

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

MINI PROJECT REPORT FOR

Encryption System

By

SHAKKIRA.K

Enrolment no. 126596130

Under the Guidance of

Dr. KABEER

COURSE COORDINATOR

AL FAROOK, FROOK COLLEGE, KOZHIKODE

Submitted to the school of Computer and Information Sciences

In partial fulfillment of the requirements for the degree of

Master of Computer Applications (MCA)

SEMESTER –IV (MCS-044)



**Indira Gandhi National Open
University, MaidanGarhi**

New Delhi - 110068

Encryption System



**SCHOOL OF COMPUTER AND INFORMATION
SCIENCES**

IGNOU, MAIDAN GARHI, NEW DELHI- 110 068

PERFORMA FOR SUGGESTIONS OF MCS-044 PROJECT PROPOSAL

Enrolment No : 126596130
Study Center : Al-Farook Educational Center
Regional Center : Vadakara, **RCCode** : 83
E-mail : shakkira.shaki@gmail.com
Telephone No : 9567398299

1. Name and Address of the Student: SHAKKIRA.K,
KACHAKKARAN

(H),KIZHAKKUMPARAMBA,MALAPPURAM (DST)

2. Title of the Project: Encryption System

3. Name and Address of the counselor : Dr. V.KABEER

4. Educational Qualification of the counsellor

5. Working /Teaching experience of the counsellor:

6. Software used in the project:jdk 7, NetBeans IDE 7.4

Signature of the student
counsellor

Signature of the

Date:

Date:

Suggestions for improving the Project :

CERTIFICATE

This is to certify that the project report entitled “**Encryption System**” submitted to **Indira Gandhi National Open University** in partial fulfillment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATION (MCA)** is an original work carried out by Miss. **SHAKKIRA**. Enrolment No **126596130** under my guidance. The matter embodied in this project is authentic and is genuine work done by the student and has not been submitted to whether to this university or to any other university /institute fulfillment of the requirement of any course of study.

Signature of the Student:
Counsellor

Signature of the

Date:

Date:

Name and Address of the
Student:

Name, Designation and
Address of the counsellor:

SHAKKIRA.K
KACHAKKARAN(HOUSE)
PANDALLUR (PO),KIZHAKKUMPARAMBA
MALAPPURAM
Enrolment No:126596130

Encryption System

Name of the Project : Encryption System Date:

Name of the Team Member	Role	Tasks and Responsibilities	
1. Shakkira	Data manager	Report writing	
2. Subeesh.U.P	Programmer	Program development	
3. Libin	Coordinator	Data collection and coordination	
4. Vinaya	Auditor	Testing and implementation	

Name and signature of the Project Team Members:

1. Shakkira

2. Subeesh

3. Lipin

4. Vinaya

Signature of the counsellor

Date

ABSTRACT

The title of our project is “ENCRYPTION SYSTEM”. This project encrypts and decrypts the files by using Advanced Encryption Standard (AES) algorithm. Our aim is to develop the software named ENCRYPTION SYSTEM that encrypts and decrypts the files by using Advanced Encryption Standard (AES) algorithm. Encryption and Decryption is strong file encryption software for personal and professional security. It protects privacy of our documents and sensitive files by encrypting them using Advanced Encryption Standard (AES) algorithm to provide high protection against unauthorized data access.

In today's world the networking plays a very important role in our life. Most of the activities occur through the network. For the safe and secured exchange of information, we need to have security. The encryption has very wide applications for securing data. Encryption refers to set of algorithms, which are used to convert the documents and any files to code or the unreadable form of files, and provides privacy. To decrypt the file to receiver uses the “key” for the encrypted files.

This project work helps you to understand what cryptography is all about and the procedures used to achieve this aim, it explains the design and implementation of computer security: data encryption and decryption and AES algorithm. The programming language used in the development of this project is java. The Java platform refers to a group of software products from Sun Microsystems. The platform is used to develop and run Java programs. The platform includes the execution engine (called a Java Virtual Machine) that allows Java programs to do the same thing on different computer systems.

ACKNOWLEDGEMENT

With all praises to the almighty God whose abundant Grace and Mercies enabled me to complete this project. I would like to express my profound gratitude to all the people who have inspired and motivated me to make this project a success.

I dedicate this project to my parents for the inspiration, strength and blessing endowed upon me.

I would like to express my sincere thanks to Dr.V.Kabeer Course Coordinator for his invaluable suggestion given at every step. I think him for all his encouragement, inspiring, guidance, advice and suggestion throughout my project work.

Thanks are also due to all my faculties and friends, in my college for their timely help during the tenure of the project. Once again I think one and all who have helped me directly and indirectly in the successful completion of the project work.

Thanking
you

TABLE OF CONTENTS

TITLE	PAGE
Abstract	5
Acknowledgement	6
Table of content	7
CHAPTER 1: INTRODUCTION	
1.1 Background	10
1.2 Objectives	11
1.3 Purpose, Scope & Applicability	12
1.3.1 Purpose	
1.3.2 Scope	
1.3.3 Applicability	
1.4 Achievements	
1.5 Organization of Report	14
CHAPTER 2: SURVEY OF TECHNOLOGIES	15
CHAPTER 3: REQUIERMENT AND ANALYSIS	16
3.1 Problem Definition	
3.2 Requirement Specification	
3.3 Planning and Scheduling	

3.4 Software and Hardware Requirements

3.5 Preliminary Product Description

3.6 Conceptual Model

CHAPTER 4: SYSTEM DESIGN

40

4.1 Basic Modules

4.2 Procedural Design

4.2.1 Logic Diagram

4.2.2 Data Structure

4.2.3 Algorithm Design

4.3 User Interface Design

4.4 Security Issues

4.5 Test Case Design

CHAPTER 5: IMPLEMENTATION AND TESTING

48

5.1 Implementation Approaches

5.2 Coding Details and Code Efficiency

5.2.1 Code Efficiency

5.3 Testing Approach

5.3.1 Unit Testing

5.3.2 Integrated Testing

5.4 Modification and Improvements

CHAPTER 6: RESULT AND DISCUSSION

84

6.1 Test Reports

6.2 User Documentation

CHAPTER 7: CONCLUSION

87

7.1 Conclusion

7.2 Limitations of the System

7.3 Future Scope of the Project

REFERENCES

GLOSSARY

APPENDIX A

APPENDIX B

LIST OF FIGURES

Figure 1: Conceptual Models	35
Figure 2: Use Case Diagram	39
Figure 3: Encryption	42
Figure 4: Decryption	43
Figure 5: Data Structure	44

CHAPTER 1: INTRODUCTION

1.1 Background

As the computers and networked systems increases in the world of today, the need for increased data security also becomes necessary and important. The development of computer network system has exposed many networks to various kinds of authentication threats and with these exposures; one can see that the need for increased network and data security is vital and important in every organization.

The security may include identification, authentication and authorization of user in a network environment. Some cryptography technique may be used to solve all these problems.

Cryptography technique has reduced unauthorized access of data. Encrypted data may be stored in to file. Then authorized person are supposed to decrypt this data with the help of some keys.

1.2 Objectives

The main objective of our project is to encrypt or decrypt the any files for personal and professional security. Encryption and Decryption protects privacy of our documents and sensitive files by encrypting them using Advanced Encryption Standard (AES) algorithm to provide high protection against unauthorized data access.

In today's world the networking plays a very important role in our life. Most of the activities occur through the network. For the safe and secured exchange

of information, we need to have security. The encryption has very wide applications for securing data. Encryption refers to set of algorithms, which are used to convert the documents and any files to code or the unreadable form of files, and provides privacy. To decrypt the file to receiver uses the “key” for the encrypted files.

If you want to send sensitive information via email, simply paste the encrypted text or any files into your email or attach the [encrypted file](#).

All the recipient has to do is to decrypt your text or any file. Encryption and Decryption works with text information and any files. Just select what you want to encrypt, and Encryption and Decryption software helps you keep documents, private information and files in a confidential way.

The project has the following objectives

1. Storing important information in encrypted form ensuring security.
2. We can prevent information loss when system crashes occurred.
3. The information will be recovered from the backup data.
4. Enhancing efficiency of data retrieval.
5. File Sending.
6. Better accuracy and improved consistency.
7. Help facility will be provided.

8. To understand and improve the computer data security through encryption of data.

9. To enhance the integrity of data.

10. To develop a platform to complement physical security.

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

In today's world most of the communication is done using electronic media. Data security plays vital role in such communication. Hence, there is a need to protect data from malicious attacks. This can be achieved by cryptography. The earlier encryption algorithm is Advanced Encryption Standard (AES) which has several loopholes such as small key size and sensible to brute force attack etc. These loopholes overcome by a new algorithm called as Advanced Encryption Standard Algorithm.

1.3.2 Scope

The scope of our project is presently specific. Both the sender and the receiver must have this software installed on their systems to encrypt or decrypt and compress or decompress the files transmitted between them. This includes all the users who want to interact electronically, whether it is through emails, sending a files etc.through local area network in order to keep their private information confidential.

- Each step is clearly stated and user will not face any ambiguity in using the software.
- The software provides clarity in its functionality even to naïve users.
- No complexity is involved.

- The various scope which cryptographic algorithms guarantees certain level of security, confidentiality and integrity of data.

1.3.3 Applicability

Encryption and Decryption Application Components provide the integration of software or hardware encryption/decryption operational modules to customize the various security control application services required for the transaction systems. The components contain the platform technologies, such as JAVA and .NET, as well as security control components and ability to authorize the authentication servers, such as Advanced Encryption Standard (AES) algorithms.

Attackers can use development tools, intended for tasks such as application monitoring or debugging, to gain access to encryption keys or simply to turn off encryption, unlocking information within the application.

1.4 Achievements

Goal of Encryption

- Conveys confidentiality to messages while in transit.
- Changes readable file into something that cannot be read.
- Discourages anyone from reading or copying the files.
- Encryption algorithms are considered secure if the security depends on only one factor - key length.

- Security does not depend on secrecy, inaccessibility, or anything else, only on the key length.
- If this factor is true, then the only possible attack against the algorithm is a brute force attack.

1.5 Organization of Report

The 1st chapter the basic definition of terms that are used this report and purpose of project and also gives the motivation behind implementing this project.

The 2nd chapter gives the details of survey of technologies this project.

The 3rd chapter gives the details of requirement and analyzing the project. It gives hardware, software and user requirements.

The 4th chapter gives the details of each modules used in this project.

The 5th chapter gives some implementation details i.e. how we implemented it, and which methods perform what operation etc.

The 6th chapter gives test reports and user documentation in this project.

The 7th chapter gives conclusion, limitations and further scope of the project.

Reference section provides source detail where we get information. It gives name books, authors, year of published etc. and also some website for implementation.

Appendix A contain source code Appendix B contain screen shorts of the project.

CHAPTER 2: SURVEY OF TECHNOLOGIES

There are a wide variety of languages available for doing this project like C, C++, C#, VC++, JAVA etc.

Java is a language that makes application development easier for distributed computer environment. Programs are executed by multiple computers across networks. Java is also capable of meeting the challenges of application development in the context of heterogeneous, network wide distributed environment. Paramount among these challenges is secure delivery of applications that consume the minimum of system resources, can run on any hardware and software platform can be extended dynamically.

While reviewing other languages like C, C++, C#, VC++ we understood these languages are convenient, but Java is a more convenient to implement a network. The java language security which is a highly great software development technology. Java is a general-

purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. The Java platform refers to a group of software products from Sun Microsystems. The platform is used to develop and run Java programs. The platform includes the execution engine (called a Java Virtual Machine) that allows Java programs to do the same thing on different computer systems.

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

Project Mission

The aim of our project is to develop software named “ENCRYPTION SYSTEM”. The project encrypts and decrypts the any files using Advanced Encryption Standard (AES) algorithm to maintain the security and integrity of data and information and to provide high protection against unauthorized data access.

Target

Our target is the common man who wants to interact client and server, whether it is through messages, documents and any files etc. Through network via file sending sensitive messages or documents over the network is very dangerous. So, our project helps him to interact in a safe and secure manner in order to keep their private information confidential.

Target Users

The main target users of our project are the people who transmit confidential information network via file sending sensitive messages or documents through the client server interaction.

Scope and Key Elements

The scope of our project is presently specific. Both the sender and the receiver must have this software installed on their systems to encrypt/decrypt and compress/decompress the files transmitted between them. The system provides the security and integrity of data or information.

- It will provide a more clear and non-ambiguous description of the functions.
- The system is highly user friendly.
- It uses secret keys for encryption and decryption.
- The software provides clarity in its functionality even to naïve users.

Analysis is the detailed study of the various operations performed by a system and their relationships within and outside the system. System analysis is an approach to study the system and find a solution. It provides a framework for visualizing the organizational facts that operate on a system. The aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider.

Encryption System

During the analysis the data are collected from the available resources for analysis. Information from the existing system, dataflow diagram, on site observations and interviews are the various tools used during the analysis to collect information. The natural problem solving process consists of the following steps.

1. The identification of the problem situation that needs to be solved.
2. Defining the problem.
3. Defining the desired outcome.
4. Provide possible alternative solution.
5. Identifies and selects the best one among them.

Identification of Need

This network project aims to achieve storing of most important information in encrypted form. They rely on a secret piece of information called the key. Hacker's objective is to obtain the key from the communication. The various scope which cryptographic algorithms guarantees certain level of security, confidentiality and integrity of data.

There is great need for encryption, authentication techniques, user identification and passwords. These will all protect your computer from damage or from hackers (people who gain access to your computer and then illegally steal your data.).

Encryption is only one method which can be used when you want to be able to protect your computer from harm and hackers. Encryption is when you put a secret key or password on your work that will then transform your data into an unreadable file if you don't have the secret key and password. It is best to encrypt something that you will transfer over the network because it has high

chances of being stolen because it is rather easy when it is being transferred so much from computer to computer. Now the best way to transfer this secret key is face to face because otherwise then the secret key might be taken by someone else will you transfer it across the network. Encryption is a pretty easy thing to do but if one of your documents really needs to be encrypted for its safety it is extremely easy to find software to help you increase your security.

Feasibility study

After investigation it is essential to determine whether the project is feasible or not. In feasibility study is tested whether the system to be developed would be able to accomplish its task on the working grounds. Its impact was also found to be not adverse. It was found that the user's requirements would be met and the resources would be used in an effective manner. In feasibility study the important aspects related to the project were considered like the problem definition and the process for solution. The cost and benefit analysis was also done. Essential features involved in the feasibility analysis are:-

- Behavioral Feasibility
- Economic Feasibility
- Technical feasibility
- Operational Feasibility

Behavioral Feasibility

An estimate should be made of how strong a reaction the users is likely to have toward the development of a new system. It is common that the new system requires special effort to educate, sell and train the users. But in the case of the Virtual Drive, anyone who has a basic knowledge of computer can easily work with this system so the users do not feel any difficult with this, so they can accept the system without any willingness.

Technical Feasibility

Technical feasibility revolves around the technical support of the project. The main infrastructure of the project included the project labs in the college campus. The systems there were easily able to absorb the new software being installed. The project thus was technically feasible. The equipment and the software produced no problem. The project's technical requirements were met. The project could be made to work correctly, fulfilling its task, with the existing software and personnel.

Economic Feasibility

The project developed, Encryption and Decryption was within budget and producing the desired results. The labor or the human ware consisted of the four group members of our project. The output consisted of getting the desired results. Thus with the consideration of the inputs, the outputs were achieved successfully. The project was within limit. The inputs didn't overdo the outputs.

Operational Feasibility

Operational Feasibility aims to determine the impact of the system on the users. The system developing has an influence on its

users. Our system “Advanced Encryption System” was new for them but it was simple enough for any naïve person to understand. The evolution of this new system required no special training for the users. Encryption and Decryption was found to be feasible in this regard. The system developed would be user friendly and no complexities would be involved in its functionalities.

3.2 Requirements Specification

The main aim of preliminary analysis is to identify the problem. First, need for the new or the enhanced system is established. Only after the recognition of need, for the proposed system is done then further analysis is possible.

Existing System

In the existing system, the encrypted key is send with the document; any user can view the encrypted document with that key. It means the security provided for the encryption is not handled properly. And also the Key byte (encrypted key) generate with random byte. Without the user interaction the Key byte is generated. As observed the current encryption/decryption software’s doing the encryption and decryption task are all very complicated in their functionality. The method of encryption/decryption and key generation of current system for a new user to understand is complex in nature.


Drawbacks

Some of the drawbacks are:

1. Lack of security
2. Key byte is generated without user interaction

Proposed System

The proposed system is quite simple to use. It is not complex in its functionalities. It is easy for a naïve user to use it.

If you want to send sensitive information via network, simply paste the [encrypted file](#) 

. All the recipient has to do is to decrypt your file. Encryption and Decryption works with any information and files. Just select what you want to encrypt, and Encryption and Decryption software helps you keep documents, private information and files in a confidential way. The proposed system will help the user to reduce consuming time. The system requires very low system resources and the system will work only in network connections.

Benefits

1. Security is enhanced in well manner.
2. Users set the byte key manually.

Software Requirements Specification

Software Requirements Specification (SRS) is the starting point of the software development activity. Little importance was given to this phases in the early days of software development. The emphasis was first on coding and then shifted to design.

The purpose of this project is to demonstrate how some of the more popular encrypting algorithms. The system will allow the user to enter any files, select an encryption method, and then view the file encrypted by the selected algorithm. When an encryption method is selected, options pertaining to that specific algorithm will be displayed for the user to customize. When the user presses the "Encrypt" button, the algorithm will then begin encrypting the files. After each

calculation, a file will be displayed to the user relating to the operation that has just been performed. These files will allow the user to understand the method used in the encryption algorithm. When the algorithm is finished, the user will have the file encrypted according to the method selected and options selected.

Some of the difficulty is due to the scope of this phase. The software project is imitated by the client needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify the requirements by talking to these people and understanding their needs. In situations where the software is to automate a currently manual process, most of the needs can be understood by observing the current practice.

The SRS is a means of translating the ideas in the minds of the clients (the input) into formal document (the output of the requirements phase). Thus the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input has none of these properties.

Performance Requirements

The project must meet the end user requirements. Accuracy and fast must be imposed in the Project. The project is developed as easy as possible for the sake of end user. The project has to be developed with view of satisfying the future requirements and future enhancement. The tool has been finally implemented satisfying the needs specified by the company. As per the performance is concerned this system said is performing. This processing as well as time taken to generate well reports were also even when large amount of data was used.

Security Requirements

Web application are available via network access, it is a difficult. If not possible, to limit the population of the end-user who may access

the applications? In order to product sensitive connect and provide secure mode be implemented throughout the infrastructure that the supports web application and within the application itself.

Design Requirements

To create project, add base masters and masters to the project, assign behaviors to the master, create and assign behavior sets, and then apply, test and validate those behaviors. It also shows how to create and build a stencil to hold the shapes.

Quality and Reliability Requirements

A software component that is developed for reuse would be correct and contain no defects. In reality, formal verification is not carried out routinely, and defects can add to occur. However, with each reuse, defects are found eliminated, and a components qualify improve as a result. Over time the components virtually defect free.

Software reliability is defined in statically term as” the probability of faultier-free operation of a computer program in a specified environment for specified tine”. The software quality and reliability, failure is nonconformance to software requirements. Failure can be only anything or catastrophic. One failure can be corrected within seconds while another requirements week even mouths to correct. Complicating the issue even further, the correction of the one failure may in fact result in the introduction of the errors that ultimately result in other failure.

3.3 Planning and scheduling

Encryption System

Project planning is part of project management .which relates to the use of schedule, such as Gantt chart to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the duration for the various task necessary to complete the work are listed and grouped into a work break down structure. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the path. Considering the total available time I had prepared a plan and schedule which is given below.

Sl.No	Duration	Activity
1	Sep First 1 week	Problem Understanding
2	Sep Last 1 week	Software Requirement
3	Oct First 1 week	Requirement Analysis
4	OCt Second 1	Data Collection
5	week	Structured Analysis
6	Oct Third 1 week	Input Design
7	Oct Last 1 week	Output Design
8	Nov First 1 week	Coding
9	Nov Second 1	Unit testing
10	week	Module testing
11	Nov Third 1	System testing

Encryption System

12	week Nov last week Dec First week Dec last week	Implementation
----	--	----------------

TASK SCHEDULING (GANTT CHART)

TASKS	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
REQUIREMENT GATHERING						
DESIGN						
TEST CASES						
CODING						

About Windows XP:-

Windows XP is an operating system produced by Microsoft for use on personal computers, including home and business desktops, laptops and media centers. It was first released in August 2001, and is the most popular version of Windows based on installed user base. The most common editions of the operating system are Windows XP Home Edition, which is targeted at home users, and Windows XP Professional, which offers additional features such as support for Windows Server Domain and two physical processors, and is targeted at power users, business and enterprise clients.

The Windows GUI:-The familiar graphical user interface it presents to the world.

About Java

Java is a high-level, third generation programming language, like C, FORTRAN, Smalltalk, Perl, and many others. You can use Java to write computer applications that play games, store data or do any of the thousands of other things computer software can do. Compared to other programming languages, Java is most similar to C. However although Java shares much of C's syntax, it is not C. Knowing how to program in C or, better yet, C++, will certainly help you to learn Java more quickly, but you don't need to know C to learn Java. A Java compiler won't compile C code, and most large C programs need to be changed substantially before they can become Java programs. What's most special about Java in relation to other programming languages is that it lets you write special programs called applets that can be downloaded from the Internet and played safely within a web browser. Java language is called as an Object-Oriented Programming language

and before beginning for Java, we have to learn the concept of OOPs(Object-Oriented Programming).

Software Engineering Para diagrams Applied:-

In this project object - oriented programming method is adopted using different class and object. Also stored procedure concept is used in the data base level for efficient data handling.

Object Oriented Analysis:-

It is a software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state and its behavior.

Object oriented analysis applies object modeling techniques to analyze the functional requirements for a system. Object oriented design laborites the analysis models to produce implementation specification. OOA focuses on what the system does, OOD on how the system does it.

Object Oriented Design:-

Object oriented design (OOD) transforms the conceptual model produced in object oriented analysis to take account of the constraints imposed by the chosen architecture and any non- functional technological or environmental – constraints, such as transaction throughout, response time, run –time platform, development, environment, or programming language.

The concepts in the analysis model are mapped onto implementation classes and interface. The result model of the solution domain, a detailed description of how the system is to be built.

Object Oriented Programming:-

It is a programming Para diagram that uses “object” data structure consisting of data fields and method together with their interaction to design applications and computer programs. Programming techniques may include features such as data abstraction, modularity, polymorphism, and inheritance.

An object is actually a discrete bundle of functions and procedures, all relating to a particular real world concept such as a bank account holder or hockey player in a computer game. Other piece of software can access the object only by calling its functions and procedures that have been allowed to be called by outsiders. A large no: of software engineers agree that isolating objects in this way makes their software easier to manage and keep track of.

The technology focuses on data rather than process, with programs composed of self- sufficient modules, each instances of which contains all the information needed conventional model, in which program is seen as a list of tasks to perform. In OOP, each object is capable of receiving messages, processing data, and sending messages

to other objects and can be viewed as an independent machine with a distinct role or responsibility.

JAVA Features:

As we know that the Java is an object oriented programming language developed by Sun Microsystems of USA in 1991. Java is first programming language which is not attached with any particular hardware or operating system. Program developed in Java can be executed anywhere and on any system.

Features of Java are as follows:

1. Compiled and Interpreted
2. Platform Independent and portable
3. Object- oriented
4. Robust and secure
5. Distributed
6. Familiar, simple and small
7. Multithreaded and Interactive
8. High performance
9. Dynamic and Extensible

1. Compiled and Interpreted

Basically a computer language is either compiled or interpreted. Java comes together both these approach thus making Java a two-stage system.

Java compiler translates Java code to Byte code instructions and Java Interpreter generate machine code that can be directly executed by machine that is running the Java program.

2. Platform Independent and portable

Java supports the feature portability. Java programs can be easily moved from one computer system to another and anywhere. Changes and upgrades in operating systems, processors and system resources will not force any alteration in Java programs. This is reason why Java has become a trendy language for programming on Internet which

interconnects different kind of systems worldwide. Java certifies portability in two ways.

First way is, Java compiler generates the byte code and that can be executed on any machine. Second way is, size of primitive data types are machine independent.

3. Object- oriented

Java is truly object-oriented language. In Java, almost everything is an Object. All program code and data exist in objects and classes. Java comes with an extensive set of classes; organize in packages that can be used in program by Inheritance. The object model in Java is trouble-free and easy to enlarge.

4. Robust and secure

Java is a most strong language which provides many securities to make certain reliable code. It is design as garbage –collected language, which helps the programmers virtually from all memory management problems. Java also includes the concept of exception handling, which detain serious errors and reduces all kind of threat of crashing the system.

Security is an important feature of Java and this is the strong reason that programmer use this language for programming on Internet.

The absence of pointers in Java ensures that programs cannot get right of entry to memory location without proper approval.

5. Distributed

Java is called as Distributed language for construct applications on networks which can contribute both data and programs. Java applications can open and access remote objects on Internet easily. That means multiple programmers at multiple remote locations to work together on single task.

6. Simple and small

Java is very small and simple language. Java does not use pointer and header files, goto statements, etc. It eliminates operator overloading and multiple inheritance.

7. Multithreaded and Interactive

Multithreaded means managing multiple tasks simultaneously. Java maintains multithreaded programs. That means we need not wait for the application to complete one task before starting next task. This feature is helpful for graphic applications.

8. High performance

Java performance is very extraordinary for an interpreted language, majorly due to the use of intermediate byte code. Java architecture is also designed to reduce overheads during runtime. The incorporation of multithreading improves the execution speed of program.

9. Dynamic and Extensible

Java is also dynamic language. Java is capable of dynamically linking in new class, libraries, methods and objects. Java can also establish the type of class through the query building it possible to either dynamically link or abort the program, depending on the reply.

Java program is support functions written in other language such as C and C++, known as native methods.

About Net Beans:-

Net Beans IDE is a free, open source, popular (with approximately 1 million downloads), integrated development environment used by many developers. Out of the box, it provides built-in support for developing in Java, C, C++, XML, and HTML. And this author especially likes the support for editingJSPs, including syntax highlighting, HTML tag completion, JSP tag completion, and Java codecompletion.

3.5 Preliminary Product Description

Functions

- Development of security control components for front-end systems.
- High-speed data encryption/decryption operations, password authentication, and key management for back-end system.
- For real-world applications, a complex web of software systems is required to ensure security.

Features

- The system is highly user friendly.
- It uses different keys (a key pair) for encryption and decryption. These algorithms are called "AES".
- The system provides security and convenience as keys never need to be transmitted or revealed to anyone.
- The system provides the integrity of data or information.
- The software provides clarity in its functionality even to naïve users.
- Network connection encryption/decryption services: Providing host encryption/decryption operations via network connection through the external security control server.
- Complete key building service: Providing complete key building and management functions, supporting multi-server key sync operations.

Benefits

In order to successfully expand the business, the security level of various application systems is enhanced to meet the requirement from authorities or organizations.

- Enhancing the security level of various application systems to prevent leakage of confidential information.
- Complying with the security specifications of various authorities and international organizations to successfully expand different businesses.

3.6 Conceptual Modules

Data Flow Diagram

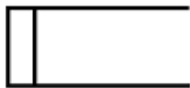
Data flow oriented techniques advocate that the major data items handled by a system must be first identified and then the processing required on these data item to produce the desired output should be determined. The DFD is a simple graphical information that can be used to represent a system in terms of input data to the system, various processing carried out on these data, and the output generated by the system. It was introduced by de macro(1978), Gane and Sarson (1979). The primitive symbols used for constructing DFD;s are:-



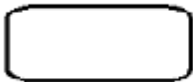
Represent data flow .



Represent process that transforms incoming data.



Represent Data source.



Source/Sink.

CONCEPTUAL MODELS AND DFD'S

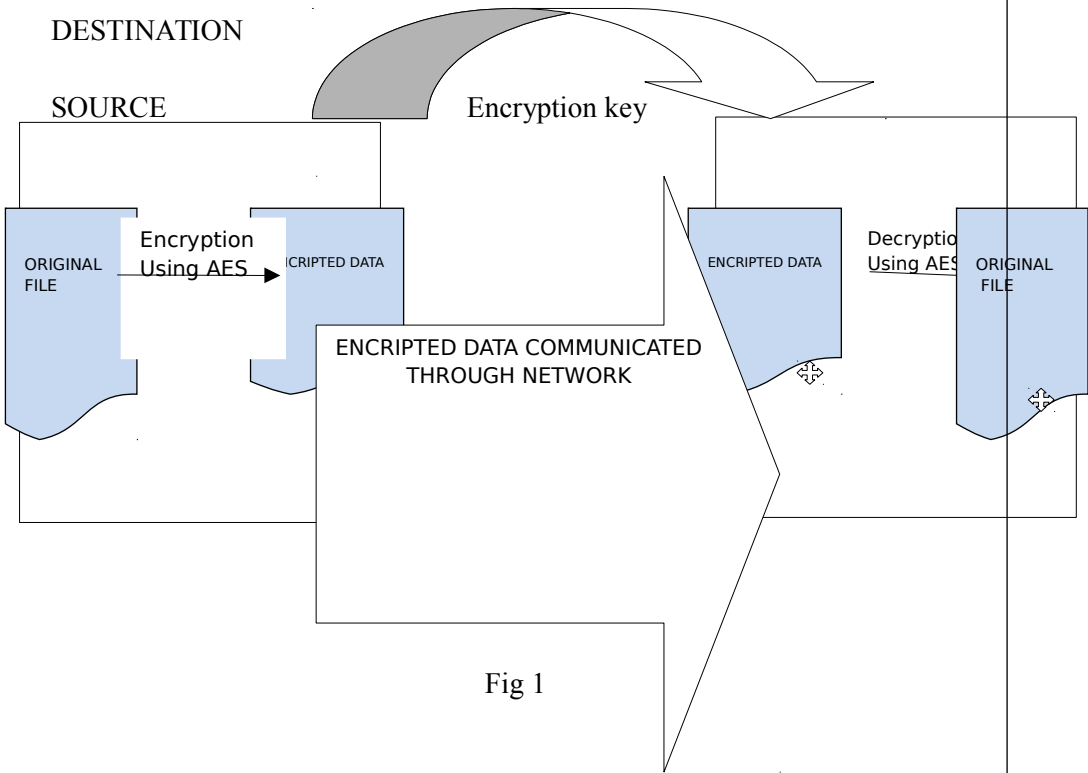
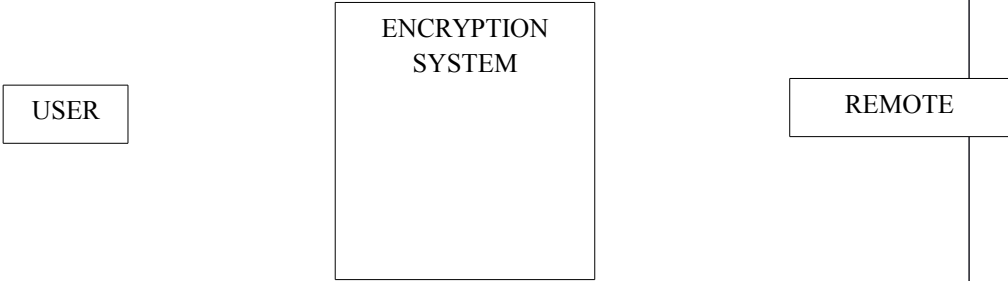


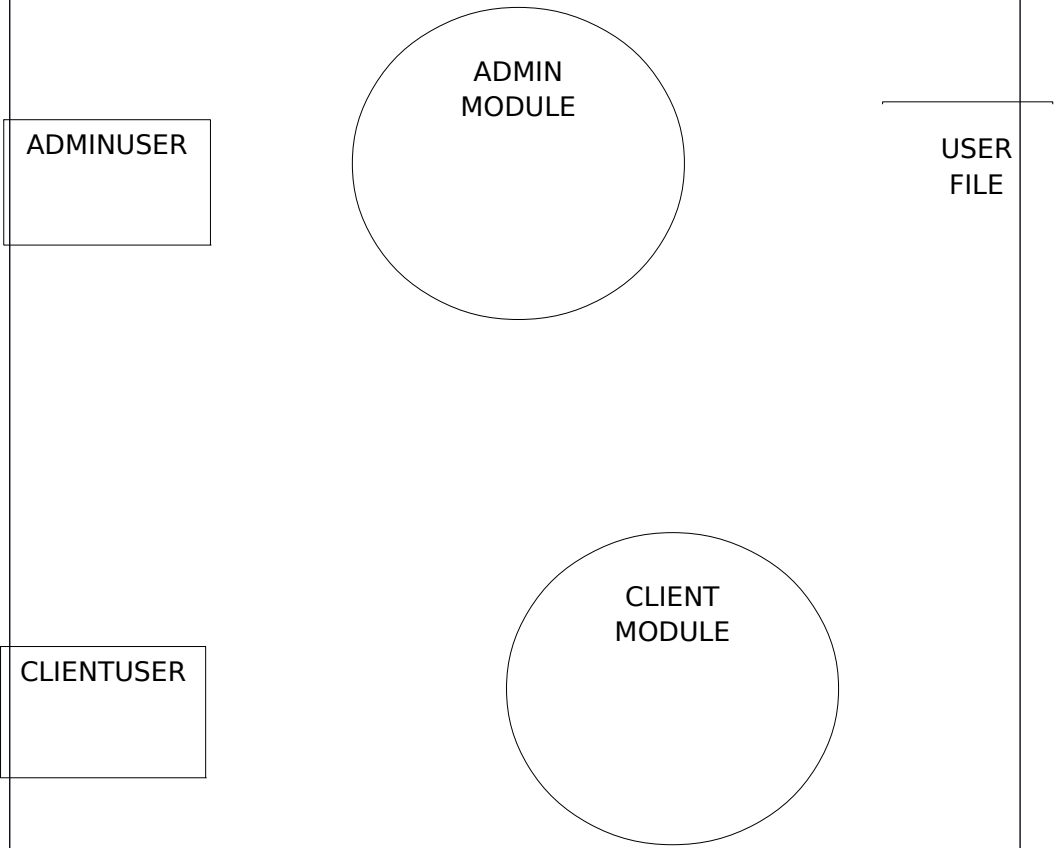
Fig 1

Encryption System

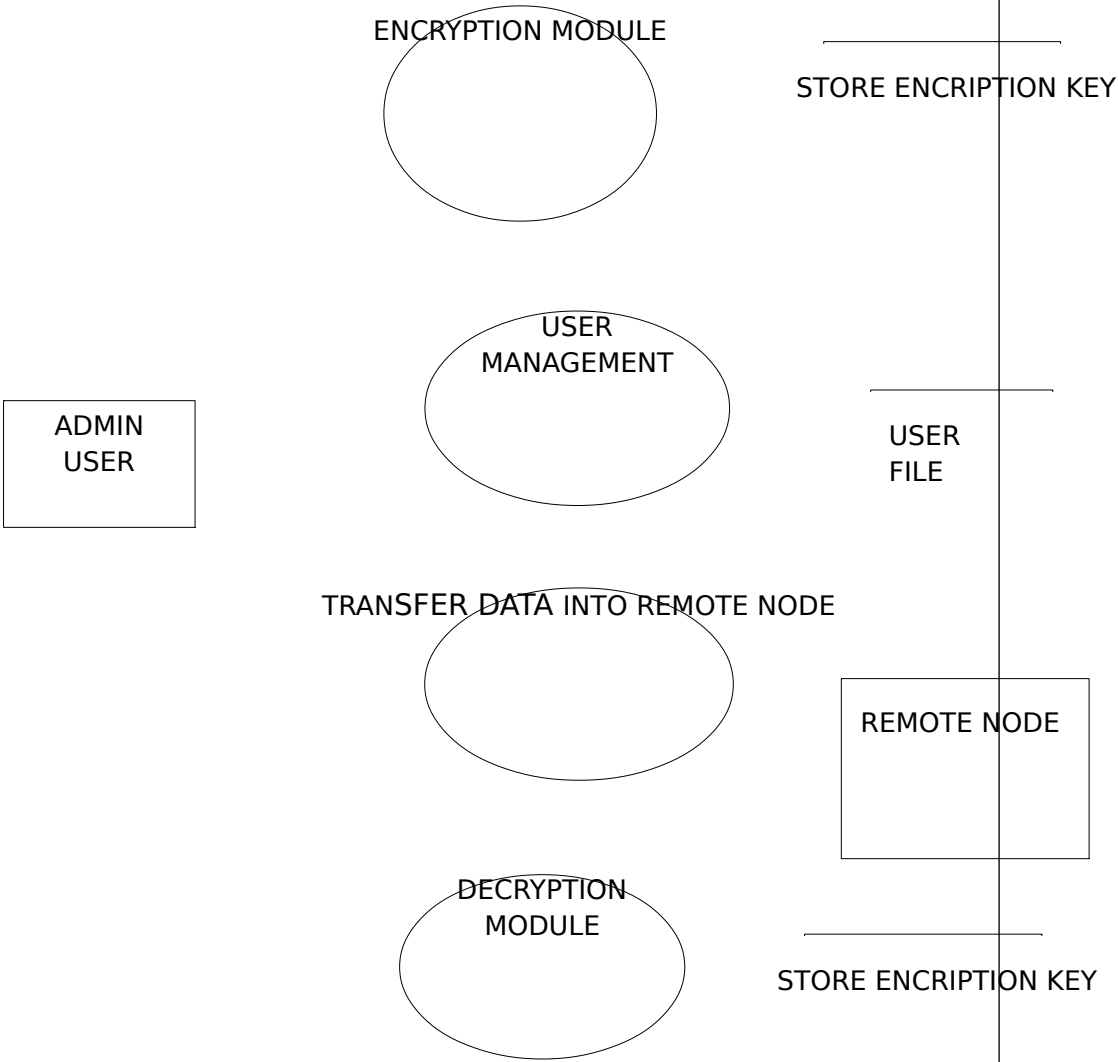
0 – Level DFD



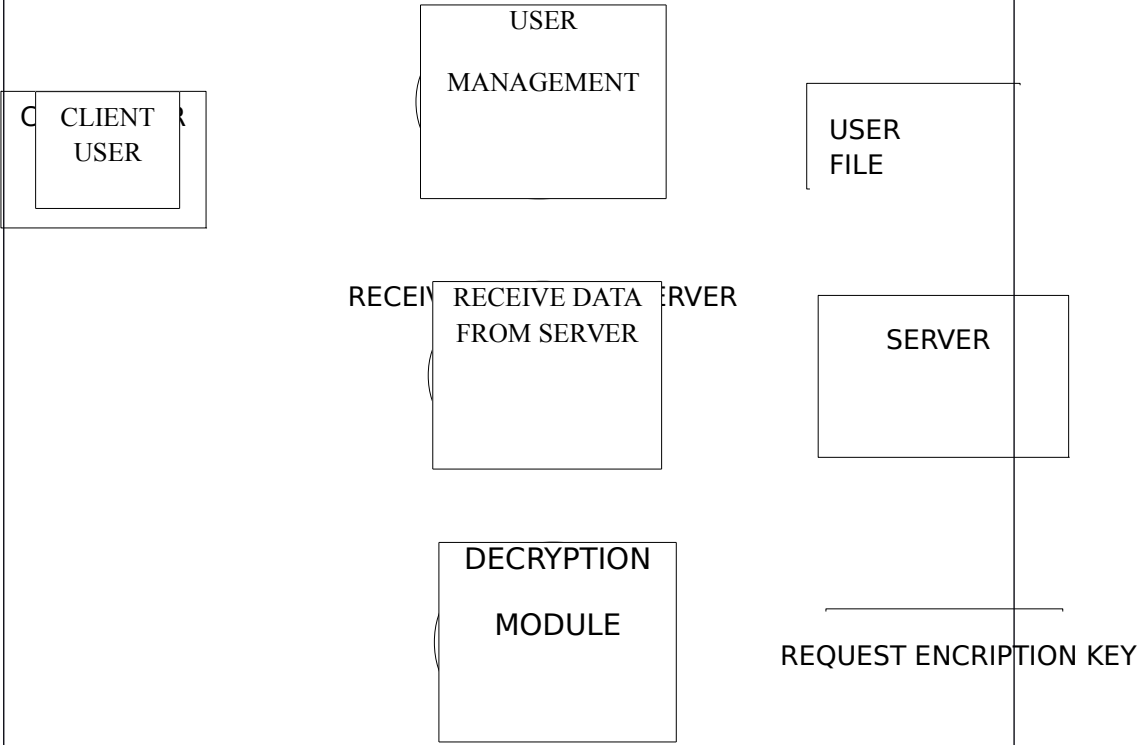
1 – Level DFD



2.1 – Level DFD



2.2 – Level DFD



Use Case Diagram

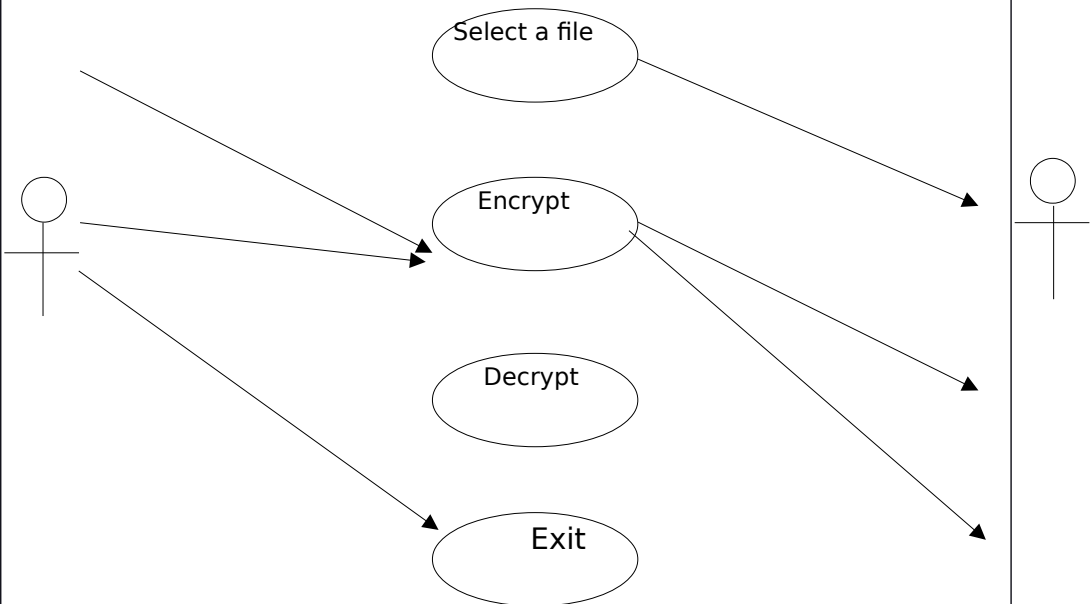


Fig 2

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules

After doing the onsite observation, interview in the authority, and by studying all the materials collected, the new system designed to remove the drawback of the existing system.

The most challenging and creative phase of system life cycle is the system design. In the design phase detailed design of the system selected in study phase is accomplished. System design is a disciplined approach to computer system designed that is easy to understand, reliable, long lasting, and efficient. It should also be cost effective. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study.

Input design

Input design is the process of converting the user –originated input to a computer based format. The design decision for handling input specify how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention.

The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a way so as to get the relevant information, extreme care is taken to obtain the pertinent information. If the data going into the system is incorrect then the processing and outputs will magnify these errors. The goal of designing input data is to make data entry as easy, logical and free from errors as possible.

The following are the objectives of input design:

- To produce a cost effective method of input.
- To make the input forms understandable to the end users.

- To ensure the validation of data inputs.

The nature of input data is determined partially during logical system design. However, the nature of inputs is made more explicit during the physical design. The impact of inputs on the system is also determined.

Effort has been made to ensure that input data remains accurate from the stage at which it is recorded and documented to the stage at which it is accepted by the computer. Validation procedures are also present to detect errors in data input, which is beyond control procedures. Validation procedures are designed to check each record, data item or field against certain criteria.

Output Design

The output design phase of the system design is concerned with the conveyance of information to the end users in a user friendly manner. The output design should be efficient, intelligible so that the systems relationship with the end user is improved and thereby enhancing the process of decision making.

The output design is an ongoing activity almost from the beginning of the project, and follows the principles of form design. Efficient and well-defined output design improves the relation of the system and the user, thus facilitating decision making. The primary considerations in the design of the output are the requirement of the information and the objectives of the end users. The system output may be of any of the following.

- A document.
- A message.

Design is the first in the development phase for any engineered product or system. System design is a process of evaluating alternate

solution, evaluating the choice following up the specification for the chosen alternative. System design work follows logically system analysis. The objective of the system design is to improve the existing system or design a new system as the case may be and implement the system with improved facilities.

Computer software design, like engineering design, approaches in other disciplines, changes continually as new methods, better analysis and broader understanding evolve. Using one of the design methods, the design steps reduces a data design , and architectural design, and procedural design.

4.2 Procedural Design

A design methodology combines a systematic set of rules for creating a program design with diagramming tools needed to represent it. Procedural design is best used to model programs that have an obvious flow of data from input to output. It represents the architecture of a program as asset of interacting processes that pass data from one to another.

4.2.1 Logic Diagrams

Encryption

Key technology: encryption. Store and transmit information in an encoded form that does not make any sense.

The basic mechanism:

- Start with text to be protected. Initial readable text is called clear text.
- Encrypt the clear text so that it does not make any sense at all. The nonsense text is called cipher text. The encryption is controlled by a secret password or number; this is called the encryption key.

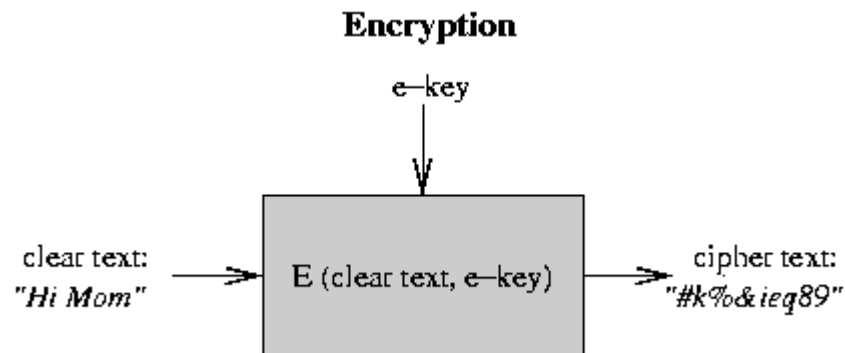


Fig 3

- The encrypted file can be stored in a readable file, or transmitted over unprotected channels.
- To make sense of the cipher text, it must be decrypted back into clear text. This is done with some other algorithm that uses another secret password or number, called the decryption key.

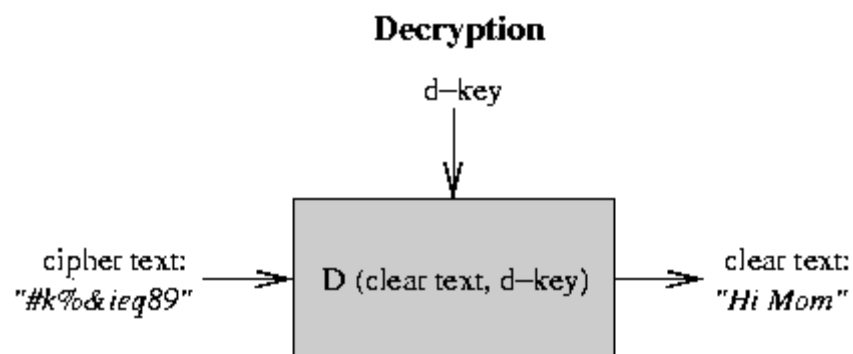


Fig 4

All of this only works under three conditions:

- The encryption function cannot easily be inverted (cannot get back to clear text unless you know the decryption key).

Encryption System

- The encryption and decryption must be done in some safe place so the clear text cannot be stolen.
- The keys must be protected. In most systems, can compute one key from the other (sometimes the encryption and decryption keys are identical), so cannot afford to let either key leak out.

Secret key encryption: new mechanism for encryption where knowing the encryption key does not help you to find decryption key, or vice versa.

- User provides a single password; system uses it to generate keys.
- In these systems, keys are inverses of each other: could just as easily encrypt with decryption key and then use encryption key to recover clear text.
- Each user keeps one key secret, publicizes the other.

4.2.2 Data Structure

Encryption System

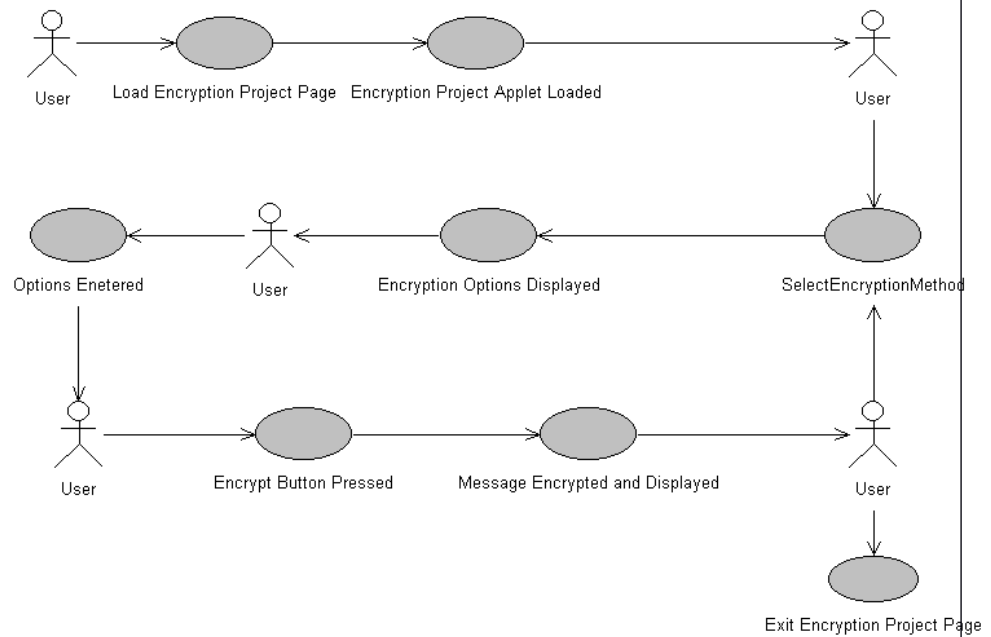


Fig 5

4.2.3 Algorithm Design

- Step1: Enter correct username and password
- Step 2: Choose the encrypt file in list of view
- Step 3: Select any one file in the list.
- Step4: Setup the AES Algorithm Value
- Step5: Save as secret Key for AES Algorithm
- Step6: Re enter secret Key for AES Algorithm
- Step7: Select Encrypt file and Decrypt any one algorithm
- Step8: Select the save file and decrypt it.
- Step9: Successfully Closed the Application

AES encryption algorithm

AES (Advanced Encryption Standard) encryption algorithm is an iterated block cipher with a variable key length. The key length can be specified to 128, 192 or 256 bits. AES (Advanced Encryption Standard) encryption is composed of key expansion algorithm and

encryption (decryption) algorithm, which includes many rounds of iteration and transformation. Procedure may be expressed as follows: key expansion, initialization and iteration rounds. Byte Sub-Transformation, Shift Row transformation, Mix Column transformation and sub-key addition are included in each round except that Mix Column transformation is not included in the last round [6]. In AES (Advanced Encryption Standard) encryption algorithm there are five operation modes. They are CBC (Cipher Block Chaining) mode, ECB (Electronic Code Book) mode, GCM (Galois/Counter Mode) mode, XTS mode and CTR (Counter) mode respectively.

4.3 User Interface Design

The user is often the weak link in the security of a system. Many security breaches have been caused by weak passwords, unencrypted files left on unprotected computers, and successful social engineering attacks. Therefore, it is vitally important that your program's user interface enhance security by making it easy for the user to make secure choices and avoid costly mistakes.

In a social engineering attack, the user is tricked into either divulging secret information or running malicious code. For example, the Melissa virus and the Love Letter worm each infected thousands of computers when users downloaded and opened files sent in email.

This project discusses how doing things that are contrary to user expectations can cause a security risk, and gives hints for creating a user interface that minimizes the risk from social engineering attacks. Secure human interface design is a complex topic affecting operating systems as well as individual programs. This project gives only a few hints and highlights.

4.4 Security Issues

Does security provide some very basic protections that we are naive to believe that we don't need? During this time when the Internet provides essential communication between tens of millions of people

and is being increasingly used as a tool for commerce, security becomes a tremendously important issue to deal with. There are many aspects to security and many applications, Ranging from secure commerce and payments to private Communications and protecting passwords. One essential aspect for secure communications is that of cryptography.

Cryptography is the science of writing in secret code and is an ancient art. The first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription.

In data and telecommunications, cryptography is necessary when communicating over any untreated medium, which includes just about any network, particularly the Internet.

Within the context of any application-to-application communication, there are some specific security requirements, including:

- **Authentication**: The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.)
- **Privacy/confidentiality**: Ensuring that no one can read the message except the intended receiver.
- **Integrity**: Assuring the receiver that the received message has not been altered in any way from the original.
- **Non-repudiation**: A mechanism to prove that the sender really sent this message. Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication.

4.5 Test Case Design

Test case design techniques I: White box Testing

White box test of software is predicted on a close examination of procedural detail. The status of the project may be tested at various points to determine whether the expected or asserted status is corresponding to the actual status. Using this, the following test cases can be derived.

- Exercise all logical conditions on their true or false side.
- Execute all loops within their boundaries and their operation bounds.
- Exercise internal data structure to ensure their validity.

Black Box Testing: -

Knowing the specified function that a product has been designed to perform, test can be conducted that each function is fully operational. Black box test is carried out to test that input to a function is properly accepted and output is correctly produced. A black box tests examines some aspects of a system with little regard for the internal structure of the software. Errors in the following categories were found through black box testing,

- Incorrect or missing function
- Interface errors
- Errors in database structure or external database access.
- Performance errors
- Initialization and termination errors.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

System implementation is the final phase i.e., putting the utility in to action implementation is the stage in the project where theoretical design turned into working system. The most crucial stage is the achieving a new successful system and giving confidence in new system that it will work efficiently effectively. The system is implemented only after through checking is done and if it is found working in according to the specification.

It involves careful planning, investigation of the current system and constraints on implementation, design of method to achieve. Two checking is done and if it is found working according to the specification, major task ok preparing the implementation are educating, training the users.

The implementation process beings with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resource and the important in implementation stage is, gaining the user confidence that the new system will work and be effective.

The system can be implemented only after through testing is done. This method is also offers the greatest security since the existing system can take over if the errors are found or inability to handle certain type of transaction. While using the new system.

5.2 Coding details and Code Efficiency

The goal of coding phase is to translate the design of the system in to code in a given programming language. The coding phase affects both testing and maintenance profoundly. Well written code can reduce the testing and maintenance effort because the testing and maintenance cost of the software are much higher than the coding cost. Simplicity and clarity should be preserved during the coding phase.

5.2.1 Code Efficiency

Comments and descriptions are given to identify the functions and to describe what the function is doing. Thus it may be useful while finding errors and thus makes the identifying of the function easy.

Login.java

```
packageadvancedencrptionsystem;

importjava.awt.Color;

importjava.io.BufferedReader;

importjava.io.FileReader;

importjava.io.IOException;

public class Login extends javax.swing.JFrame {

    public Login() {

        initComponents();

    }

    @SuppressWarnings("unchecked")

    private void

    jButton1ActionPerformed(java.awt.event.ActionEventevt) {
```

```
// TODO add your handling code here:

String s = null,p=null;

String user=jTextField1.getText();

String pass=jPasswordField1.getText();

booleannullCheck=nullCheck(user, pass);

if (nullCheck) {

if(user.equals("admin")&&pass.equals("admin"))

    {

newAdminHome().setVisible(true);

    }

else

    {

booleanisFound = false;

        String record = null;

        FileReader in = null;

        try{

in = new FileReader ("C:\\aes\\login.txt");

        BufferedReader bin = new BufferedReader(in);

        record = new String();

        while ((record = bin.readLine()) != null)

            {

if (jTextField1.getText().contentEquals(record))

                {

record=bin.readLine();

if(jPasswordField1.getText().contentEquals(record))

newUser_Home().setVisible(true);


```

```
        }

    else

        {

jLabel4.setBackground(Color.red);
jLabel4.setText("Please Enter Valid UserName And Password");

        }

    }

    bin.close();
    bin = null;
} catch(IOException ioe){

        }

    }

}

else

    {

jLabel4.setBackground(Color.red);
jLabel4.setText("Please Enter UserName And Password" );

    }

}

private void
jButton2ActionPerformed(java.awt.event.ActionEventevt) {

    this.dispose();
```



```
    }

    public static void main(String args[]) {

        //<editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">

        try {

            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                }

            }

        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Login.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Login.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Login.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException
ex) {

            java.util.logging.Logger.getLogger(Login.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);

        }

        //</editor-fold>
    }
}
```

```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new Login().setVisible(true);  
    }  
});  
}  
  
public boolean nullCheck(String userName, String password) {  
    if (userName.equals("") || password.equals("")) {  
        return false;  
    } else {  
        return true;  
    }  
}  
  
    // Variables declaration - do not modify  
    private javax.swing.JButton jButton1;  
    private javax.swing.JButton jButton2;  
    private javax.swing.JLabel jLabel1;  
    private javax.swing.JLabel jLabel2;  
    private javax.swing.JLabel jLabel3;  
    private javax.swing.JLabel jLabel4;  
    private javax.swing.JPanel jPanel1;  
    private javax.swing.JPasswordField jPasswordField1;  
    private javax.swing.JTextField jTextField1;  
  
    // End of variables declaration  
}
```

AdminHome.java

```
packageadvancedencrptionsystem;

importjava.io.BufferedReader;
importjava.io.BufferedWriter;
importjava.io.File;
importjava.io.FileNotFoundException;
importjava.io.FileReader;
importjava.io.IOException;
importjava.io.InputStream;
importjava.io.InputStreamReader;
importjava.io.ObjectOutputStream;
importjava.io.OutputStream;
importjava.io.OutputStreamWriter;
importjava.io.PrintWriter;
importjava.net.InetAddress;
importjava.net.ServerSocket;
importjava.net.Socket;
importjava.net.UnknownHostException;
importjava.util.logging.Level;
importjava.util.logging.Logger;
importjavax.swing.DefaultListModel;

public class AdminHome extends javax.swing.JFrame {

    private static Socket socket;

    String number=null;

    String path = "C:\\aes\\upload";

    String path2 = "C:\\aes\\upload";

    publicAdminHome() {
```

```
initComponents();

try {

    String files;

    File folder = new File(path);

    File[] listOfFiles = folder.listFiles();

    DefaultListModel<String> resultList = new DefaultListModel();

    for (int i = 0; i < listOfFiles.length; i++)

        {

            if (listOfFiles[i].isFile())

                {

                    files = listOfFiles[i].getName();

                    resultList.addElement(files);

                    jList1.setModel(resultList);

                }

        }

    } catch (Exception ex) {

        Logger.getLogger(AdminHome.class.getName()).log(Level.SEVERE, null, ex);

    }

}

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    newUser_Registration().setVisible(true);

}
```

```
    }

    private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    newEncriptionform().setVisible(true);

}

    private void
jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    int port = 1234;

    String hostname = "localhost";

    String input,output;

    try {

        Socket skt = new Socket(hostname, port);

        BufferedReader in = new BufferedReader(new

        InputStreamReader(skt.getInputStream()));

        System.out.println("Server:" + in.readLine());

        PrintWriter out = new PrintWriter(skt.getOutputStream(), true);

        out.println("lalala");

        out.close();

        in.close();

        skt.close();

    }

    catch(Exception e) {

        System.out.print("Error Connecting to Server\n");

    }

}
```

Encryption System

```
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{

    System.out.println("haiiiiiiiii");

}

private void
jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    // TODO add your handling code here:
    String files2;
    File folder2 = new File(path2);
    File[] listOfFiles2 = folder2.listFiles();
    DefaultListModel resultList2 = new DefaultListModel();
    for (int i = 0; i < listOfFiles2.length; i++)
    {

        if (listOfFiles2[i].isFile())
        {

            files2 = listOfFiles2[i].getName();
            resultList2.addElement(files2);

        }
    }

    ServerSocket myServerSocket = new ServerSocket(9999);
    Socket skt = myServerSocket.accept();

    ObjectOutputStream objectOutput = new
    ObjectOutputStream(skt.getOutputStream());
    objectOutput.writeObject(resultList2);
    objectOutput.flush();
    skt.close();
    myServerSocket.close();

    } catch (IOException ex) {
        Logger.getLogger(AdminHome.class.getName()).log(Level.
        SEVERE, null, ex);
    }
}
try {

    int port = 25001;
    ServerSocket serverSocket = new ServerSocket(port);
```

Encryption System

```
System.out.println("Server Started and listening to the port  
25000");
```

```
        //Server is running always. This is done using this  
while(true) loop  
InputStreamReaderisr;  
BufferedReaderbr;
```

```
while(true)  
{  
        //Reading the message from the client  
socket = serverSocket.accept();  
InputStream is = socket.getInputStream();  
isr = new InputStreamReader(is);  
br = new BufferedReader(isr);  
number = br.readLine();  
System.out.println("Message received from client is  
"+number);
```

```
is.close();  
serverSocket.close();  
socket.close();
```

```
jTextField2.setText(number);  
}  
        } catch (IOException ex) {  
Logger.getLogger(AdminHome.class.getName()).log(Level.  
SEVERE, null, ex);  
}  
  
}
```

```
private void  
jButton4ActionPerformed(java.awt.event.ActionEventevt) {  
        // TODO add your handling code here:  
}
```

```
private void  
jButton3ActionPerformed(java.awt.event.ActionEventevt) {  
}
```

```
Private void  
jButton9ActionPerformed(java.awt.event.ActionEventevt) {
```

```
this.dispose();
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new AdminHome().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JButton jButton8;
    private javax.swing.JButton jButton9;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JList jList1;
    private javax.swing.JList jList2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}
```

Encryptionform.java

```
package advancedencryptionsystem;

import static
advancedencryptionsystem.CipherExample.encrypt;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
```


Encryption System

```
import java.io. FileWriter;
import java.io. IOException;
import java.io. InputStream;
import java.io. OutputStream;
import java.util.logging. Level;
import java.util.logging. Logger;
import javax.crypto. Cipher;
import javax.crypto. CipherInputStream;
import javax.crypto. CipherOutputStream;
import javax.crypto. SecretKey;
import javax.crypto. SecretKeyFactory;
import javax.crypto.spec. DESKeySpec;
import javax.swing. JFileChooser;
public class Encryptionform extends javax.swing.JFrame {
    String fname=null;
    public Encryptionform() {
        initComponents();
    }
    private void
    jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        final JFileChooser fc = new JFileChooser();
        fc.showOpenDialog(this);
        String f = fc.getSelectedFile().getAbsolutePath();
        fname=fc.getSelectedFile().getName();
        jTextField1.setText(f);
    }
    private void
    jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        String key = "squirrel123"+fname; // needs to be at least
        8 characters for DES
        byte[] Key = null;
        try {
            Key = encrypt(key);
            //new AdminHome().setVisible(true);
            this.dispose();
        } catch (Exception ex) {
            Logger.getLogger(Encryptionform.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```
    }
    String strkey=null;
    strkey=Key.toString();
    System.out.println("....."+strkey);
    FileInputStreamfis;
    try {
        String s=jTextField1.getText();
        String result = s.replace("\\", "/");
        fis = new FileInputStream(result);
        FileOutputStreamfos = new
        FileOutputStream("C:\\aes\\upload\\"+fname+""");
        encrypt(strkey, fis, fos);
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Encriptionform.class.getName()).log(Level.SEVERE, null, ex);
    } catch (Throwable ex) {
        Logger.getLogger(Encriptionform.class.getName()).log(Level.SEVERE, null, ex);
    }
}

FileOutputStream fop = null;
File file;
try {
    FileWriterfw = new FileWriter("C:\\aes\\filekey.txt", true);
    BufferedWriterbw = new BufferedWriter(fw);
    bw.newLine();
    String NL = System.getProperty("line.separator");
    bw.write(fname+NL);
    bw.write(strkey+NL);
    bw.write("$$$"+NL);
    bw.close();

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fop != null) {
            try {
                fop.close();
            }
            catch (IOException ex) {
                Logger.getLogger(User_Registration.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}
```

Encryption System

```
    }

    private void
    jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Encryptionform().setVisible(true);
            }
        });
    }
    public static byte[] encrypt(String x) throws Exception {

        java.security.MessageDigest digest = null;

        digest =
        java.security.MessageDigest.getInstance("SHA-1");

        digest.reset();

        digest.update(x.getBytes("UTF-8"));

        return digest.digest();
    }

    public static void encrypt(String key, InputStream is,
        OutputStream os) throws Throwable {
        encryptOrDecrypt(key,
        Cipher.ENCRYPT_MODE, is, os);
    }

    public static void decrypt(String key, InputStream is,
        OutputStream os) throws Throwable {
        encryptOrDecrypt(key,
        Cipher.DECRYPT_MODE, is, os);
    }

    public static void encryptOrDecrypt(String key, int
        mode, InputStream is, OutputStream os) throws Throwable {
```

Encryption System

```
        DESKeySpec dks = new
DESKeySpec(key.getBytes());
        SecretKeyFactory skf =
SecretKeyFactory.getInstance("DES");
        SecretKey desKey = skf.generateSecret(dks);
        Cipher cipher = Cipher.getInstance("DES"); //
DES/ECB/PKCS5Padding for SunJCE

        if (mode == Cipher.ENCRYPT_MODE) {
            cipher.init(Cipher.ENCRYPT_MODE,
desKey);
            CipherInputStream cis = new
CipherInputStream(is, cipher);
            doCopy(cis, os);
        } else if (mode == Cipher.DECRYPT_MODE) {
            cipher.init(Cipher.DECRYPT_MODE,
desKey);
            CipherOutputStream cos = new
CipherOutputStream(os, cipher);
            doCopy(is, cos);
        }
    }

    public static void doCopy(InputStream is,
OutputStream os) throws IOException {
        byte[] bytes = new byte[64];
        int numBytes;
        while ((numBytes = is.read(bytes)) != -1) {
            os.write(bytes, 0, numBytes);
        }
        os.flush();
        os.close();
        is.close();
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

User_Home.java

```
packageadvancedencrptionsystem;
import                                                                    static
advancedencrptionsystem.CipherExample.decrypt;
import                                                                    static
advancedencrptionsystem.Encrptionform.doCopy;
importjava.io.BufferedReader;
importjava.io.BufferedWriter;
importjava.io.File;
importjava.io.FileInputStream;
importjava.io.FileNotFoundException;
importjava.io.FileOutputStream;
importjava.io.FileReader;
importjava.io.IOException;
importjava.io.InputStream;
importjava.io.InputStreamReader;
importjava.io.ObjectInputStream;
importjava.io.OutputStream;
importjava.io.OutputStreamWriter;
importjava.io.PrintWriter;
importjava.net.InetAddress;
importjava.net.ServerSocket;
importjava.net.Socket;
importjava.net.UnknownHostException;
importjava.util.logging.Level;
importjava.util.logging.Logger;
importjavax.crypto.Cipher;
importjavax.crypto.CipherInputStream;
importjavax.crypto.CipherOutputStream;
importjavax.crypto.SecretKey;
importjavax.crypto.SecretKeyFactory;
importjavax.crypto.spec.DESKeySpec;
importjavax.swing.DefaultListModel;
public class User_Home extends javax.swing.JFrame {
private static Socket socket3;
private static Socket socket4;
publicUser_Home() {
initComponents();
try {
Socket socket = new Socket("192.168.1.26",9999);
DefaultListModel resultList2 = new DefaultListModel();
try {
```

Encryption System

```
ObjectInputStream objectInput = new
ObjectInputStream(socket.getInputStream()); //Error Line!
try {
    Object object = objectInput.readObject();
    resultList2 = (DefaultListModel) object;

    socket.close();

    jList1.setModel(resultList2);
    } catch (ClassNotFoundException e) {
        System.out.println("The title list has not come from the
server");
        e.printStackTrace();
    }
    } catch (IOException e) {
        System.out.println("The socket for reading the object has
problem");
        e.printStackTrace();
    }
    } catch (UnknownHostException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String msg=null;
        Object obj=jList1.getSelectedValue();
        msg=String.valueOf(obj);

        String host = "192.168.1.26";
        int port = 25001;
        InetAddress address = InetAddress.getByName(host);
        socket3 = new Socket(address, port);

        //Send the message to the server
        OutputStream os = socket3.getOutputStream();
        OutputStreamWriter osw = new OutputStreamWriter(os);
        BufferedWriter bw = new BufferedWriter(osw);

        String number = "2";
```

Encryption System

```
String sendMessage = msg + "\n";
bw.write(sendMessage);

System.out.println("Message sent to the server : "
"+sendMessage);
bw.flush();

    } catch (UnknownHostException ex) {
    Logger.getLogger(User_Home.class.getName()).log(Level.S
EVERE, null, ex);
    } catch (IOException ex) {
    Logger.getLogger(User_Home.class.getName()).log(Level.S
EVERE, null, ex);
    }
    InputStreamReader isr;

    BufferedReader br1;

    try
    {
    System.out.println("aaa");
        String host = "192.168.1.26";
    int port = 25005;
    System.out.println("bbb");
    InetAddress address = InetAddress.getByName(host);
        socket4 = new Socket(address, port);
    //      Send the message to the server
    System.out.println("aaaaa");
        //Get the return message from the server
    InputStream is = socket4.getInputStream();
    System.out.println("ccc");
    isr = new InputStreamReader(is);
    System.out.println("ccc");
        br1 = new BufferedReader(isr);
    System.out.println("ccs");

    System.out.println(""+br1.readLine());
        String message = br1.readLine().toString();
    System.out.println("ccc");
    System.out.println("Message received from the server : "
+message);

    //      jTextField1.setText(message);
    }
    catch (Exception exception)
```

```
        {
        exception.printStackTrace();
        }
    finally
    {
        //Closing the socket
    try
        {
        socket4.close();
        }
    catch(Exception e)
        {
        e.printStackTrace();
        }
    }
}
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    FileInputStream fis2 = null;
    try {
        // TODO add your handling code here:
        String key=jTextField1.getText();
        String filenm=null;
        Object obj=jList1.getSelectedValue();
        filenm=String.valueOf(obj);
        System.out.println(""+filenm);

        fis2 = new
        FileInputStream("C:\\aes\\upload\\"+filenm+"");
        System.out.println("cvxcvxc");
        FileOutputStream fos2 = new
        FileOutputStream("C:\\aes\\download\\"+filenm+"");

        decrypt(key, fis2, fos2);

        } catch (FileNotFoundException ex) {
        Logger.getLogger(User_Home.class.getName()).log(Level.S
        EVERE, null, ex);
        } catch (Throwable ex) {
        Logger.getLogger(User_Home.class.getName()).log(Level.S
        EVERE, null, ex);
        } finally {
        try {
        fis2.close();
```



```
        } catch (IOException ex) {
        Logger.getLogger(User_Home.class.getName()).log(Level.S
EVERE, null, ex);
        }
    }

    private void
    jButton3ActionPerformed(java.awt.event.ActionEventevt) {
    InputStreamReaderisr;

    BufferedReader br1;
    try
    {
    System.out.println("aaa");
        String host = "192.168.1.26";
    int port = 25005;
    System.out.println("bbb");
    InetAddress address = InetAddress.getByName(host);
        socket4 = new Socket(address, port);

    //      Send the message to the server
    System.out.println("aaaaa");

        //Get the return message from the server
    InputStream is = socket4.getInputStream();
    System.out.println("ccc");
    isr = new InputStreamReader(is);
    System.out.println("ccc");
        br1 = new BufferedReader(isr);
    System.out.println("ccs");
    System.out.println(""+br1.readLine());
        String message = br1.readLine().toString();
    System.out.println("ccc");
    System.out.println("Message received from the server : "
+message);

    //      jTextField1.setText(message);
    }
    catch (Exception exception)
    {
    exception.printStackTrace();
    }
    finally
    {
```

```
//Closing the socket
try
{
socket4.close();
}
catch(Exception e)
{
e.printStackTrace();
}
}

private void
jButton4ActionPerformed(java.awt.event.ActionEventevt) {
this.dispose();    // TODO add your handling code here:
}

public static void main(String args[]) {

setting code (optional) ">

try {
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName
());
break;
}
}
} catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(User_Home.class.getNa
me()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(User_Home.class.getNa
me()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(User_Home.class.getNa
me()).log(java.util.logging.Level.SEVERE, null, ex);
} catch
(javax.swing.UnsupportedLookAndFeelException ex) {
```

Encryption System

```
java.util.logging.Logger.getLogger(User_Home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        newUser_Home().setVisible(true);
    }
});
}
```

```
public void listFilesForFolder(final File folder) {
    for (final File fileEntry : folder.listFiles()) {
        if (fileEntry.isDirectory()) {
            listFilesForFolder(fileEntry);
        } else {
            System.out.println(fileEntry.getName());
        }
    }
}
```

```
public static void encrypt(String key, InputStream is,
    OutputStream os) throws Throwable {
    encryptOrDecrypt(key,
        Cipher.ENCRYPT_MODE, is, os);
}
```

```
public static void decrypt(String key, InputStream is,
    OutputStream os) throws Throwable {
    encryptOrDecrypt(key,
        Cipher.DECRYPT_MODE, is, os);
}
```

```
public static void encryptOrDecrypt(String key, int
    mode, InputStream is, OutputStream os) throws Throwable {

        DESKeySpec dks = new
        DESKeySpec(key.getBytes());
        SecretKeyFactory skf =
        SecretKeyFactory.getInstance("DES");
        SecretKey desKey = skf.generateSecret(dks);
```

Encryption System

```
Cipher cipher = Cipher.getInstance("DES"); //
DES/ECB/PKCS5Padding for SunJCE

if (mode == Cipher.ENCRYPT_MODE) {
    cipher.init(Cipher.ENCRYPT_MODE,
desKey);
        CipherInputStreamcis      =      new
CipherInputStream(is, cipher);
        doCopy(cis, os);
    } else if (mode == Cipher.DECRYPT_MODE) {
        cipher.init(Cipher.DECRYPT_MODE,
desKey);
        CipherOutputStreamcos      =      new
CipherOutputStream(os, cipher);
        doCopy(is, cos);
    }
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JList jList1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}
```

User_Registration.Java

```
package advancedencrptionsystem;

import java.awt.Color;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;
```

```
import java.util.logging.Level;
import java.util.logging.Logger;

public class User_Registration extends javax.swing.JFrame {

    public User_Registration() {
        initComponents();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    {
        String user=jTextField2.getText();
        String pass=jPasswordField1.getText();
        String conf=jPasswordField2.getText();

        if(pass.equals(conf))
        {
            FileOutputStream fop = null;

            File file;

            String content = user+"\n"+pass+"\n"+"$$$";

            try {
                FileWriter fw = new FileWriter("C:\\aes\\login.txt", true);
                BufferedWriter bw = new BufferedWriter(fw);
                bw.newLine();

                String NL = System.getProperty("line.separator");

                bw.write(user+NL);
                bw.write(pass+NL);
                bw.write("$$$"+NL);
                bw.close();
            }
        }
    }
}
```

```
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
if (fop != null) {
    try {
        fop.close();
    } catch (IOException ex) {
        Logger.getLogger(User_Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

jTextField1.setText("");
jTextField2.setText("");
jPasswordField1.setText("");
jPasswordField2.setText("");
jLabel5.setText("");
    }
else
    {
        jLabel5.setBackground(Color.red);
        jLabel5.setText("password incorrect");
    }

}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    this.dispose();    // TODO add your handling code here:
}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            newUser_Registration().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JPasswordField jPasswordField2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
```

```
// End of variables declaration  
}
```

5.3 Testing Approaches

Software testing is a critical element of software quality assurance and represents the ultimate review of specifications, design & code generation. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences.

Testing objectives.

- Testing is the process of executing a program with the intent of finding errors.
- Preparing a test case that has high probability of finding yet undiscovered error.
- Testing to erase out all kind of bugs from the programmed.
- Verification Testing.
- Requirement Verification.

For the successful completion of the project requirement verification plays a major role. Once the required analysis is done, the required analysis document created should have to be verified with the client to ensure for perfectness of the analysis done. The next stages of the project should be created out only after getting the approval from the client.

Software testing is a critical element of Quality Assurance and represents the ultimate provision of specification, design and coding. Testing represents an interesting anomaly phase it was attempt to build software from an abstract concept to a tangible implementation.

The testing phase involves the testing of the development system using various test data. After preparing the test data the system under study is tested using these test data. While testing the system using the test data, errors were found and corrected. Thus a series of test were performed system before the system was ready for implementation. The various types of testing done on the system are:

- Unit testing.
- Integration testing.
- Validation testing
- Output testing.
- User acceptance testing.
- Black box testing.
- White box testing.

5.3.1 Unit Testing

Unit testing focus verification effort on the smallest limit of software design. Using the unit test plan prepared in the design phase of the system, important control paths are tested to uncover the errors within the module. This testing was carried out doing the coding itself. In this

testing step, each module is going to be working satisfactory as the expected output from the module.

5.3.2 Integration Testing: -

It is the systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with the interface. The objective is to take unit-tested module and build the program structure that has been dictated by design. All modules are combined in this testing step. Then the entire program is tested as a whole. If a set of errors is encountered correction is difficult because the isolation of causes is complicated by vastness of the entire program.

Using integrated test plans prepared in the design phase of the system developed as a guide, the integration was carried out. All the errors found in the system were corrected for the next testing steps.

5.4 Modifications and Improvements

Debugging is the process of isolating and correcting the causes of known errors. Success at debugging requires highly developed problem solving skills. Commonly used debugging methods include induction, deduction and backtracking. Debugging by induction involves the following steps: -

- ❖ Collect the following information.
- ❖ Tool for patterns.
- ❖ From one or more hypothesis.
- ❖ Prove or dispose each hypothesis.
- ❖ Implement appropriate correctness.
- ❖ Verify the correctness.
- ❖ Debugging by deduction invokes.
- ❖ List possible cause for the observed facilities.
- ❖ Elaborate the remaining hypothesis.

CHAPTER 6: RESULTS AND DISCUSSIONS

6.3 Test Report

A report is very formal document that is written for a variety of purpose, in any organization. This used to represent the terms in a particular manner the software “Advanced Encryption System” some contains report such as:

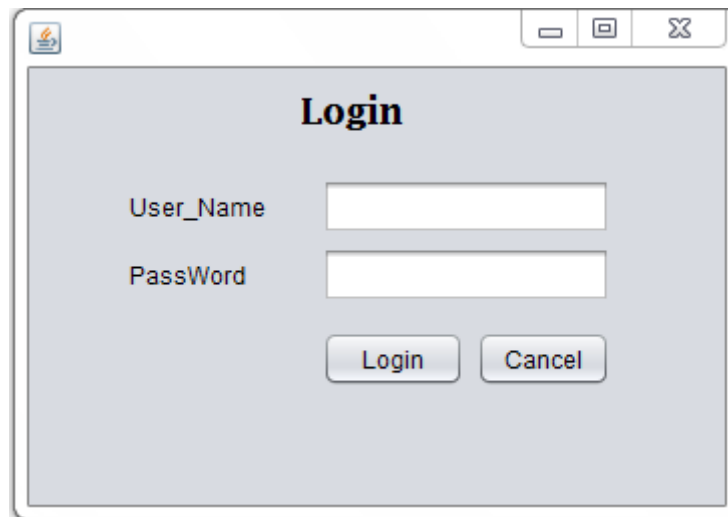
- Encryption files
- Decryption files
- Download files
- Upload Files
- Key Files

6.4 User Documentation

Encryption System

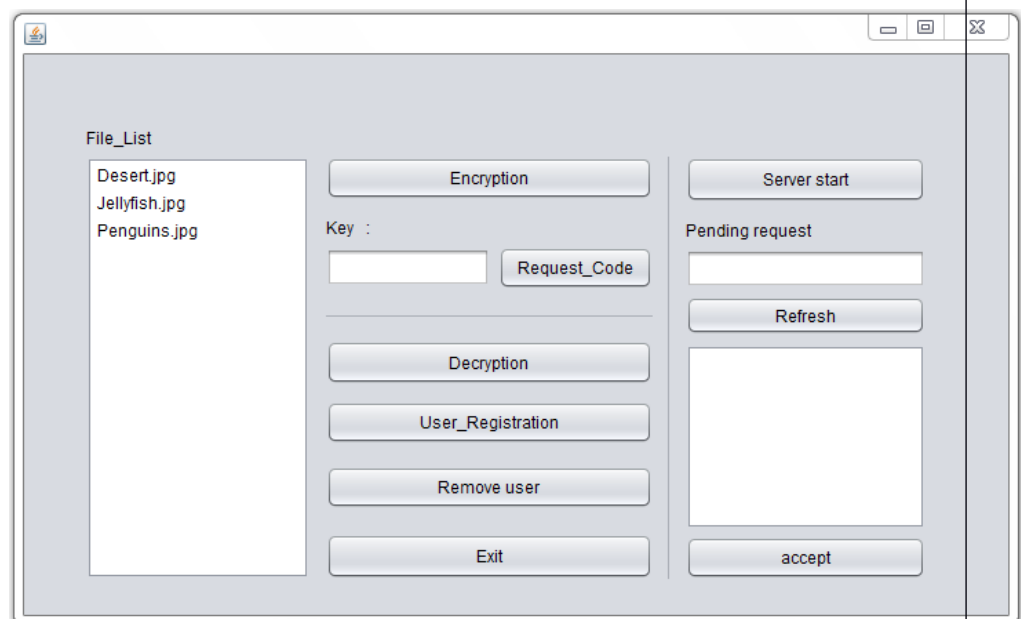
The users shall be provided with tutorials that provide details on how the software may be successfully used.

Login Form



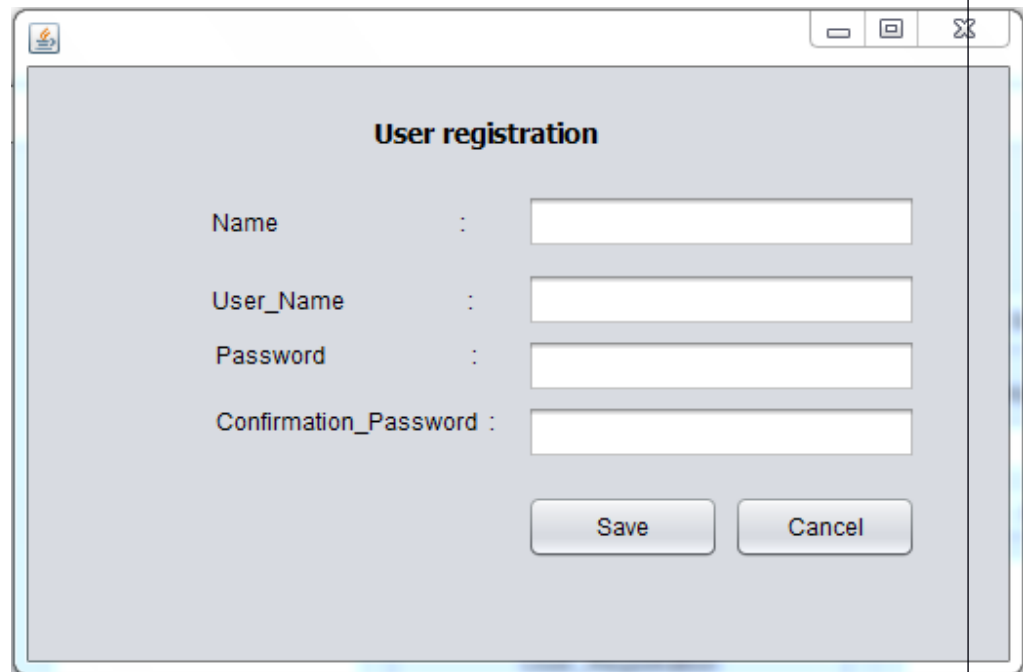
A screenshot of a 'Login' window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area has a light gray background. At the top center, the word 'Login' is displayed in a bold, black, serif font. Below this, there are two labels: 'User_Name' and 'PassWord', each followed by a white rectangular text input field. At the bottom center, there are two buttons: 'Login' and 'Cancel', both with a light gray gradient and a thin black border.

Admin MainForm



A screenshot of an 'Admin MainForm' window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area has a light gray background. On the left side, there is a label 'File_List' above a white rectangular list box containing three items: 'Desert.jpg', 'Jellyfish.jpg', and 'Penguins.jpg'. To the right of the list box, there is a vertical stack of buttons: 'Encryption', 'Decryption', 'User_Registration', 'Remove user', and 'Exit'. Above the 'Encryption' button is a label 'Key :' followed by a white rectangular text input field and a 'Request_Code' button. To the right of this stack, there is a 'Server start' button at the top, followed by a 'Pending request' label above a white rectangular text input field, then a 'Refresh' button, and finally an 'accept' button at the bottom.

User Registration Form



A screenshot of a 'User registration' dialog box. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area has a light gray background and is titled 'User registration' in bold. It contains four labels with corresponding text input fields: 'Name', 'User_Name', 'Password', and 'Confirmation_Password'. Each label is followed by a colon and a white rectangular input field. At the bottom right, there are two buttons: 'Save' and 'Cancel'.

User registration

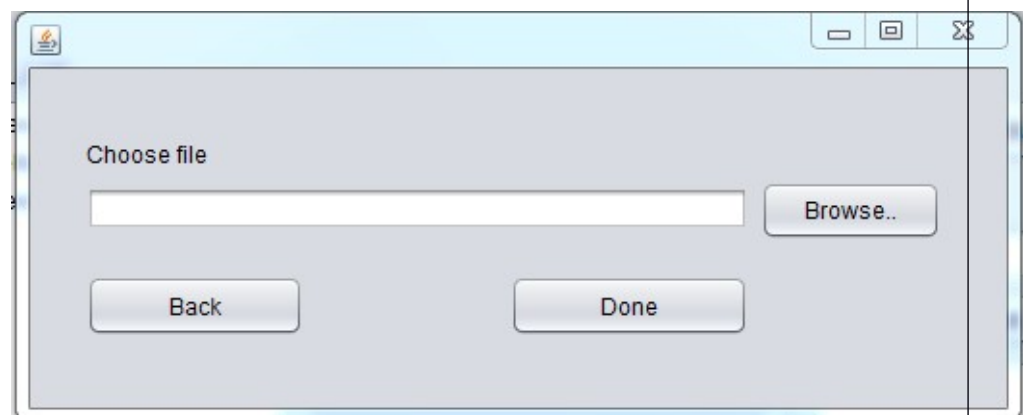
Name :

User_Name :

Password :

Confirmation_Password :

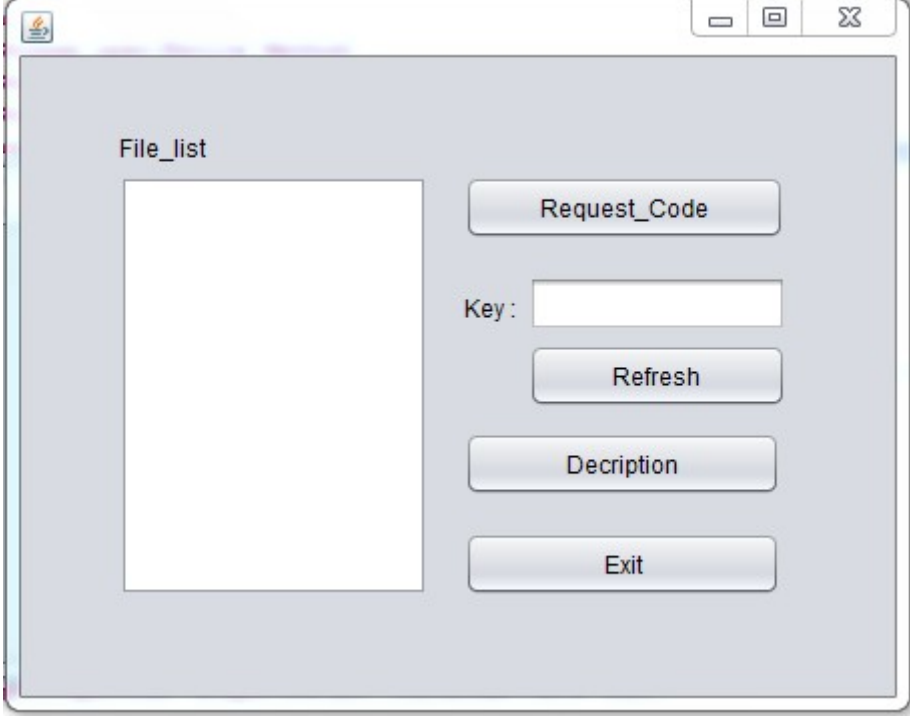
File Chooser Form



A screenshot of a 'File Chooser' dialog box. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area has a light gray background and is titled 'Choose file'. It contains a single text input field. To the right of the input field is a button labeled 'Browse..'. At the bottom, there are two buttons: 'Back' and 'Done'.

Choose file

User Main Form



The screenshot shows a graphical user interface window titled "User Main Form". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- A label "File_list" positioned above a large, empty white rectangular box.
- A button labeled "Request_Code" located to the right of the "File_list" box.
- A label "Key:" followed by a small white text input field.
- A button labeled "Refresh" located below the "Key:" input field.
- A button labeled "Decription" (note the spelling) located below the "Refresh" button.
- A button labeled "Exit" located at the bottom right of the main content area.

CHAPTER 7: CONCLUSIONS

7.1 Conclusion

The whole system has been evaluated with simple data. It is absolutely a menu driven system. The present system is free from all sorts' problem and drawbacks of existing system. This also feasible.

This software was developed with grater user friendliness, because entire program are menu driven. So a new user can use it easily.

The efficiency of the system can be improved by applying some more modification. It would be necessary to make few corrections in the programs has on the changes in the system and users advanced need. This soft provides facility for future needs.

7.2 Limitations of the System

This project has an assumption that is both the sender and receiver must have shared some secret information before imprisonment. Pure cryptography means that there is none prior information shared by two communication parties.

Technology constraint:

The problem encountered here is searching information about computer security through Data Encryption and Key Algorithm and another problem is since the secret key has to be sending to the receiver of the encrypted data, it is hard to securely pass the key over the network to the receiver.

Time constraint:

The time giving for the submission of this project work was not really enough for the researcher to extensively carry out more research on this work.

Financial constraint:

There was not enough money to extensively carry out this work.

7.3 Future Scope of the Project

The project “Advanced Encryption system” is designed for many future additions so that any user requirements can be made easy. Though the system is working on various assumptions it can be modified easily to a kind of requirements.

Future enhancements are possible even in specific modules as entire systems are computerized and modifiable approach. The system is flexible enough to incorporate new database to existing one. Since the entire system is developed in a modular approach, modification if necessary can be done on specific module without distributing the system.

REFERENCES

- www.google.com
- http://en.wikipedia.org/wiki/AES_algorithm
- “Data Communications and Networking”, 4th Edition, by Behrouz A. Forouzan, Tata McGraw Hill
- William Stallings, “Cryptography and Network Security”.
- AES Key Generator for default keys used:

GLOSSARY

APPENDIX A

APPENDIX B

