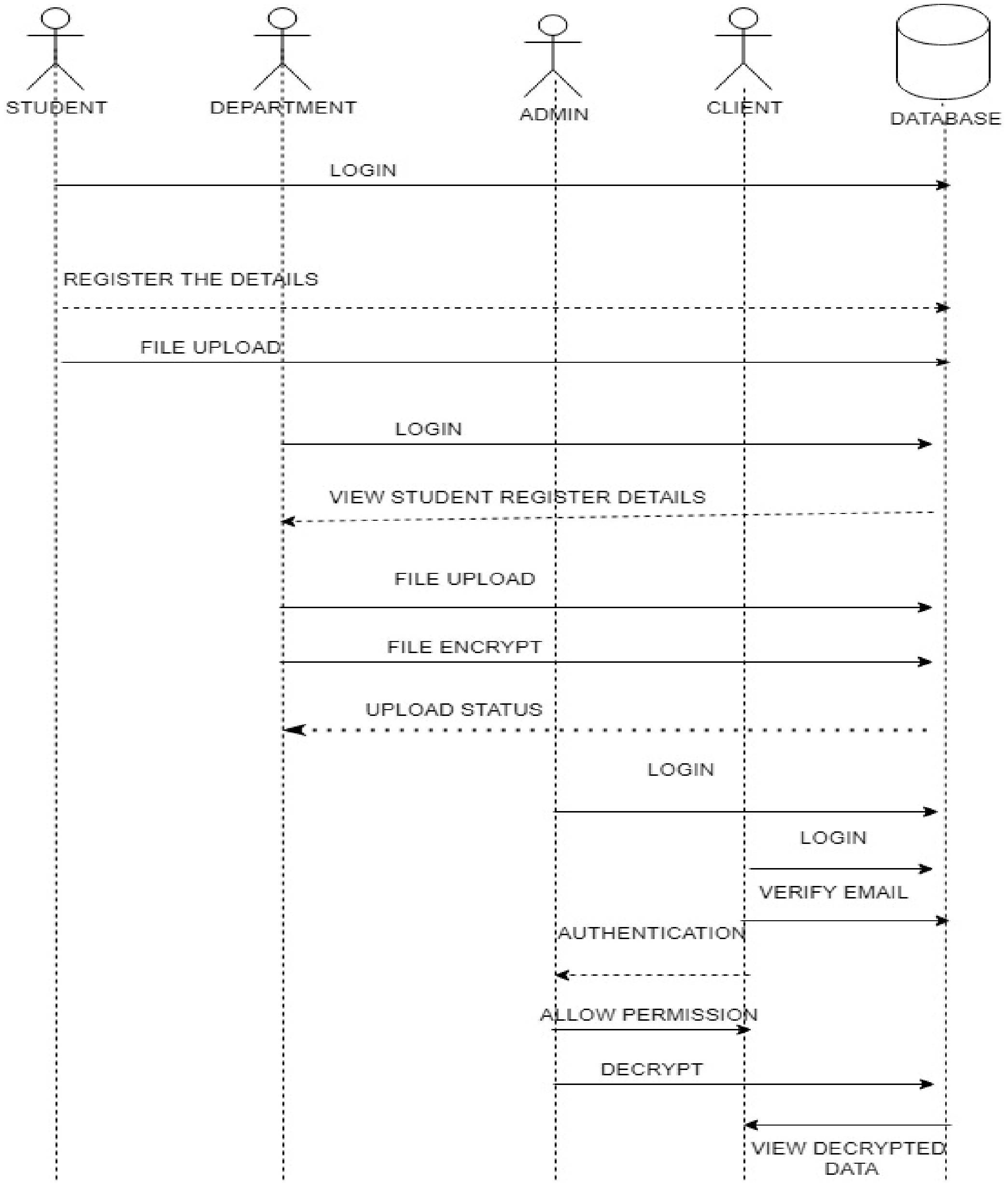

Fig 3.4 : sequence diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

3.5 USE CASE DIAGRAM

"Securing Data with Encrypted k-Nearest Neighbors (k-NN) Defending Against Statistical Analysis," a use case diagram would represent the primary actors, use cases, and their interactions with the system to perform secure k-NN classification while maintaining data privacy. Here’s a brief outline of how the use case diagram might look:

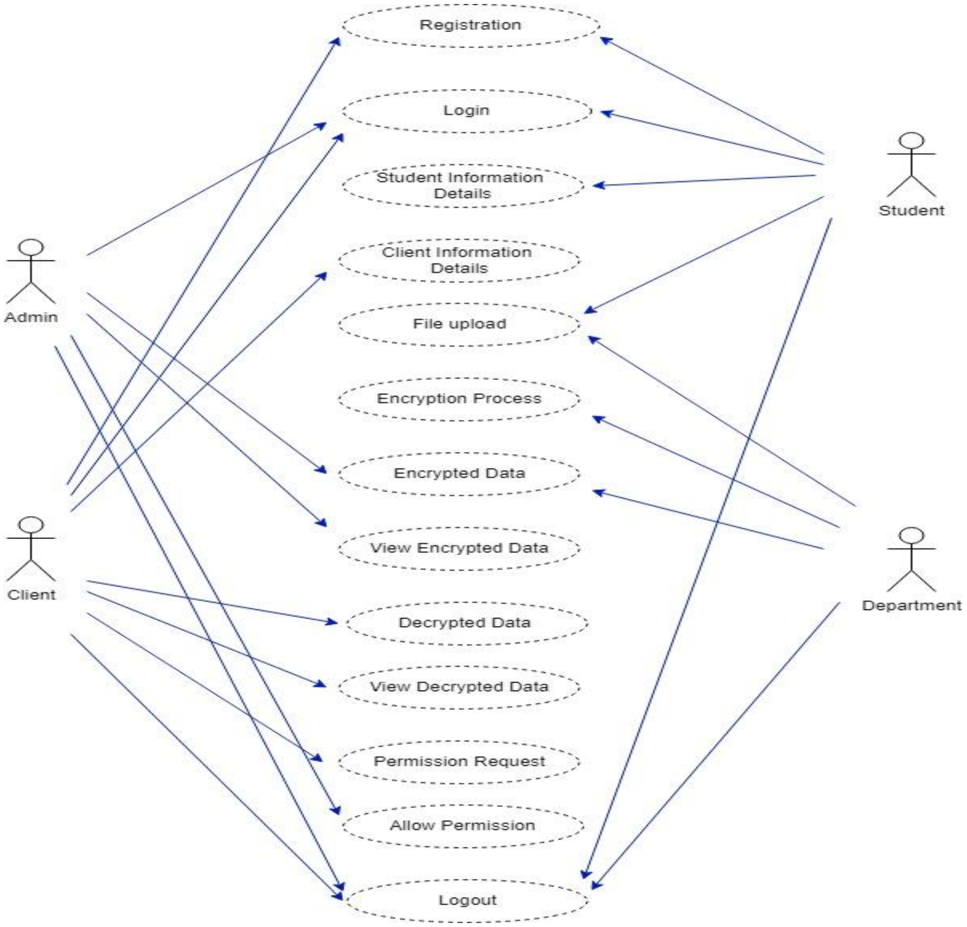

Fig 3.5: use case diagram

3.6 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A “DFD” is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

3.6.1 DFD-1

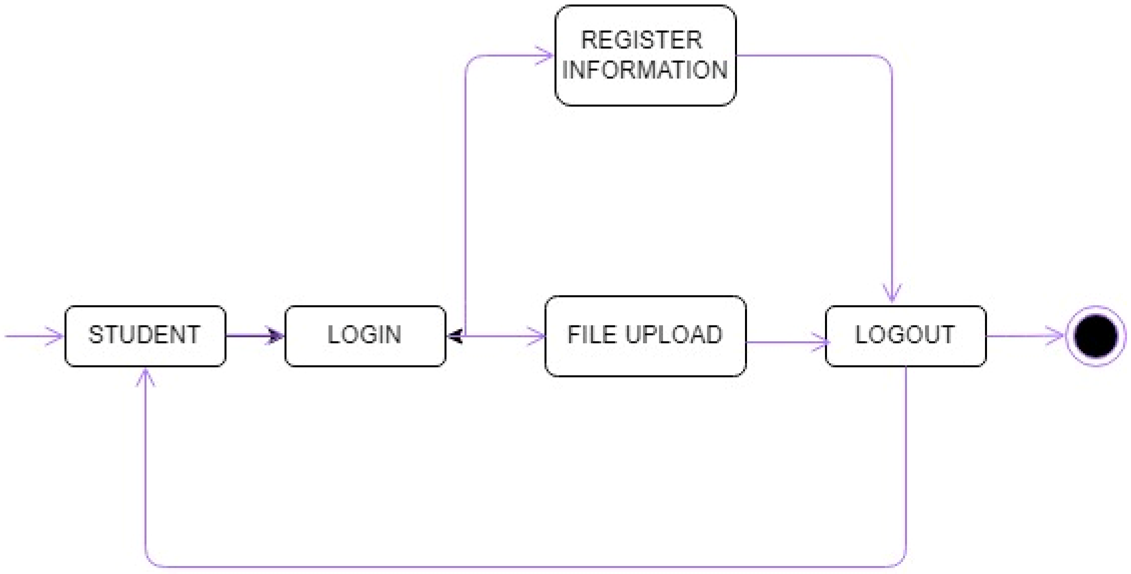

Fig 3.6.1: DFD-1

- **Raw Data → Data Encryption**: The raw data flows from the data owner to the encryption process.
- **Encrypted Data → Server Storage**: Once encrypted, the data flows to the server for secure storage.
- **User Query → Encrypted Query Submission**: The data user’s query flows to the encryption process and is sent to the server as an encrypted query.
- **Encrypted Data and Query → Encrypted k-NN Processing**: Encrypted data and the encrypted query flow into the k-NN processing module.
- **k-NN Result → Statistical Defense Application**: The initial k-NN result flows through the defense module to mitigate any possible statistical analysis.

- **Encrypted Result → Data User**: The final encrypted result is returned to the data user, who then decrypts it.

3.6.2 DFD-2

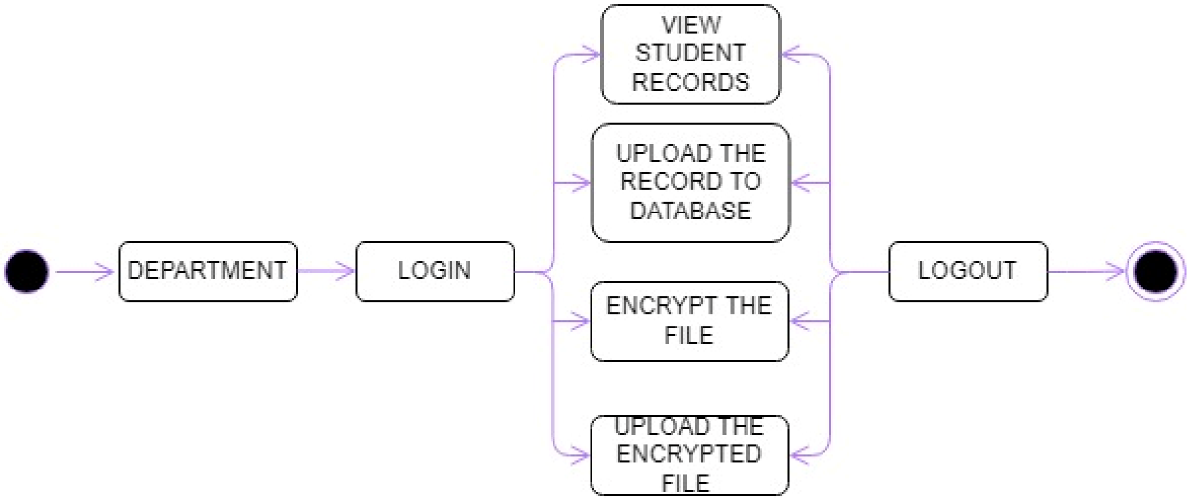

Fig 3.6.2 : DFD- 2

- **Data Owner**: Supplies the original dataset, encrypts it, and uploads it for secure storage.
- **Data User**: Submits encrypted queries to perform k-NN classification and retrieves encrypted results.
- **Server**: Stores encrypted data and processes encrypted k-NN queries.

3.6.3 DFD -3

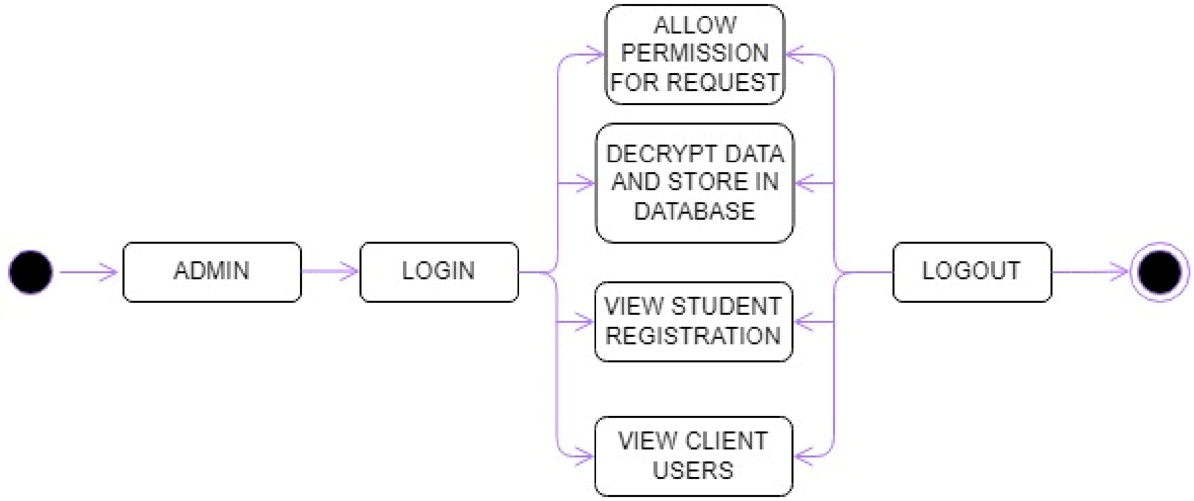
ADMIN
LOGIN
ALLOW PERMISSION FOR REQUEST
DECRYPT DATA AND STORE IN DATABASE
VIEW STUDENT REGISTRATION
VIEW CLIENT USERS
LOGOUT

Fig 3.6.3: DFD-3

- **Data Encryption**: The data owner encrypts the raw dataset to secure it against unauthorized access. This process ensures that sensitive data remains private before it is shared or stored.
- **Store Encrypted Data**: Encrypted data is stored on the server, making it accessible for k-NN processing without exposing raw values.
- **Submit Encrypted Query**: The data user encrypts their query and sends it to the server, ensuring that the query itself remains confidential.
- **Encrypted k-NN Processing**: The server performs k-NN classification on the encrypted data and query, calculating distances or similarities without decrypting any information.
- **Apply Statistical Analysis Defense**: The server implements techniques, such as secure encryption schemes or noise injection, to guard against statistical analysis of encrypted data. This step helps protect against inference attacks.
- **Return Encrypted Result**: The server sends back the encrypted classification result to the data user.
- **Decrypt Result**: The data user decrypts the result locally to view the classification outcome securely.

CHAPTER 4

PROJECT MODULES

4 . MODULES

The project consists of Four modules. They are as follows,

1. Student module
2. Department module
3. Admin module
4. Client module

4.1Student Module :

In this module, you have a register page before login to this module. Once the registration is completed. Then the student will log in through the login page with the registered email id and password. Once the student has logged in then the registration page will appear, which contains the email id, password, address, university, qualification, grade, first name, last name, country, mobile number, date of birth. Once the registration details are completed click submit, it will redirected to the next web page. After that, the file has been chosen from the local drive, it has been uploaded and that will be stored in the database, then it will be returns to the main page.

4.2 Department Module :

In this module the educational department will register using registration form details. Once the registration has been completed, the department will login with the registered email id and password to this module. Redirected page displays the student records for view the data, then the student file has been uploaded in an encrypted format using AES encryption algorithm implementation and it has stored in database.

4.3Admin Module :

In this module admin will login in the login page, Once the login has been completed it enter into the admin registration and stores in database. Then admin main page will displays menu option which contains the three options such as requested list, student registration records and client registration records. The requested list contains client requested records which contains file id, file name, date, status, and allow confirmation. Once admin has allowed then it shows the pop up message as it has been allowed. After these process it has been redirected to the main page of the admin. In main page it has the another menu option as a view details, if the menu is clicked then it has been redirected to the another page it shows the students registration records with the encrypted file also. It has a logout button to return to the main page. The main page has users menu, once it is clicked then it will redirected to the another page, it shows the total client registered records in an table format. These activities or operations has been performed in this module.

4.4Client Module :-

In this module Client it has a register page to login this module, once it has been completed then the client will login using registered email and password. It redirects to the email validation page for an authentication purpose. If the student is registered using that mail id then it redirected to the another page, which shows the entire details of the student and the client will select the job and expected salary package along with that they upload the resume as optionally. Once the process has been completed then it redirected to the another webpage. It contains the list of clients information and the client will receive the permission from admin, if the admin has been approved the decrypted data has been displayed with its file name, file id, semantic information and download option. If client has click the download option then the data has been downloaded and stored in system storage successfully.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 INTRODUCTION

This chapter involves the technology used, the hardware requirements and the software requirements for the project .

5.2 REQUIREMENTS

5.2.1 Hardware Requirements

- Hardware requirements:
- Processor : Intel (R) Pentium (R)
- Speed : 1.6 GHz and Above

- RAM : 4 GB and Above
- Hard Disk : 120 GB
- Monitor : 15’’ LED SVGA
- Input Devices : Keyboard, Mouse

5.2.2 Software Requirements

- Operating system : Windows 7 / 8 / 8.1 / 10
- Coding Language : JAVA / J2EE
- Java Version : JDK v8
- IDE : Eclipse Oxygen / Neon
- Database : MySQL v5.1
- Database Tool : HeidiSQL v11.0
- Application Server : Apache Tomcat 8.x / 9.x

5.3.1 Software description

5.3.1.1 java

Java is a set of computer software and specifications developed by Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. While they are less common than standalone Java applications, Java applets run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

5.3.1.2 Platform

The Java platform is a suite of programs that facilitate developing and running programs written in the Java programming language. A Java platform will include an execution engine (called a virtual machine), a compiler and a set of libraries; there may also be additional servers and alternative libraries that depend on the requirements. Java is not specific to any processor or operating system as Java platforms have been implemented for a wide variety of hardware and operating systems with a view to enable Java programs to run identically on all of them. Different platforms target different classes of device and application domains:

- Java Card: A technology that allows small Java-based applications (applets) to be run securely on smart cards and similar small-memory devices.

- Java ME (Micro Edition): Specifies several different sets of libraries (known as profiles) for devices with limited storage, display, and power capacities. It is often used to develop applications for mobile devices, PDAs, TV set-top boxes, and printers.
- Java SE (Standard Edition): For general-purpose use on desktop PCs, servers and similar devices.
- Java EE (Enterprise Edition): Java SE plus various APIs which are useful for multi-tier client–server enterprise applications.

The Java platform consists of several programs, each of which provides a portion of its overall capabilities. For example, the Java compiler, which converts Java source code into Java bytecode (an intermediate language for the JVM), is provided as part of the Java Development Kit (JDK). The Java Runtime Environment (JRE), complementing the JVM with a just-in-time (JIT) compiler, converts intermediate bytecode into native machine code on the fly. The Java platform also includes an extensive set of libraries.

The essential components in the platform are the Java language compiler, the libraries, and the runtime environment in which Java intermediate bytecode executes according to the rules laid out in the virtual machine specification.

5.3.1.3 Java Virtual Machine

The heart of the Java platform is the concept of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what hardware or operating system the program is running under. There is a JIT (Just In Time) compiler within the *Java Virtual Machine,* or JVM. The JIT compiler translates the Java bytecode into native processor instructions at run-time and caches the native code in memory during execution.

The use of byte code as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter.

Although Java programs are cross-platform or platform independent, the code of the Java Virtual Machines (JVM) that execute these programs is not. Every supported operating platform has its own JVM.

5.3.2 JAVA FXML

FXML is an XML-based user interface markup language created by Oracle Corporation for defining the user interface of a JavaFX application. It provides a convenient alternative to constructing such graphs in procedural code, and is ideally suited to defining the user interface of a JavaFX application, since the hierarchical structure of an XML document closely parallels the structure of the JavaFX scene graph. However anything that is created or implemented in FXML can be expressed using JavaFX directly.

CHAPTER 6

CONCLUDING REMARKS

6.1 CONCLUSION

In conclusion, the project "Securing Data with Encrypted k-Nearest Neighbors (k-NN) Defending Against Statistical Analysis" provides a secure framework for performing k-NN classification on sensitive data while ensuring privacy and preventing unauthorized access. By using advanced encryption techniques, such as homomorphic encryption or other privacy-preserving methods, the system enables secure computation on encrypted data, allowing classification to occur without revealing raw data or user queries.

Furthermore, statistical analysis defenses, like noise injection or obfuscation, are incorporated to protect against inference attacks that could otherwise exploit patterns in encrypted data. This approach ensures that both the data owner and data user can operate within a secure environment, maintaining data confidentiality throughout the storage, processing, and retrieval stages.

Overall, the project demonstrates an effective solution for privacy-preserving data mining in scenarios where data sensitivity is paramount. This framework is applicable to fields like healthcare, finance, and secure communications, where sensitive data needs to be processed securely and with robust defenses against statistical attacks.

REFERENCES

1. Min Deng, Xinbo Gao, "Achieving Practical and Privacy-Preserving kNN Query over Encrypted Data," IEEE, 2024.

2. Zekeriya Erkin, "Privacy-Preserving kNN Computation on Encrypted Databases," IEEE Transactions on Information Forensics and Security, 2014.

3. Chang Liu, Shucheng Yu, "Efficient Privacy-Preserving k-Nearest Neighbor Search over Encrypted Data in Cloud," IEEE, 2015.

4. Cheng Wang, Jianwei Niu, "A Secure kNN Query Scheme for Privacy Protection in Cloud Computing," IEEE, 2017.

5. Qi Liu, Xueli Yang, "Secure kNN Query Processing Framework Based on MapReduce," IEEE Access, 2018.

6. Hongwei Zhu, Li Li, "Privacy-Preserving kNN Query on Encrypted Cloud Data," IEEE, 2020.

7. Jiansheng Liu, Xiaohua Wang, "PRkNN: Efficient and Privacy-Preserving Reverse kNN Query for Cloud Computing," IEEE, 2021.

8. He Zhang, Tian Luo, "Privacy-Preserving Boolean kNN Query Over Cloud-Based Spatial Data," IEEE, 2022.

9. Zhiqiang Liu, Tao Zhang, "A k-Nearest Neighbor Algorithm Based on Homomorphic Encryption," IEEE, 2022.

10. Shucheng Yu, Hongwei Zhu, "Secure Optimal k-NN on Encrypted Cloud Data," IEEE, 2022.

11. Qi Liu, Xinbo Gao, "Efficient Privacy-Preserving kNN Query and Classification Scheme Using k-Dimensional Tree," IEEE, 2023.

12. Kai Li, Yonggang Wen, "Efficient k-Nearest Neighbor Classification over Encrypted Data," IEEE, 2023.

13. Min Deng, Zekeriya Erkin, "Efficient and Privacy-Preserving k-NN Query for Cloud-Based Services," IEEE, 2022.

14. Fan Zhang, Jianwei Sun, "Secure Reverse kNN Query in Encrypted Data Clouds," IEEE, 2023.

15. Chang Liu, Xiaowei Zhang, "Privacy-Preserving and Efficient kNN Query Processing on Encrypted Cloud Data," IEEE, 2023.