

SECURE DIGITAL FILE



By

Shahzaib Hassan

Riaz Ahmed

Masooma Zahra

Mozzama Rani

Submitted to the Faculty of Computer Software Engineering

National University of Sciences and Technology, Islamabad

ABSTRACT

With the advent of the modern information age, cyber war is characterized as 5th generation warfare where adversaries fight for the secret sensitive information. Many techniques have been used since the ancient time to keep data secure at rest and in transmission and are top priority in government organizations therefore secure storage is very important.

Modern day security concepts stand on three primary pillars known as Confidentiality, Integrity and Availability a.k.a. The CIA Triad. The product in development (Secure Digital File) will focus mainly on the Confidentiality of data through encryption/Decryption.

Secure Digital File will be responsible to remotely encrypt/decrypt the data stored on any system and ensure that data is not accessible to unauthorized personnel. Interactive interface will be provided to ensure ease of usage to system users. Software will be compatible with the already hardware installed at the government organization.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the SRS– Secure Digital File carried out by Shahzaib Hassan, Riaz Ahmed and Masooma Zahra in supervision of Asst. Prof M.M.WaseemIqbal for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Asst. Prof M.M. WaseemIqbal

CSE DEPARTMENT

MCS

DECLARATION

No portion of the work presented in this dissertation has been submitted in

Support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and cooperation,

a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support which makes us what we are. We are extremely grateful to our project supervisor Asst. Prof M.M. WaseemIqbal from MCS who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course work. Their knowledge, guidance and training enabled us to carry out this whole work.

Finally we are grateful to the faculty of Computer Software Department of the Military College of Signals, NUST.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

TABLE OF CONTENTS

1	INTRODUCTION.....	13
1.1	Overview	13
1.2	Problem Statement	13
1.3	Approach	13
1.4	Scope	13
1.5	Contributions:.....	14
1.6	Organization	14
1.7	Deliverables	15
2	LITERATURE REVIEW	17
2.1	XTS-AES Cryptographic Algorithm.....	17
2.2	Ordering Convention for the Cipher text Stealing Case	17
2.3	Key Management	18
3	OVERALL DESCRIPTION.....	20
3.1	Purpose.....	20
3.2	Intended Audience and Reading Suggestions.....	20
3.2.1	Intended audience	20
3.2.2	Reading suggestions.....	21
3.3	Product Scope	21

3.4	OVERALL DESCRIPTION	22
3.4.1	Product Perspective	22
3.4.2	Product Functions	22
3.4.3	User Classes and Characteristics.....	22
3.5	Design and Implementation Constraints	23
3.6	User Documentation	23
3.7	Assumptions and Dependencies	23
3.8	External Interface Requirements	24
3.8.1	User Interfaces	24
3.8.2	Hardware Interfaces	24
3.8.3	Software Interfaces.....	24
3.9	System Features	24
3.9.1	Remotely File Encryption:	24
3.9.2	Remotely File Decryption:.....	25
3.9.3	Key Management	27
3.10	Other Nonfunctional Requirements	27
3.10.1	Performance Requirements.....	27
3.10.2	Security Requirements.....	27
3.10.3	Software Quality Attributes	28

4	DESIGN AND DEVELOPMENT	30
4.1	Purpose	30
4.2	Project Scope.....	30
4.3	Definitions	30
4.4	References	31
4.4.1	NIST Standard:	31
4.5	Overview of Document	32
4.6	Work Breakdown Structure.....	33
4.7	System Architecture Description	34
4.8	Overview of Modules.....	34
4.9	Structure and Relationships.....	35
4.9.1	System Block Diagram	35
4.9.2	User View (Use case Diagram).....	36
4.9.3	Sequence Diagram	45
4.9.4	Implementation View (Class Diagram)	49
4.9.5	Dynamic View (Activity Diagram)	51
4.10	User Interface	52
4.11	Detailed Description of Components.....	53
4.11.1	Application UI.....	53

4.11.2	File Manager	55
4.11.3	MAC Manager	56
4.11.4	Key Management	58
4.12	Reuse and Relationships to other products.....	59
4.13	Design Decisions and tradeoffs	60
5	SYSTEM IMPLEMENTATION	62
5.1.1	User Access a Modified File	62
5.1.2	Server receive request	63
5.1.3	For Settings Menu.....	63
5.1.4	For User Management	64
6	ANALYSIS AND EVALUATION	66
6.1	Test Items	66
6.2	Features to be tested	67
6.3	Approach	67
6.4	Item Pass/Fail Criteria	67
6.5	Suspension Criteria and Resumption Requirements	68
6.6	Test Deliverables	68
6.6.1	User Interface Testing:.....	68
7	FUTURE WORK	76

8 CONCLUSION	78
7.1 Overview	78
9 BIBLIOGRAPHY	80
APPENDIX A	81

TABLE OF FIGURES:

Figure 4-1 System Block Diagram	35
Figure 4-2 Use case Diagram.....	36
Figure 4-3-1 Sequence Diagram (Upload and Generate Modified File)	45
Figure 4-3-2 Sequence Diagram (File Management)	46
Figure 4-3-3 Sequence Diagram (File Decryption)	46
Figure 4-3-4 Sequence Diagram (Key Management and Location).....	47
Figure 4-3-5 Sequence Diagram (Return Modified File)	48
Figure 4-4 Class Diagram.....	49
Figure 4-5 Activity Diagram.....	51
Figure 4-6 User Interface Demo.....	52

TABLE OF TABLES:

Table 1-1 : Deliverables.....	15
Table 4-1 : Class Description	50
Table 6-1: Test Cases ForTo generate Modified File	68
Table 6-2: Test Cases For Access Modified File	68
Table 6-3: Test Cases For View All Files	69
Table 6-4: Test Cases ForUser will receive an Email.....	69
Table 6-5: Test Cases For Attempt Details of file	70
Table 6-6: Test Cases For Location.....	70
Table 6-7: Test Cases For To Lock/Unlock File for a person.....	71
Table 6-8: Test Cases For Add to allowed/disallowed list	71
Table 6-9: Test Cases For Import CSV file.	72
Table 6-10: Test Cases For Download Original/Modified file from server.....	72
Table 6-11: Test Cases For Add user	73
Table 6-12: Test Cases For Remove/Update User Profile	73
Table 6-13: Test Cases For Add Employee	74

CHAPTER: 1
INTRODUCTION

1 INTRODUCTION

1.1 Overview

The Secure Digital File is aimed for public and private sector organizations. The main purpose of this project is the development of an application that would allow the users to give you remote access control (Read/Write/Execute) of files via the network.

1.2 Problem Statement

There are a number of protocols designed and implemented to monitor the file remotely. The importance of Control factor on data is still there as anyone can do anything to it without your permission. So it must be taken into consideration.

1.3 Approach

The main focus of the project is to provide a software based implementation for On-The-Fly encryption of the data using XTS mode of AES which has been developed by IEEE Security in Storage Working Group (SISWG) and approved by National Institute of Standards and Technology (NIST).

1.4 Scope

As the purpose of this project is the development of an application that would allow the users to give you remote access control (Read/Write/Execute) of files via the network. So it can be utilized in military institutions and universities. Especially for data control and security. The software will provide desktop application with user friendly Interface. Aim & Objectives

The objectives of project include:

Using software engineering techniques for gathering requirements during the development process, designing the software, implementing and testing requirements gathered.

- To learn Java programming language.
- To learn development different files type.
- To learn workings of Windows File Systems.
- To learn Basic Cryptographic concepts especially XTS mode for AES algorithm.

1.5 Contributions:

This is an industrial project designed and developed for NESCOM.

1.6 Organization

The first part of thesis is the abstract which describes the main details of Secure Digital File software, followed by the introduction section which specifies the problem statement, approach, scope and objectives. The literature review section state the various resources read online before the commencement of the project. They include learning about basic cryptographic concepts such as Encryption algorithms and key management. The design and development part illustrate the diagrams which describe the detailed design of the Secure Digital File - its components, interfaces and data necessary for the implementation phase. The analysis and evaluation part give details of the black box testing, unit testing and system integration testing; actual results against expected results. The future work gives states the enhancements that can be applied to the application.

1.7 Deliverables

Table 1-1 : Deliverables

Deliverable Name	Deliverable Summary Description
Software Requirements Specification (SRS) Document	Complete Description of what the system will do, who will use it. Detailed description of functional and non-functional requirements and the system features.
Design Document	Complete description of how the system will be implemented i.e. the detailed design.
Code	Complete code with the API.
Testing Document	The whole system is tested according to the specification described in the SRS document. Black box, unit and System integration testing is done.
Complete System	Complete working system.

CHAPTER: 2
LITERATURE REVIEW

2 LITERATURE REVIEW

The software uses XTS mode of AES which has been developed by IEEE Security in Storage Working Group (SISWG) and approved by National Institute of Standards and Technology (NIST) for the encryption of the data. This mode works within the constraints of the hard disks while keeping the security that the Advanced Encryption Standard (AES) algorithm provides.

2.1 XTS-AES Cryptographic Algorithm

The XTS-AES algorithm is a mode of operation of the Advanced Encryption Standard (AES) algorithm. The Security in Storage Working Group (SISWG) of the P1619 Task Group of the Institute of Electrical and Electronics Engineers, Inc (IEEE) developed and specified XTS-AES in IEEE Std. 1619-2007. The XTS-AES mode was designed for the cryptographic protection of data on storage devices that use of fixed length “data units”. The XTS-AES mode was not designed for other purposes, such as the encryption of data in transit.

The XTS-AES mode is an instantiation of Rogaway’s XEX (XOR Encrypt XOR) tweak able block cipher, supplemented with a method called “cipher text stealing” to extend the domain of possible input data strings. In particular, XEX can only encrypt sequences of complete blocks, i.e., any data string that is an integer multiple of 128 bits; whereas for XTS-AES, the data string may also consist of one or more complete blocks followed by a single, non-empty partial blocks. The XTS-AES mode provides confidentiality for the protected data. Authentication is not provided, because the P1619 Task Group designed XTS-AES to provide encryption without data expansion, so alternative cryptographic methods that incorporate an authentication tag are precluded. In the absence of authentication or access control, XTS-AES provides more protection than the other approved confidentiality-only modes against unauthorized manipulation of the encrypted data.

2.2 Ordering Convention for the Cipher text Stealing Case

If the length of the data units for an instance of XTS-AES is not an integral multiple of the block size, then the plaintext, as a sequence of complete blocks, P_0, P_1, \dots, P_{m-1} , followed by a single, non-empty partial block P_m , where m is a positive integer determined by the length of the data unit.

In this case, the encrypted form of the data unit, i.e., the cipher text, has the same structure: a sequence of complete blocks, denoted C_0, C_1, \dots, C_{m-1} , followed by a single, non-empty partial block C_m , whose length is the same as the length of P_m .

For some implementations, an alternative ordering convention, in which the positions of C_{m-1} and C_m are swapped, may be desirable for the physical storage of the bits, because that ordering corresponds more closely with the generation of the cipher text. In particular, C_m is the truncation of a block that is derived from P_{m-1} , and C_{m-1} is derived from P_m , concatenated with the discarded bits from the truncation.

2.3 Key Management

Key management is important for XTS-AES, as for any keyed cryptographic algorithm. Consistent with the 220 block limit for a data unit, an implementation of XTS-AES may further restrict the length of the data units for any key. For example, an implementation may support only data units that are sequences of complete blocks. In this case, the ciphertext stealing components in the implementations of the XTS-AES-Enc and the XTS-AES-Dec procedures would be unnecessary, and these procedures essentially would be reduced to the XTS-AESblockEnc and the XTS-AES-blockDec procedures.

Similarly, an implementation may restrict its support to either the 256-bit key size (for XTS-AES-128) or the 512-bit key size (for XTS-AES-256). Restrictions on the supported lengths of the key or the data units may affect interoperability with other implementations.

CHAPTER: 3
OVERALL DESCRIPTION

3 OVERALL DESCRIPTION

This part of the document contains information about the product, its features, perspective, users' characteristics and constraints.

3.1 Purpose

The purpose of the Software Requirements Specification (SRS) is to give the user a clear and precise description of the functionality of the Cyber file controller, an application that would be built to give remotely control access of data.

This document is aimed to eliminate ambiguities and misunderstandings that may exist. For the user, the SRS will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation and maintenance.

The main purpose of this project is the development of a system that would allow the users to encrypt/decrypt the file remotely, stored on any system, through an easy to use interface.

3.2 Intended Audience and Reading Suggestions

3.2.1 Intended audience

The intended readers of the SRS are the students and as well as military employees who will have the system implemented. This document serves as an agreement between both parties (Development team and the concerned authorities) regarding the product to be developed.

- **Project Supervisor (Asst. Prof. M.M. WaseemIqbal)**

This document will assist in supervision and guiding the team. Further, document will act as a reference to ensure completion of all requirements and proper implementation.

- **Developers and Testers (Project group)**

This document provides the guideline for developers to code during implementation phase and testers to create test cases during testing phase.

- **Project Evaluation Team (MCS NUST)**

It will help the evaluation team to evaluate the progress of FYP project. The document will provide the evaluators with the scope, requirements and details of the software to be developed. It will also be used as basis for the evaluation of the implementation of the project.

3.2.2 Reading suggestions

It would be suggested to the users to go through the requirement section thoroughly.

For the developers it is suggested that they read and understand the product scope, overall description and system features thoroughly.

Testers should go through the operating environment, constraints, and the non-functional requirements before developing the test scenarios for the system.

3.3 Product Scope

The document only covers the requirements specifications for Secure Digital File All the external interfaces and the dependencies are also identified in this document.

For	Universities and military Institutions
What	Giving remotely Control access of data to user
The	Cyber file Controller
Is	An application
That	Provides transparent, easy to use, remotely, On-The-Fly encryption and decryption procedure for data stored on any system.

Cyber file Controller shall provide an application that would run on top of any Windows operating system. The scope of this project is to develop encryption/decryption application for the data stored on any system. It is mainly focused on encrypting/decrypting the files remotely (excluding executable files). The encryption/decryption process is implemented using XTS-AES algorithm only. The software will provide desktop application with user friendly Interface.

3.4 OVERALL DESCRIPTION

3.4.1 Product Perspective

Cyber File Controller is new project built for military and sensitive organizations to provide security services to removable disks. Other products available in market but they are mainly focusing on data integrity. It is built for organizations who do not want to use third party tools due to danger of sensitive data being leaked and possible backdoors.

3.4.2 Product Functions

The main features of Cyber File Controller are highlighted below:

- NIST-approved implementation of XTS-AES 256-bit algorithm for disk encryption
- NIST-approved implementation of Password Based Cryptography for key expansion.
- Supports all files except the executable ones.
- Supports remotely encryption for file stored on any system, which is connected to internet.
- Supports confidentiality and integrity of data.
- Authentication and recovery mechanism.
- Remotely and fully automated encryption/decryption functionality.
- Supports On-The-Fly encryption /decryption.

3.4.3 User Classes and Characteristics

Defining system users for the Secure Digital File

3.4.3.1 UCC-1 Universities Students and Military Employee

Universities Students and Military Employee will be the general users who will use the system. They will be interacting with desktop application.

3.4.3.2 UCC-2 Developers

Cyber File Controller application and source code will be provided to them for general development or future possible maintenance.

3.4.3.3 UCC-3 Software Testers

Software will be used for Alpha Testing by our project group and Beta Testing by concerned authorities.

3.4.3.4 Operating Environment

This included all the specifics required for software to be built.

3.4.3.5 Hardware

Desktop/ Laptop personal computer, Cell phone

3.4.3.6 Software

Windows 7/8/10 Operating System

3.5 Design and Implementation Constraints

- For Cyber File Controller to operate successfully there must be an internet connection.
- All files will be encrypted except executable ones.
- Time of encryption will depend upon the size of file.
- As Cyber File Controller require sensitive permissions which can only be granted in the Administrator mode in Windows OS therefore only Administrator users can access complete functionality of Cyber File Controller

3.6 User Documentation

User manuals will be provided which will take users step by step to complete remotely encryption/decryption process using Secure Digital File A tutorial will provide a quick start, a walk-through of major system features, and further reference sources for the complete system features.

3.7 Assumptions and Dependencies

- Cyber File Controller will be run with these minimum hardware specifications:
 - Intel Core i3 processor or equivalent
 - 1 GB RAM or higher
- Cyber File Controller will run on Windows 7/8/10 Operating system

3.8 External Interface Requirements

3.8.1 User Interfaces

The homepage will contain buttons for all the important functionalities provided by the software (Get file, Store, file encryption, file decryption). The software will run in the background so its icon will be available in the task bar to access it easily at all times. Each screen will be explanatory regarding the options and functionality provided by the software.

3.8.2 Hardware Interfaces

- Cyber File Controller requires at least internet connection to work on.
- Software will provide on-the-fly encryption, therefore minimum 1 GB RAM is required.

3.8.3 Software Interfaces

- Cyber File Controller will require Windows 7/8/10 operating systems

3.9 System Features

System features are organized in functional hierarchy so that the main functions of the system will be understandable.

3.9.1 Remotely File Encryption:

This feature will allow the user to remotely encrypt a complete non-executable file stored on any system

3.9.1.1 *Description:*

One of the basic functionalities of the system is to provide remote encryption for a any file given that it is a non-executable. When a user wants to encrypt a file with data already stored in it, this feature option is used.

3.9.1.2 *Stimulus/Response:*

- **Basic Data Flow**

- User is provided with encryption option.(files only)
- User will select file encryption option.
- User is provided with a list already encrypted files.
- User will select the upload the file, he wants to encrypt.
- User will confirm the encryption process after it has completed successfully.
- Once the file is encrypted user is provided with decryption option.
- User will select only file that is already encrypted otherwise warning message would be shown.
- User can select an encrypted file which may be located remotely.
- User will confirm the decryption process after it has completed successfully.

3.9.1.3 Functional Requirements:

REQ-1. The system shall provide the option to remotely encrypt/decrypt a complete file.

REQ-2. The system shall present a list already encrypted files.

REQ-3. The system shall allow the user to select a non-executable file.

REQ-4. The system shall allow the file to be encrypted remotely in place.

REQ-5. The system shall provide the size of the selected file.

REQ-6. The system shall collect data of the selected file.

REQ-7. The system shall remotely encrypt the file as the final step.

3.9.2 Remotely File Decryption:

This feature will allow the user to remotely decrypt a already encrypted file stored on any system

3.9.2.1 Description:

One of the basic functionalities of the system is to provide remote decryption for a any already encrypted file. When a user wants to decrypt a file with data already stored in it, this feature option is used.

3.9.2.2 Stimulus/Response:

- **Basic Data Flow**
 - User is provided with decryption option.(Already encrypted files only)
 - User will select file decryption option.
 - User is provided with a list already decrypted files.
 - User will select the file from the list, he wants to decrypt.
 - User will confirm the decryption process after it has completed successfully.
- **Alternative flow**
 - User is provided with the interface.
 - User wants to decrypt a file which is not already encrypted so a warning message would appear.

3.9.2.3 Functional Requirements:

REQ-1. The system shall provide the option to remotely decrypt already encrypted file.

REQ-2. The system shall present a list already encrypted files.

REQ-3. The system shall allow the user to select from only that list.

REQ-4. The system shall allow the file to be decrypted remotely in place.

REQ-5. The system shall provide the size of the selected file.

REQ-6. The system shall collect data of the selected file.

REQ-7. The system shall remotely decrypt the already encrypted file as the final step.

3.9.3 Key Management

3.9.3.1 *Description:*

A 512-bit key is required during the encryption process of the data. Secure Digital File will generate this key and will store it on the hard drive. Key generation should comply NIST SP800-133, authentication and recovery key mechanism will be using password based cryptography standards.

3.9.3.2 *Stimulus/Response:*

No user interaction required

3.9.3.3 *Functional Requirements:*

REQ-1. The system will generate a 512-bit key for encryption/decryption process.

REQ-2. The system will generate separate keys for separate volumes.

REQ-3. The system will generate keys using entropy pool of host operating system.

REQ-4. The system will store the keys in encrypted form.

REQ-5. The protection keys will be generated using user provided password by implementing password based cryptography standards.

REQ-6. The system will provide the recovery key in case user forgets the password.

3.10 Other Nonfunctional Requirements

3.10.1 Performance Requirements

REQ-1. The system shall not have an encryption rate of more than 256MB/minute.

3.10.2 Security Requirements

REQ-2. The system shall allow the user to upload a file without exposing its data.

REQ-3. The system shall provide no access to the file once it's decrypted.

REQ-4. The system shall provide no way to recover data if the encrypted file is deleted.

3.10.3 Software Quality Attributes

3.10.3.1 Usability:

REQ-5. The system shall provide wizards for a specific functionality.

3.10.3.2 Data Integrity:

REQ-6. The system shall not change or corrupt user data during encryption or decryption process.

3.10.3.3 Confidentiality

REQ-7. The system shall keep data confidential. Data must not be stored or sent over internet.

REQ-8. The system shall provide no way to access the encrypted file.

3.10.3.4 Standard Compliance

REQ-9. The system shall only use implementation of XTS-AES which is verified by NIST against the known test vectors provided by NIST itself.

REQ-10. The system shall only use implementation of SHA which is verified by NIST against the known test vectors provided by NIST itself.

CHAPTER: 4
DESIGN AND DEVELOPMENT

4 DESIGN AND DEVELOPMENT

4.1 Purpose

This software design specification (SDS) document describes the architecture and system design of Secure Digital File. It mostly contains different design diagrams and their explanation. The document is intended to inform stakeholders, developers and support team at organization the details of the design and the design process. This document will help the developer(s) in implementation and maintenance of the Software.

4.2 Project Scope

The scope of this project is to develop encryption software for the data stored on hard drives. Secure Digital File shall provide an application that would run on top of any Windows operating system even in cell phones. The scope of this project is to develop encryption/decryption application for the specific data stored on any system and to determine its location as well. It is mainly focused on encrypting/decrypting the files remotely (excluding executable files). The encryption/decryption process is implemented using MD5 algorithm only. The software will provide desktop application with user friendly Interface.

4.3 Definitions

AES: Advanced Encryption Standards

NIST: National Institute of Science and Technology.

FIPS: Federal Information Processing Standards.

PKCS: Public key Cryptography Standards.

CDB: Critical Data Block.

CDK: Critical Data Key.

4.4 References

4.4.1 NIST Standard:

Encryption of Storage Devices

- Dworkin, M., 2010. Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices, s.l.: NIST Special Publication 800-38E
- Scarfone, K., Spappaya, M. & Sexton, M., 2007. Guide to Storage Encryption Technologies for End User Devices, s.l.: NIST Special Publication 800-111.
- The Advanced Encryption Standard (AES), s.l.: Federal Information Processing Standards- 197.

4.5 Overview of Document

This document shows the design and working of Secure Digital File. It starts from higher level details for a non-technical reader to understand just by seeing the diagrams to the lower level details that aid the developer to code and understand other technical details of the application.

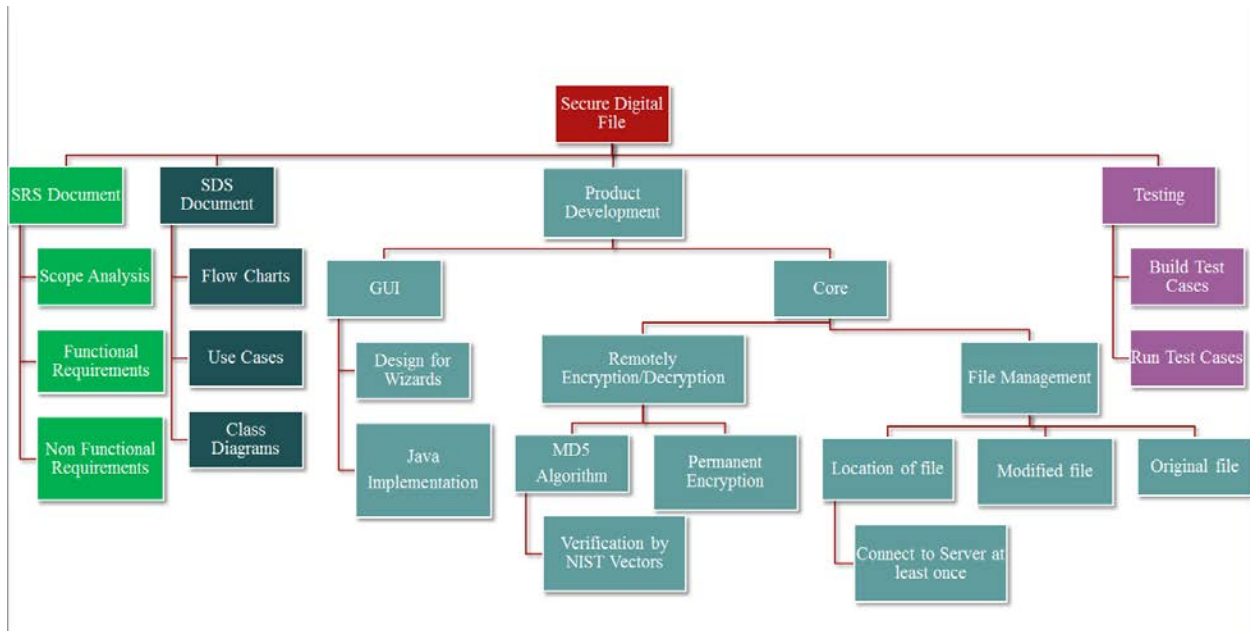
In Section 2, the **System Architecture Description** gives a detailed overview of the application.

Section 2.1 Overview of Modules/Components shows the main component of the application and their inter-relationships. Section 2.2 Structure and Relationships shows the higher level details of system working by the means of System Block, Activity, State Transition, and Use Case diagrams. Lower level details are described using the Class, Sequence diagrams and Structure Chart. Section 2.3 describes how the application is designed to curb the tendency of User Interface Issues and problems during User Interaction.

In Section 3, **Detailed Description of Component** is given to show the working of modules with low level details. It shows the purpose, function, subordinates, dependencies, interfaces, resources, processing and data of the components and their relationships with each other.

Section 4 shows the **Reuse and Relationship to other Products** i.e.; information about work done in the same project before and any reuse of the same work. The section also provides a key to reuse this system for further upgrades.

4.6 Work Breakdown Structure



4.7 System Architecture Description

This section provides detailed system architecture of Secure Digital File application. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

4.8 Overview of Modules

This Secure Digital File application has following required modules. Here we give a brief overview of all these modules. Detailed descriptions of these modules are presented in section 3.

1. Remotely File Encryption:

One of the basic functionalities of the system is to provide encryption for file given that it is a non-executable. When a user wants to encrypt a file with data, this feature option is used.

2. Remotely File Decryption:

The second basic functionality of the system is to provide remote decryption for any already encrypted file. When a user wants to decrypt a file with data already stored in it, this feature option is used.

3. Location of File

In order for the user to get the location of application generated files, user will go to already encrypted file section and resultantly will get the location of that file in the form of Mac and IP addresses.

4. Permanent Encryption

Secure Digital File will provide mechanism to permanently encrypt the file if the user lost File or if it gets into the wrong hands.

5. Original and Modified File Management

Modified and original file can be extracted from the server in case if the user loses the file.

4.9 Structure and Relationships

This section covers the technical description of Secure Digital File. It shows relationships between different components and how system modules are connected. This section also covers working with respect to different point-of-views. This also covers its higher and lower levels details, user interfaces, and system architecture and design pattern.

4.9.1 System Block Diagram

This diagram shows the higher level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

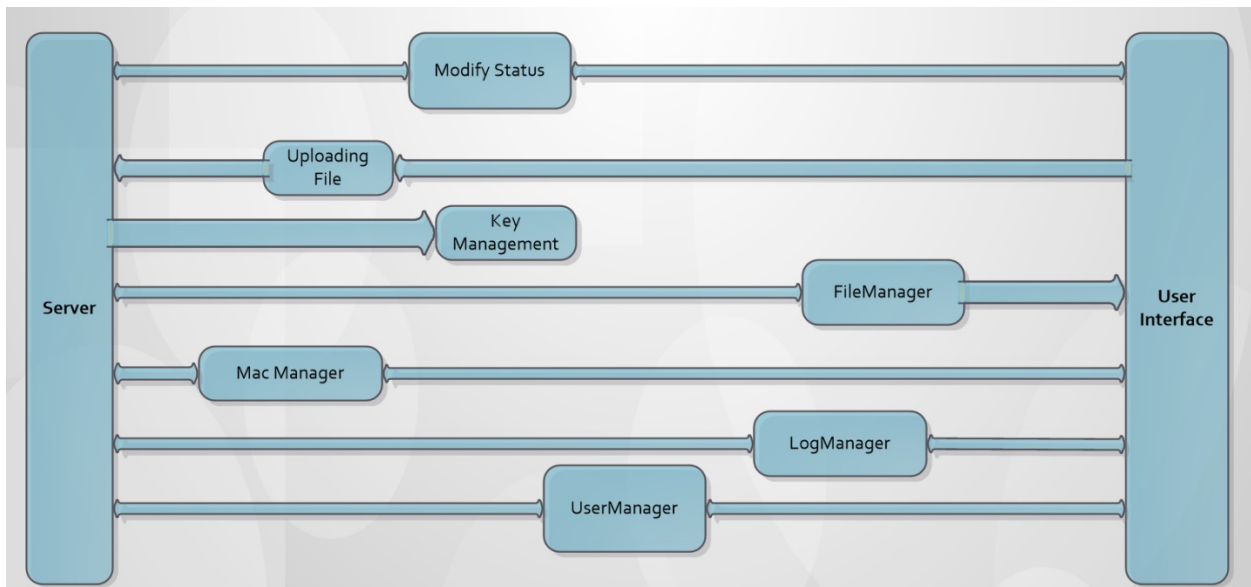
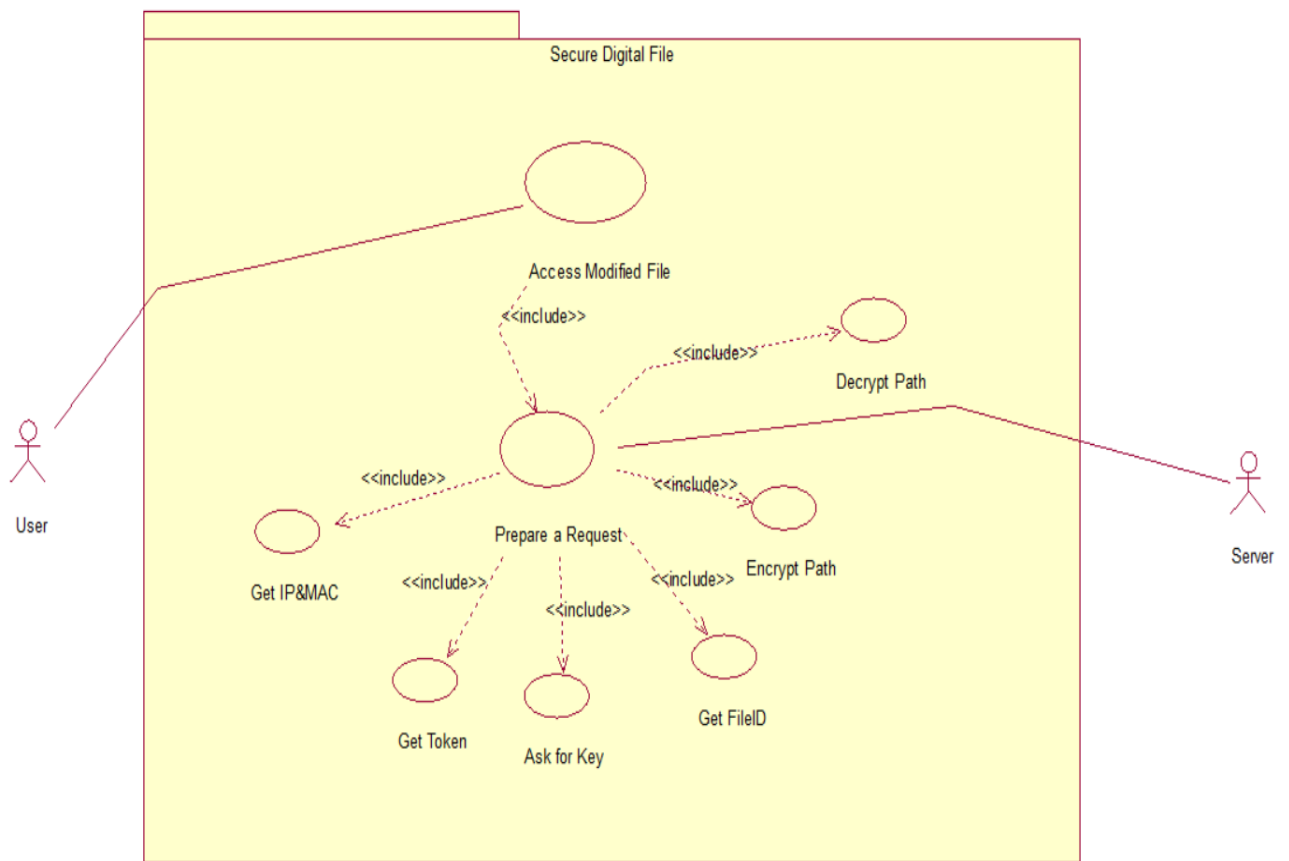


Fig 2.2.1.1 System Block Diagram

4.9.2 User View (Use case Diagram)

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interacts with system.

4.9.2.1 Client Side



4.9.2.2 Server Side

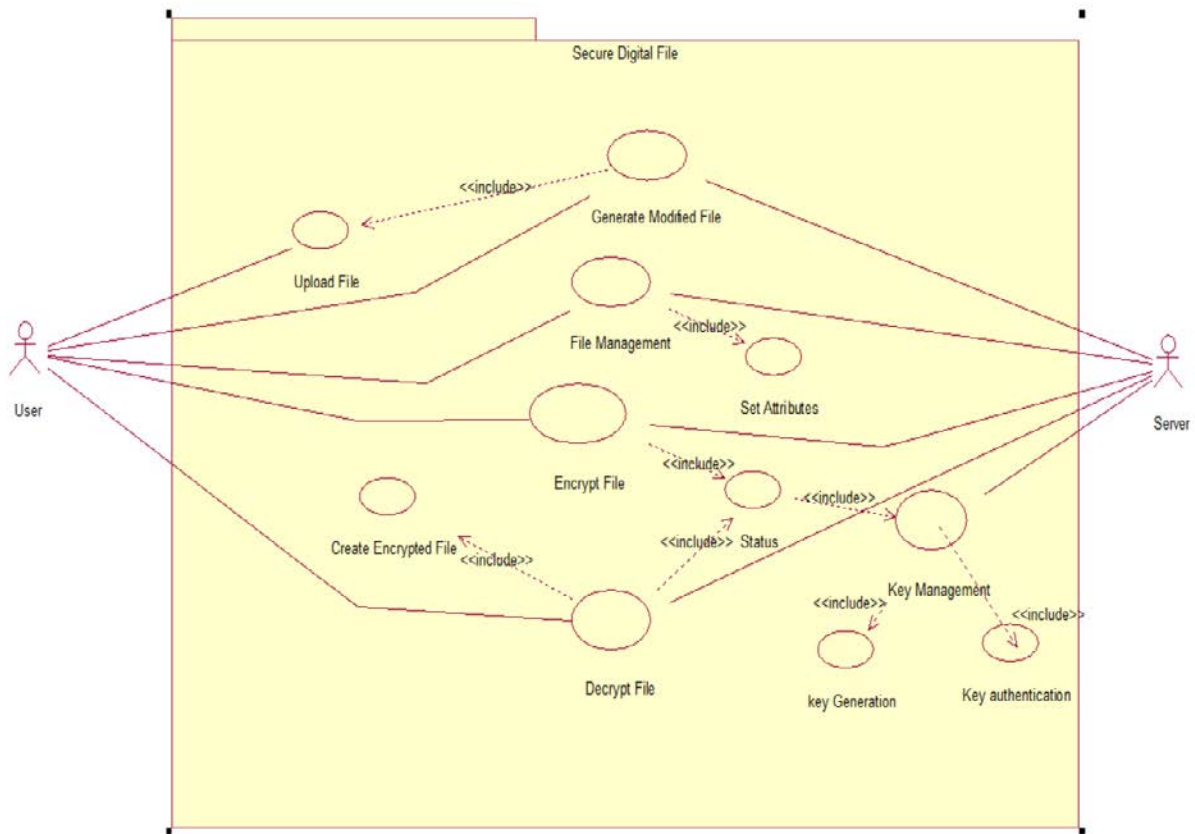


Fig 2.2.2.1 Use case Diagram

4.9.2.3 Actors

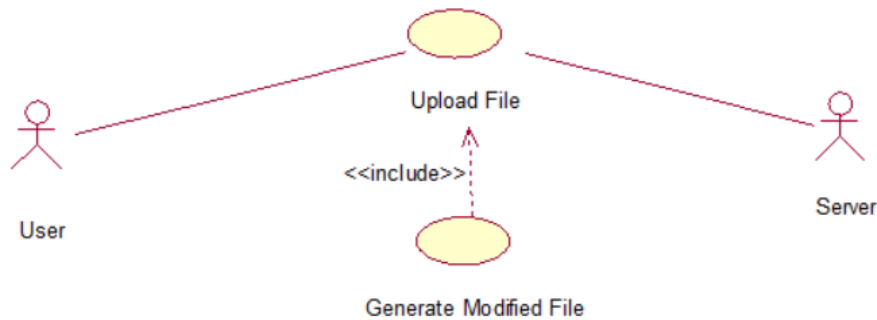
- User
- Server

4.9.2.4 Use Cases

- Upload file
- Generate Modified File
- File Management
- Encrypt File
- Decrypt File
- Location
- Key Management

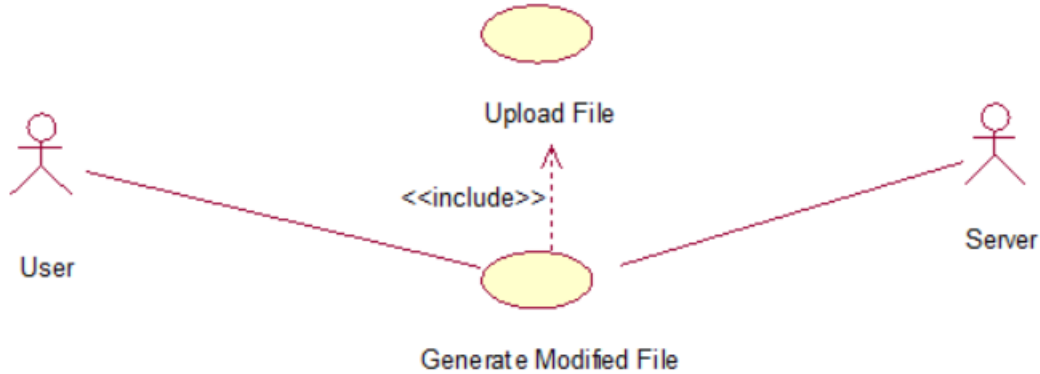
4.9.2.5 Use Case Description

4.9.2.6 UPLOAD FILE



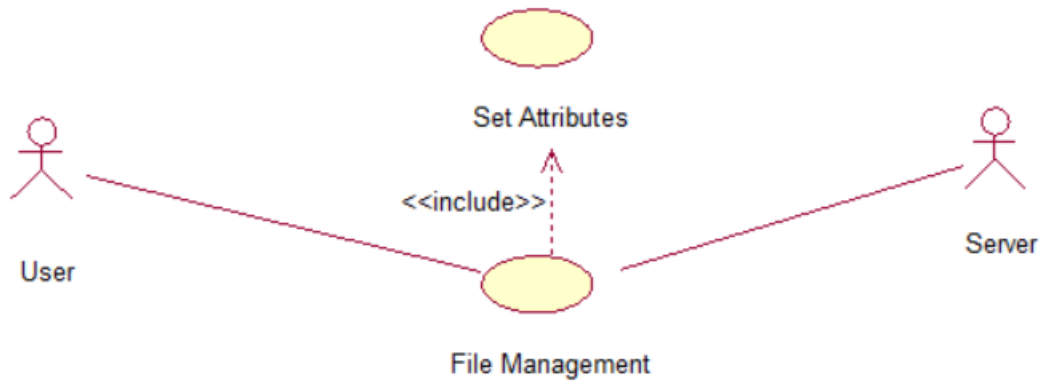
Use Case: Upload file
Actors: User, Server
Use Case Description: This use case will ask the user to upload a file so that server provides encryption for that file given that it is a non-executable.
Normal Flow: <ul style="list-style-type: none">• User will select Upload file option.• User is provided with a list of options either he wants to encrypt it or not.• User will select the options.
Alternate Flow: <ul style="list-style-type: none">• An error message is displayed if an error occurs during the uploading process.
Preconditions: N/A
Post conditions: A file will be uploaded to server.
Includes: Create Encrypted file
Extends: N/A

4.9.2.7 GENERATE MODIFIED FILE



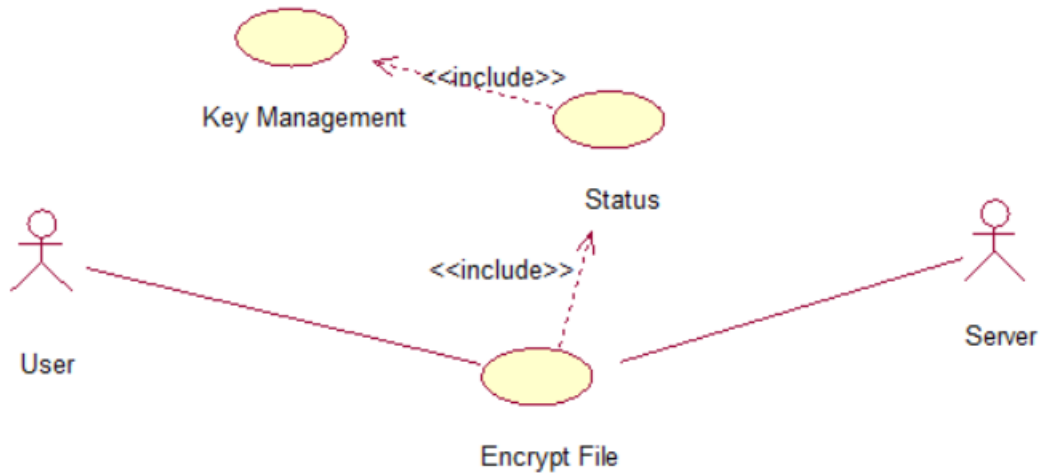
Use Case: Generate Modified File
Actors: User, Server
Use Case Description: This use case provides encryption for an uploaded file given that it is a non-executable.
Normal Flow: <ul style="list-style-type: none"> • User will select Add files option. • User is provided with Status option. • User will select “Not Available” if the user wants to encrypt and can also select date. • User will confirm the encryption process after it has completed successfully
Alternate Flow: <ul style="list-style-type: none"> • An error message is displayed if an error occurs during the encryption process.
Preconditions: N/A
Post conditions: An encrypted file is created which is returned to the user.
Includes: upload file, Key Management, Get file size
Extends: N/A

4.9.2.8 FILE MANAGEMENT



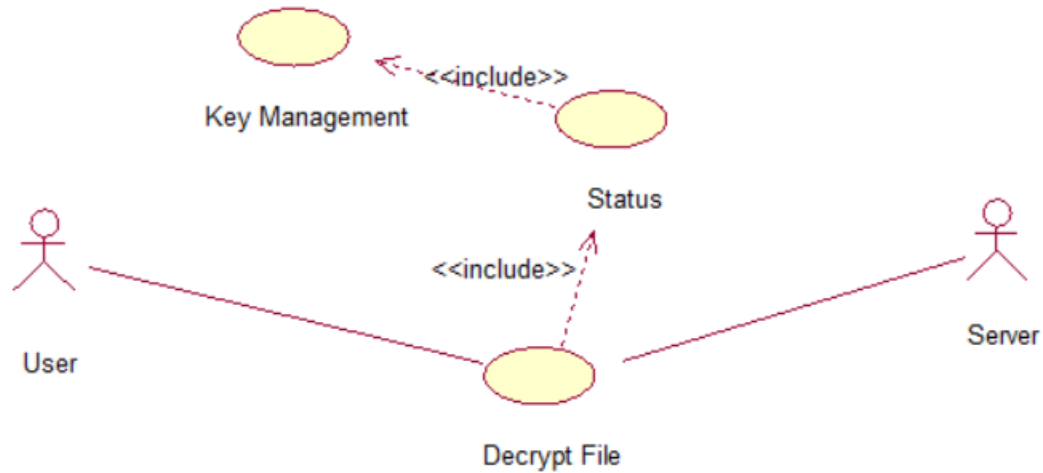
Use Case: File Management
Actors: User, Server
Use Case Description: This use case provides Original file, which was uploaded by the user, and as well as the modified file generated by the server. It will act as normal file and can be downloaded, deleted.
Normal Flow: <ul style="list-style-type: none">• User will select File management option.• User will choose the file which attributes he wants to reset.• User will be provided with Download option in case if original file is lost with a name for the file.• User will confirm the name, location and size of the file before its successful
Alternate Flow: <ul style="list-style-type: none">• An error message is displayed if an error occurs during the process.
Preconditions: N/A
Post conditions: An uploaded File attributes can be changed.
Includes: Provide Original File, Provide modified file, get file size, Set attributes
Extends: N/A

4.9.2.9 ENCRYPT FILE



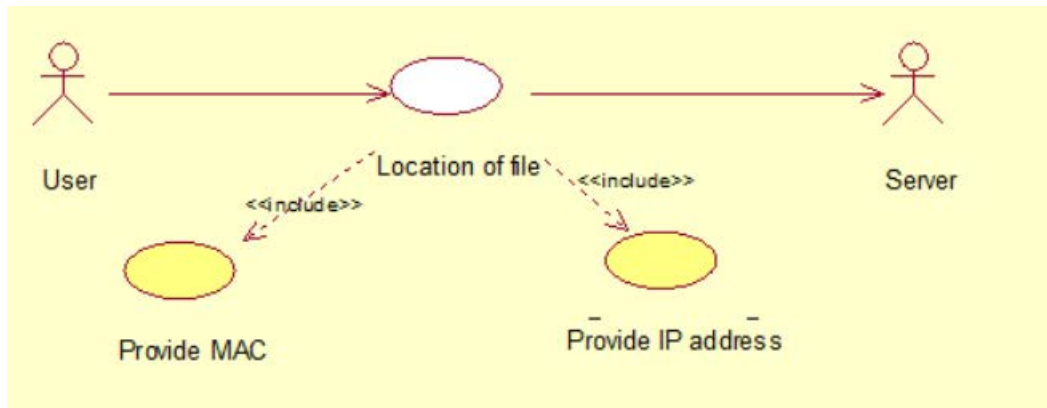
Use Case: Encrypt File
Actors: User
Use Case Description: This use case provides encrypted file to the user in response to uploaded file given that it is a non-executable.
Normal Flow: <ul style="list-style-type: none"> • User will select Add files option. • User is provided with Status option. • User will select “Not Available” if the user wants to encrypt and can also select date. • User will confirm the encryption process and provide encrypted file.
Alternate Flow: <ul style="list-style-type: none"> • An error message is displayed if an error occurs during the encryption process.
Preconditions: N/A
Post conditions: An encrypted file is created which is returned to the user.
Includes: Create encrypted file
Extends: N/A

4.9.2.10 DECRYPT FILE



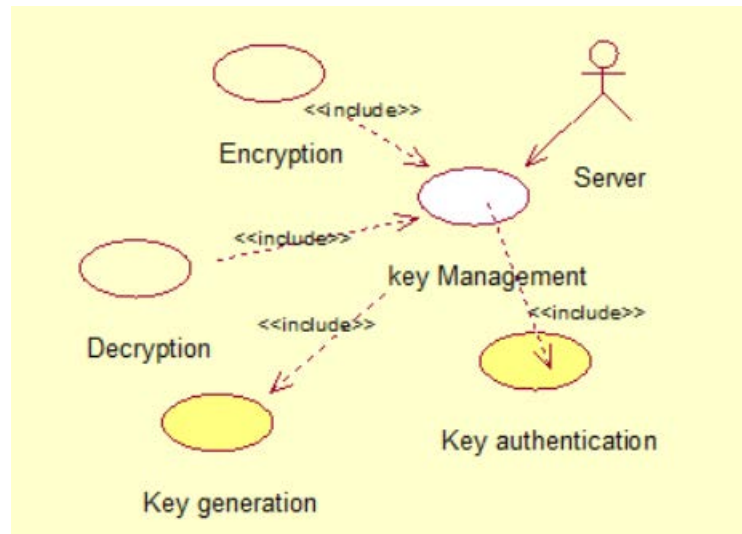
Use Case: Decrypt file
Actors: User, Server
Use Case Description: This option allows the user to access the encrypted data. The data within these files and folders is decrypted On-The-Fly (OTF) when they are selected to be viewed.
Normal Flow: <ul style="list-style-type: none"> • User will click on Status option in order to decrypt the file. • User will be provided with a list of available options. • User will select “Available” in order to decrypt. • Key will be authenticated by the Server if anyone wants to access the file.
Alternate Flow: <ul style="list-style-type: none"> • Volume is not mounted and an error message is generated if the user provides incorrect password for the volume.
Preconditions: Volume should already be created.
Post conditions: Encrypted data is accessible.
Includes: Password Authentication, Lists available drive letters.
Extends: N/A

4.9.2.11 LOCATION



Use Case: Location
Actors: User, Server
Use Case Description: This use case allows the user to locate his file placed in any system through IP and MAC addresses.
Normal Flow: <ul style="list-style-type: none"> • User will click on File management option. • User will click on the file which he wants to locate. • User will be provided with IP and MAC addresses.
Alternate Flow: N/A
Preconditions: File should connect to server at least once.
Post conditions: IP and MAC addresses will be provided
Includes: Provide MAC , Provide IP
Extends: N/A.

4.9.2.12 KEY MANAGEMENT

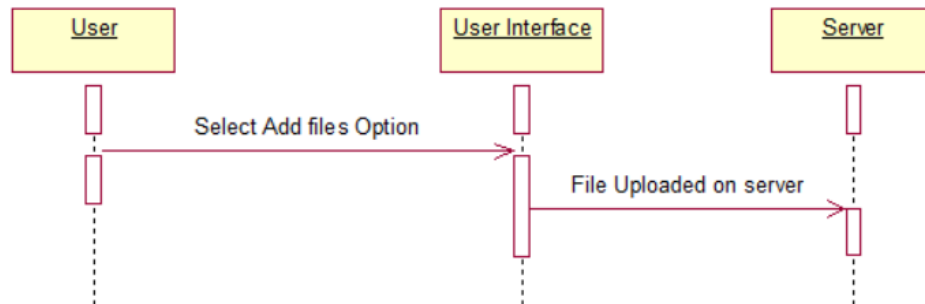


Use Case: Key Management
Actors: Server
Use Case Description: This use case will generate key for encryption and will store it on the server. Then it will be authenticated whenever authorized person wants to decrypt the file.
Normal Flow: <ul style="list-style-type: none"> • Data will be collected for key generation. • Key will be generated as a result of encryption. • Then it will be authenticated whenever decryption process undergoes.
Alternate Flow: N/A
Preconditions: Key generation should be available.
Post conditions: N/A
Includes: Generate key, key authentication, Encryption , Decryption
Extends: N/A.

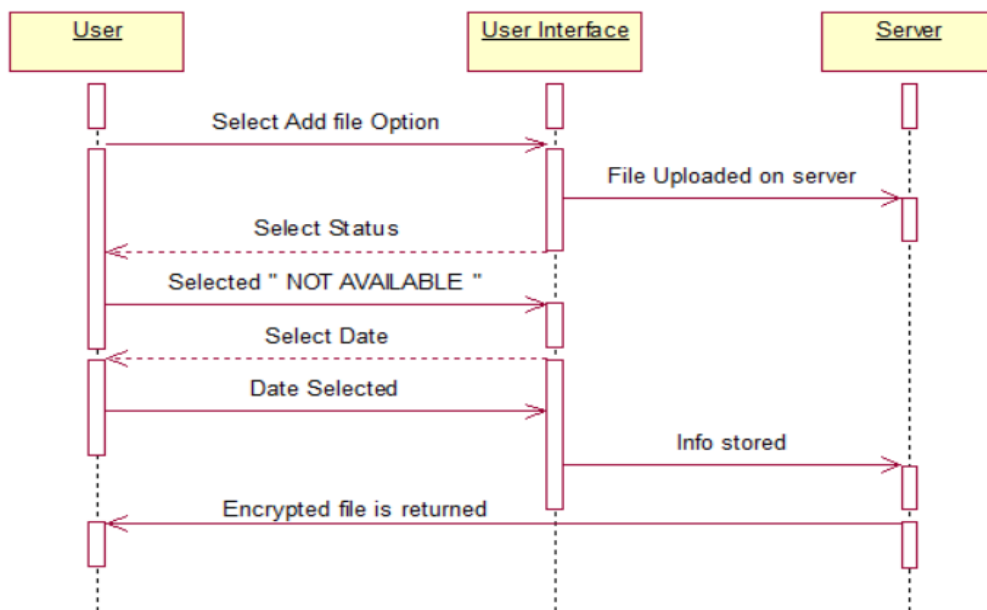
4.9.3 Sequence Diagram

Following sequence diagrams show the sequence of activities performed in all use cases described in section 2.2.2.

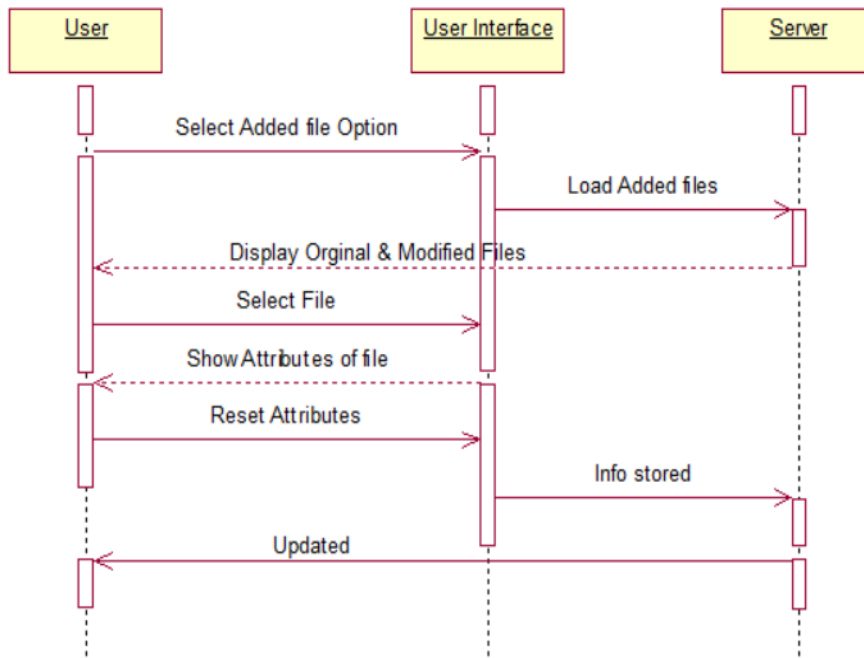
4.9.3.1 Upload



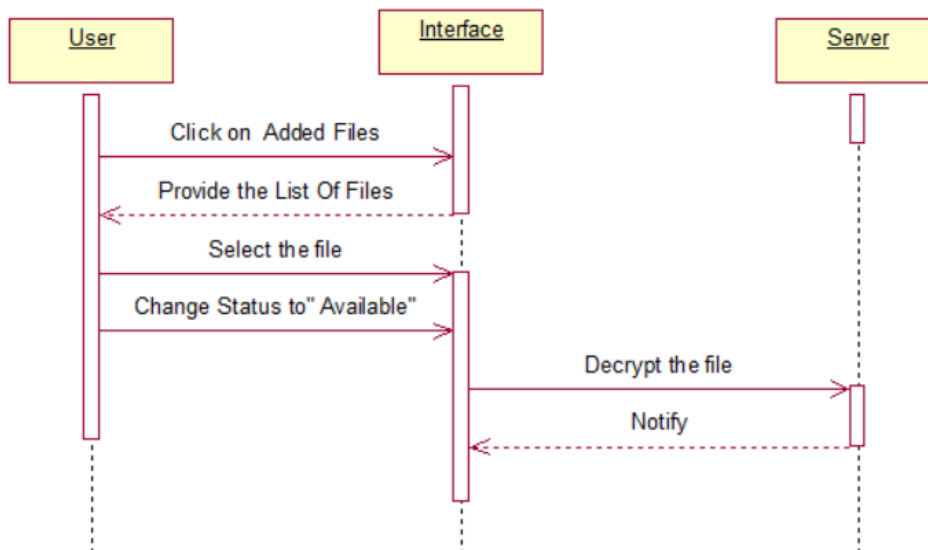
4.9.3.2 Create Encrypted File



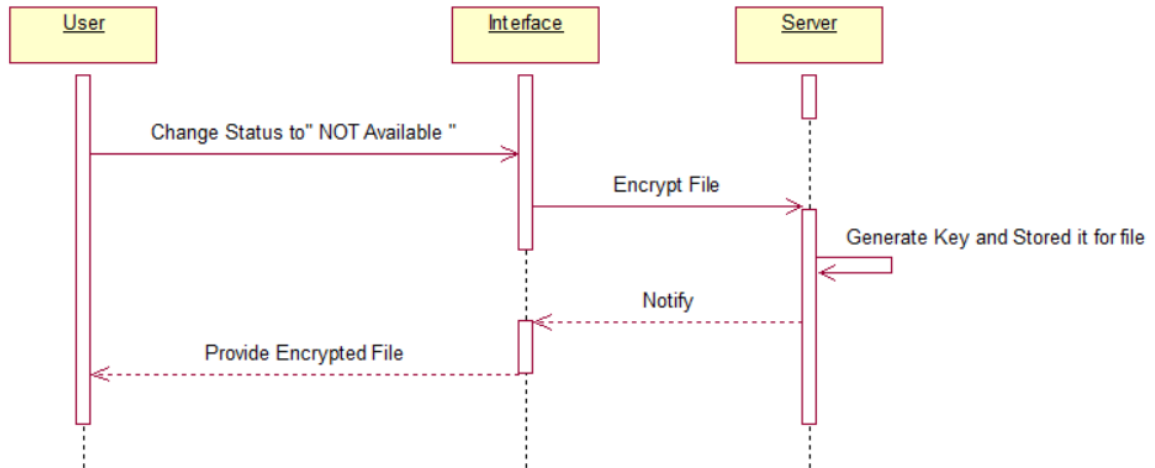
4.9.3.3 File Management



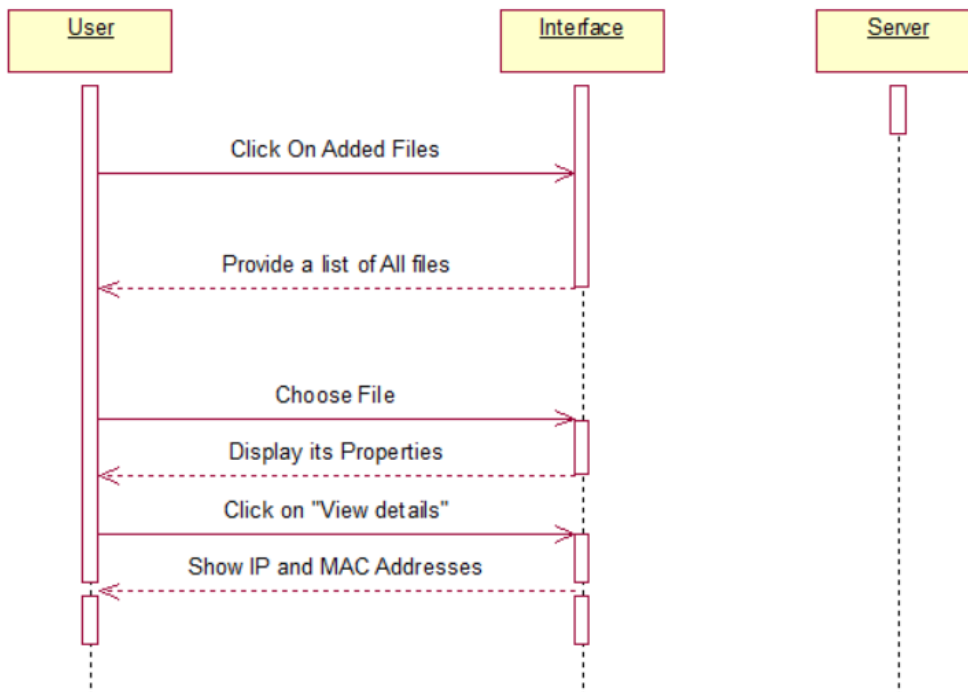
4.9.3.4 File Decryption



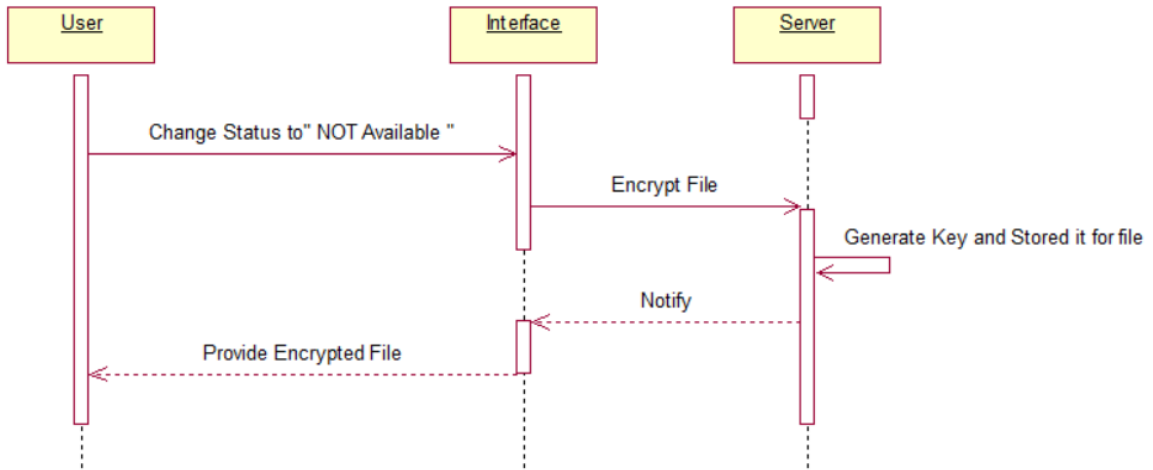
4.9.3.5 Manage key



4.9.3.6 Location



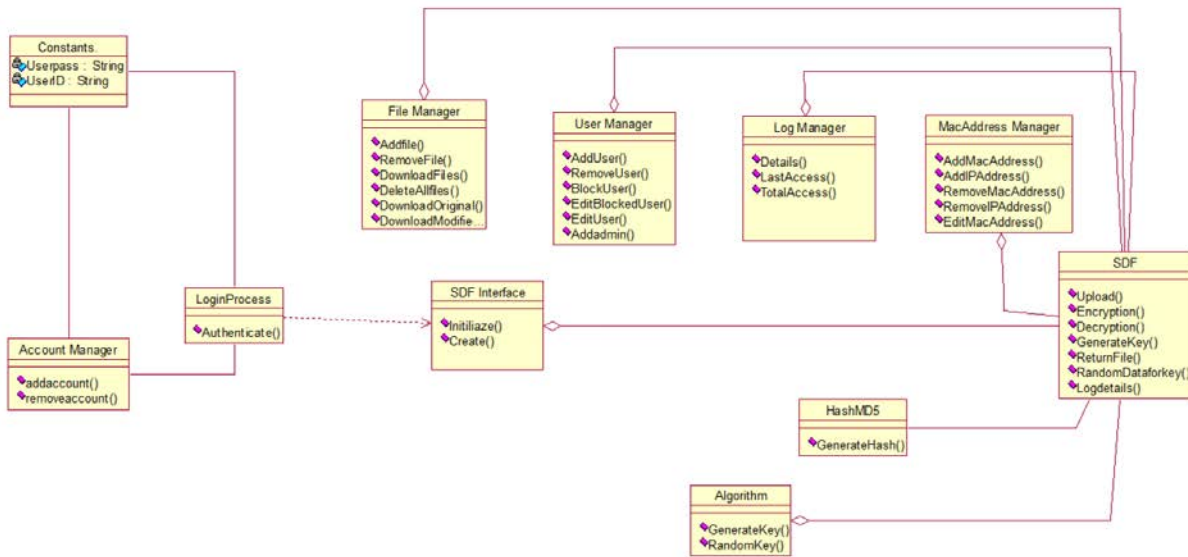
4.9.3.7 Return Encrypted File



4.9.4 Implementation View (Class Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

Server Side



ClientAPP Side

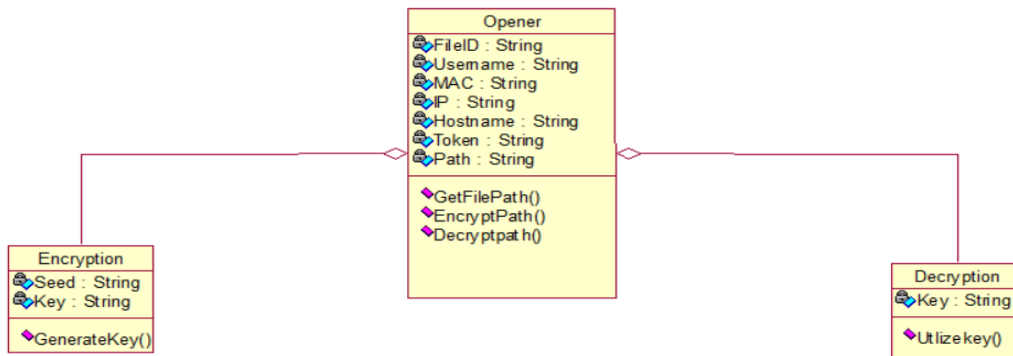
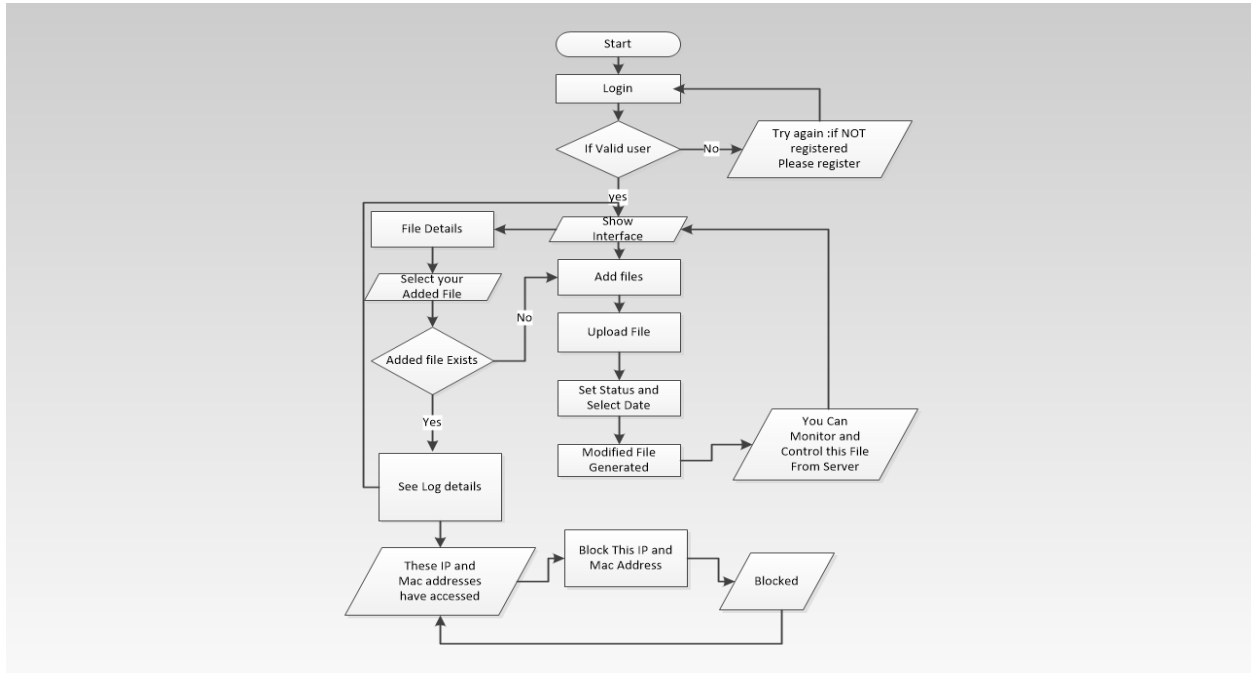


Fig 2.2.4.1 Class Diagram

Classes	Description
<i>Secure Digital File UI</i>	The graphical user interface class contains all the wizards and the graphical interface forms implemented in Java.
<i>Opener</i>	The class in Java, which should be available at client side to run the Modified File.
<i>Algorithm</i>	The key derivation class which contains key derivation function which handles the derivation of master key.
<i>Secure Digital File (SDF)</i>	The main class of the software, which is the base of all. It will initialize all the functions
<i>Hash MD5</i>	The Hashing driver class which contains the MD5 hashing functions.
<i>File Manager</i>	This Class will be interacting with main SDF class and will be responsible for all File actions.
<i>User Manager</i>	This Class will be interacting with main SDF class and will be responsible for all user actions.
<i>Log Manager</i>	This Class will be interacting with main SDF class and will give information about those who have accessed the modified file.
<i>Constraint</i>	Those constraints which are mandatory for login process.
<i>Mac Manager</i>	This Class will be interacting with main SDF class and will inform us about the locations of the user who have accessed the modified file.

4.9.5 Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.



4.10 User Interface

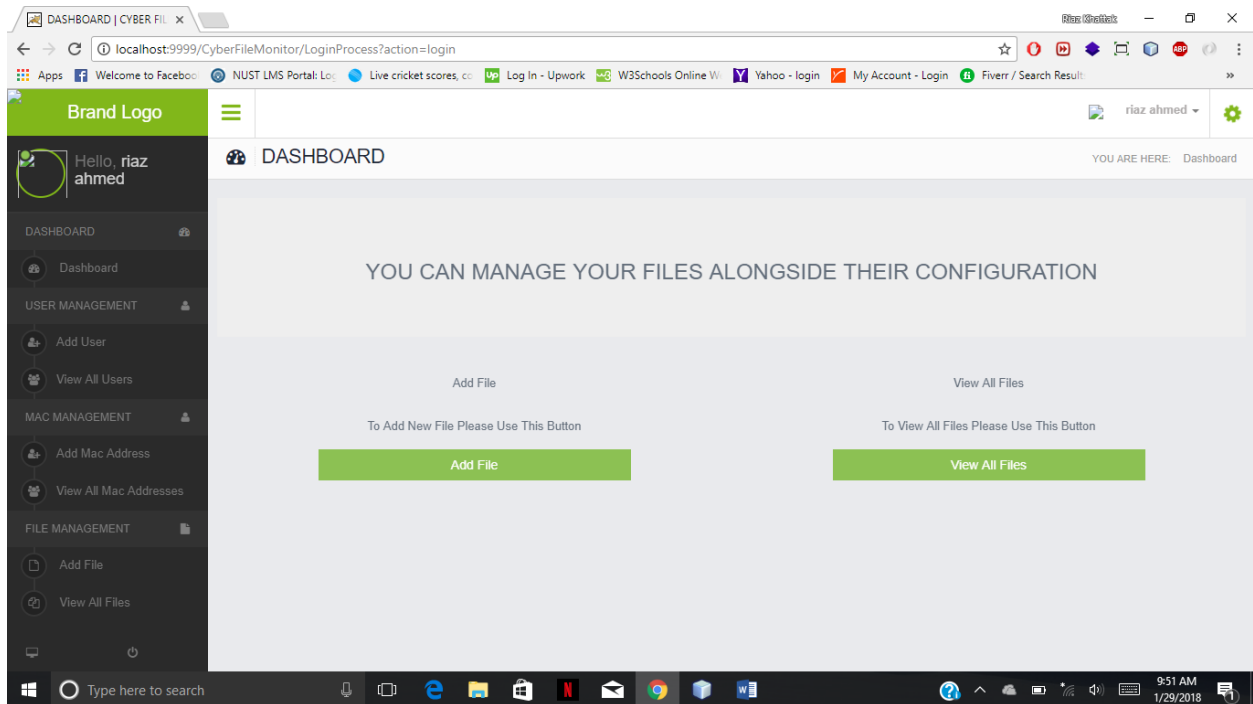


Fig 2.3.1 User Interface Demo

4.11 Detailed Description of Components

This section describes in detail all the modules of Secure Digital File.

4.11.1 Application UI

Identification	<p><i>Name:</i> Application UI</p> <p><i>Location:</i> Client side of the system architecture</p>
Type	UI component
Purpose	<p>The User is able to see different wizards which provide different options and their explanations for the encryption process.</p> <p>This component fulfills some of the functional requirements (as specified in SRS Document) related to user interface in the application:</p> <p>4.1 Remotely File Encryption:</p> <p>One of the basic functionalities of the system is to provide encryption for file given that it is a non-executable. When a user wants to encrypt a file with data, this feature option is used.</p> <p>4.2 Remotely File Decryption:</p> <p>The second basic functionality of the system is to provide remote decryption for any already encrypted file. When a user wants to decrypt a file with data already stored in it, this feature option is used.</p> <p>4.3 Location of File</p> <p>In order for the user to get the location of application generated files, user will go to already encrypted file section and resultantly will get the</p>

	<p>location of that file in the form of Mac and IP addresses.</p> <p>4.4 Permanent Encryption:</p> <p>Secure Digital File will provide mechanism to permanently encrypt the file if the user lost the file or it gets into the wrong hands. The system provides separate mechanisms to permanently encrypt the file</p> <p>4.5 Original and Modified File Management</p> <p>Modified and original file can be extracted from the server in case if the user loses the file.</p>
Function	User interface is one of the ways to interact with application. It packages all those screens, dialog boxes and forms that are visible to user. It provides user access to different options provided by the software.
Subordinates	This component has three subordinates; the Add files wizards, View files and the Add Users wizards.
Dependencies	Working of this component is dependent on integration of all other components.
Interfaces	N/A
Resources	Software: Java
Processing	User Interface displays wizards, notifications and messages to user passed by other components.
Data	Details of storage devices

4.11.2 File Manager

Identification	<i>Name:</i> File Manager <i>Location:</i> Server side of the system architecture
Type	Logical Component
Purpose	<p>Following functional requirements mentioned in SRS are fulfilled by this component:</p> <p>4.1 Partition File Encryption:</p> <p>This feature will allow the user to encrypt a complete non-executable file.</p> <p>Description: One of the basic functionalities of the system is to provide encryption for file given that it is a non-executable. When a user wants to encrypt a file with data, this feature option is used.</p> <p>4.2 File Container Encryption:</p> <p>This feature will allow the user to encrypt a file.</p> <p>Description: The second basic functionality of the system is to provide remote decryption for any already encrypted file. When a user wants to decrypt a file with data already stored in it, this feature option is used</p> <p>4.5 Original and Modified File Management</p> <p>Modified and original file can be extracted from the server in case if</p>

	the user loses the file.
Function	The function of this component is to upload the data to be encrypted/decrypted.
Subordinates	This component has two functions: one is to upload a non-executable file and other is to set the status as Available or Not Available.
Dependencies	This component is independent module.
Interfaces	Information will be send by UI.
Resources	Hardware: RAM, Processor. Software: Java libraries
Processing	This component first asks the user for the file which they want to secure i.e. type, size, location and name etc.

4.11.3 MAC Manager

Identification	<i>Name:</i> Mac Manager <i>Location:</i> Server side of the system architecture
Type	Logical Component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this component: 4.3 Location of File In order for the user to get the location of application generated files, user will go to already encrypted file section and resultantly will get the location of that file in the form of Mac and IP addresses.

Function	The function of this component is to locate the file through IP and MAC addresses when the user uploads the file
Subordinates	This component has two functions: one is the location of the file and second is MAC & IP management.
Dependencies	This component is dependent module.
Interfaces	N/A
Resources	Software: Java core libraries.
Processing	This component allow the user to locate any file they want from the list provided by the system i.e. type, size, location and name etc.

4.11.4 Key Management

Identification	<p><i>Name:</i> Key Management</p> <p><i>Location:</i> Server side of the system architecture</p>
Type	Logical Component
Purpose	<p>Following functional requirements mentioned in SRS are fulfilled by this component:</p> <p>4.5 Key Management:</p> <p>This feature manages the complete process of generation, authentication and storage of key</p> <p>Description: This module will be responsible to generate a 512-bit key for encryption and will store it on the CDB (Critical Data Block) called the Master Key.</p>
Function	The function of this component is to manage the encryption keys. to generate, store and authenticate the keys
Subordinates	This component has three subordinates: one is the generation of the key, second is the authentication of the key and the third is the storage of the key.
Dependencies	This component depends on the encryption/ decryption module.
Interfaces	N/A
Resources	Software: Java core libraries,
Processing	This component collects the data for key generation.
Data	Data collected for key generation.

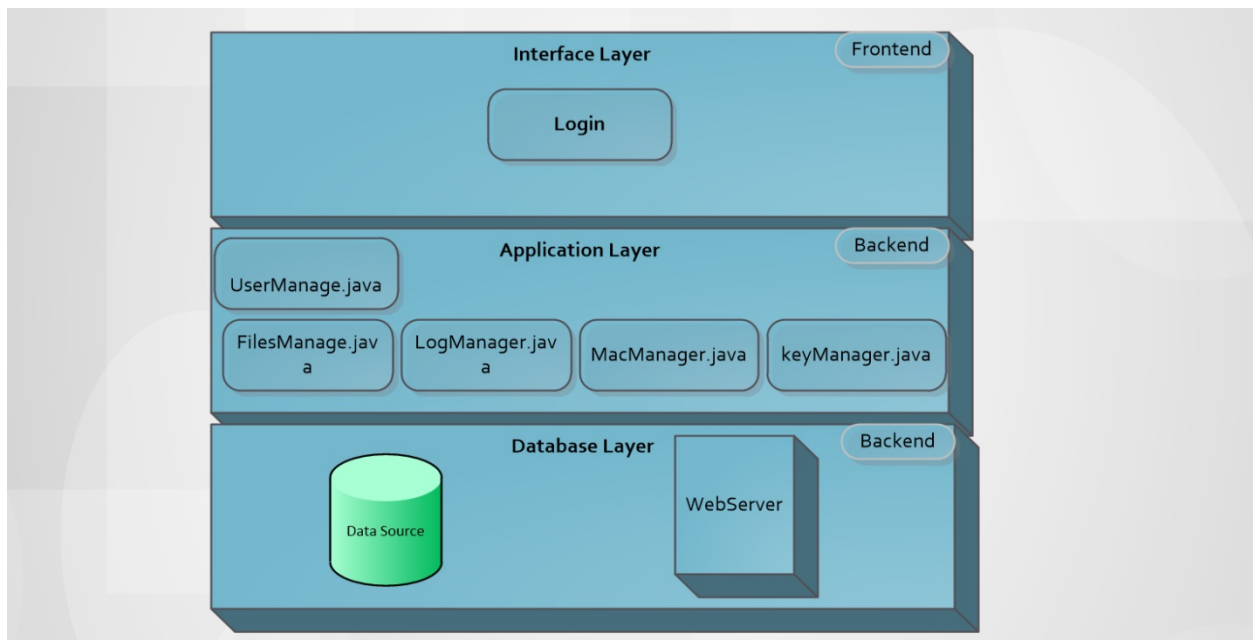
4.12 Reuse and Relationships to other products

Secure Digital File is not an extension of any other applications at any level but it is based on the techniques used by renowned disk encryption software and NIST provided guidelines for storage device encryption. It can be evolved into a bigger and more complex system with more features and functionality. Developers can also reuse the modules of the software. The practical usage of the system can be increased by adding more and more services to software like shredder, compression.

The primary motive of this application is to provide an application which is not proprietary and source code is available to re-use, maintain and extend in any way desired.

4.13 Design Decisions and tradeoffs

The functionality of Secure Digital File software basically includes interactive user interface with all of the logic controlled by the Server. User interacts with the interface to access the functionality of the software which is then serviced by the SDF class. This main driver class delegates the tasks further to sub-classes. Thus, the System Architecture of the software can be considered as **3- tier** as the output of each layer is input of above layer



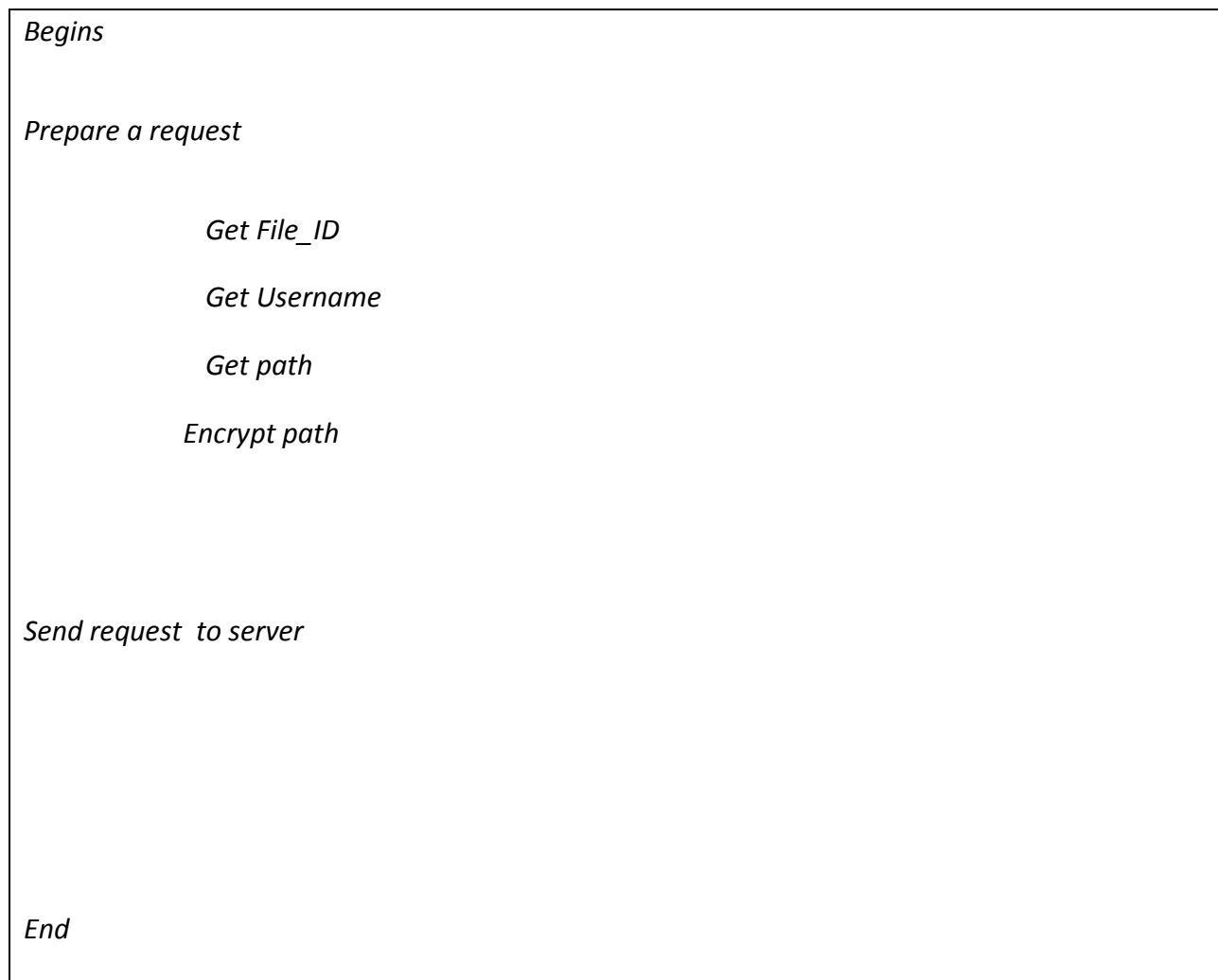
CHAPTER: 5

SYSTEM IMPLEMENTATION

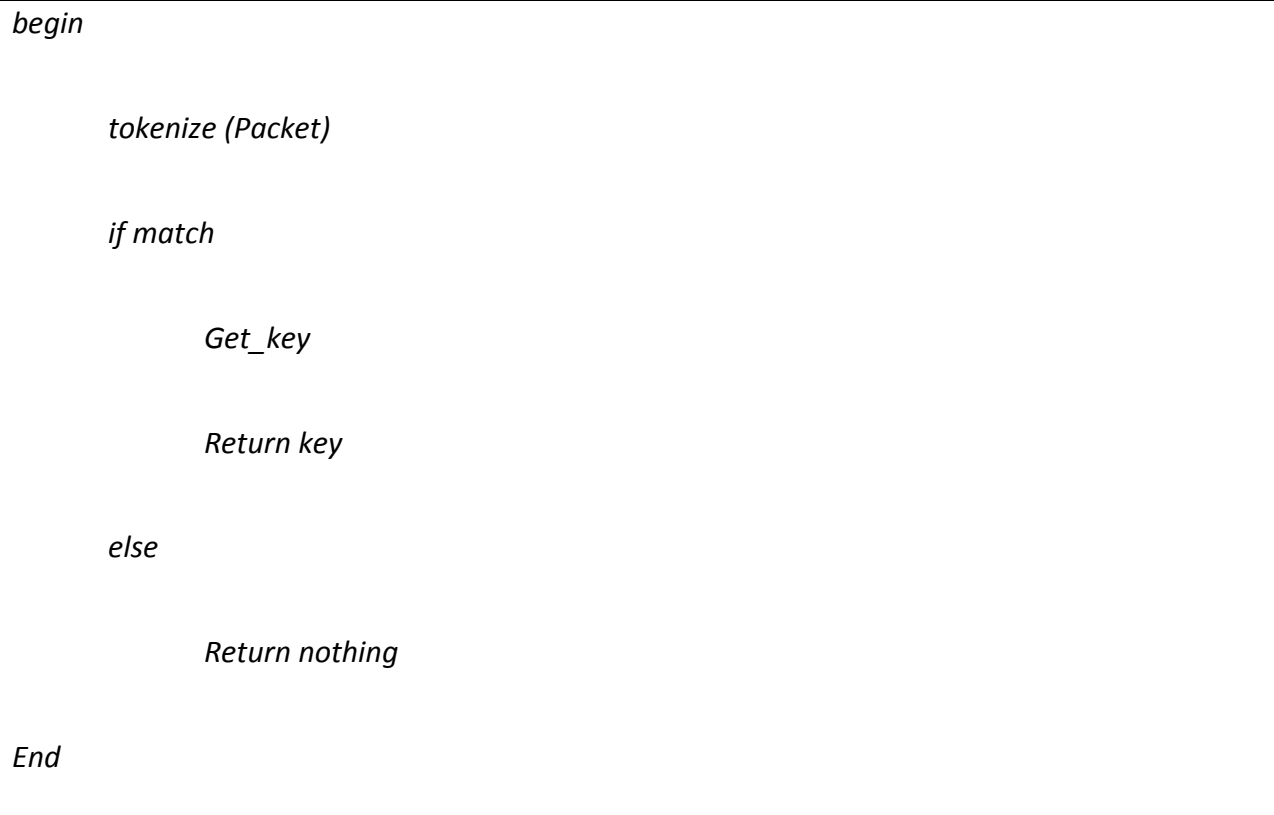
5 SYSTEM IMPLEMENTATION

The application works like when a user tries to access a file, generated by our application, the client application prepare a request. That request contains file ID, path, Token, And Motherboard number etc. then request is sent to the server. When a server receives a request packet it tokenizes it and compare the attributes with already stored attributes on the database. If it matches it send back a packet to user containing key to decrypt the file. But if does not match then key isn't provided.

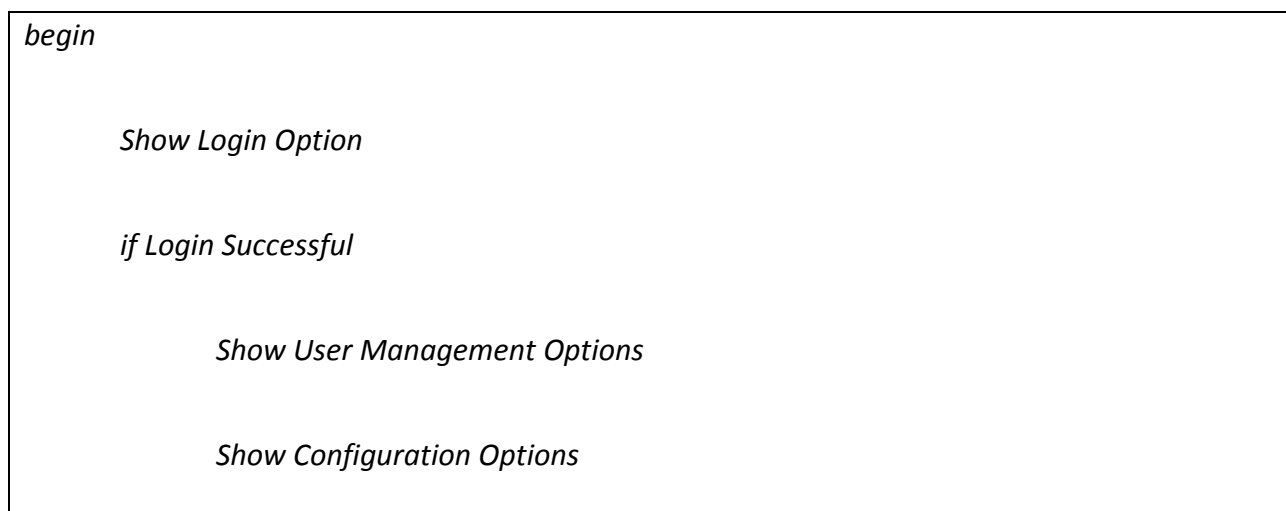
5.1.1 User Access a Modified File



5.1.2 Server receive request



5.1.3 For Settings Menu




```
    else

        returnToMain

End
```

5.1.4 For User Management

```
begin

    show User ID List

    if user selects addNewUser

        take ID & Pass and add new user

    if user selects deleteUser

        delete User from DataBase

    if user selects changePassword

        take password and save

End
```

CHAPTER: 6
ANALYSIS AND EVALUATION

6 Introduction

This test plan document describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of the "Secure Digital File" Disk Encryption Software. The test plan will ensure that Secure Digital File meets the customer requirements at an accredited level.

Manual Testing will be followed which includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Each Unit will be tested separately and then will be integrated with other units; therefore, Unit Testing and Integration testing will be followed. For each unit, Black box Testing is done and for combined units Acceptance Testing is done.

The test scope includes the Testing of all functional, application performance and use cases requirements listed in the *requirement document*.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure the testing of Secure Digital File. The pass/fail criteria of the test items are also defined. The document tracks the necessary information required to effectively define the approach to be used in the testing of the product.

6.1 Test Items

- Develop test cases.
- Execute tests based on the developed test cases for the software.
- Report defects from the executed test cases if any.
- Provide complete test report.
- Incorporate or manage changes later in the stage of the project development.

6.2 Features to be tested

Following Features are tested:

- Software will be able to create a new modified file for an uploaded file.
- Software will be able to store the file to view its access and attempt details.
- Software will be able to provide location of the file.
- Software will be able to Lock/Unlock the file against particular system.

6.3 Approach

Acceptance test will be executed based on this acceptance test plan. And after all test cases are executed, a test report will be summarized to show the quality of Secure Digital File. Following test approaches will be used in test execution:

- **Unit test.** Developers are responsible for unit testing. The implementation of each module and individual component will be verified separately.
- **Integration test.** After the unit test is passed above the defined quality threshold, testers will execute the integration test cases. After all the modules are integrated, it is crucial to test the product as a black-box.
- **Positive and negative testing design technique.** This approach will be combined with unit test and integration test. Test cases are designed in obvious scenarios, which ensure that all functional requirements are satisfied. What's more, different test cases will also be covered to show how the system reacts with invalid operations.

6.4 Item Pass/Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

- Preconditions are met
- Inputs are carried out as specified
- The result works as what specified in output => Pass
- The system doesn't work or not the same as output specification => Fail.

6.5 Suspension Criteria and Resumption Requirements

Any bugs found can be fixed by developers quickly and no need to start the testing process from the beginning. However, when major bugs will block some test cases as they are interdependent and the testing has to be paused.

6.6 Test Deliverables

Following are the Test Cases:

6.6.1 User Interface Testing:

Test Case Number	01
Test Case Name	To generate Modified File
Description	Testing Add file button on the left side menu bar.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Application should be opened.
Input Values	Click on "Add files" Button
Steps	<ul style="list-style-type: none">• Open the application• Main Screen is open.• Select the 'Add file' button.• A new screen will appears that will ask to upload the file.• Fill the other required attributes.
Expected output	Modified file generates.
Actual output	Modified file generates.
Status	Test case passed successfully.

Test Case Number	02
Test Case Name	Access Modified File
Description	Testing whether a user can access the above generated file while setting File Status to "Available".
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Click on "Generated Modified File"
Steps	<ul style="list-style-type: none">• Open the File.
Expected output	File will be opened
Actual output	File Opened.
Status	Test case passed successfully.

Test Case Number	03
Test Case Name	View All Files
Description	Testing View All Files button on the left side menu bar.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Click on " View All Files " Button
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button.
Expected output	Screen with different file names along with their attr. appears.
Actual output	Screen with different file names along with their attr. appears.
Status	Test case passed successfully.

Test Case Number	04
Test Case Name	User will receive an Email.
Description	User, who has uploaded its file, will receive an email alert that someone has tried to access your file.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Check Your mail.
Steps	<ul style="list-style-type: none"> • Check your mail.
Expected output	User will receive an email.
Actual output	User received an email alert.
Status	Test case passed successfully.

Test Case Number	05
Test Case Name	Access Modified File
Description	Testing whether a user can access the above generated file while setting File Status to "NOT Available".
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Open "Generated Modified File"
Steps	<ul style="list-style-type: none"> • Open the File.
Expected output	File will not be opened.
Actual output	File NOT Opened.
Status	Test case passed successfully.

Test Case Number	06
Test Case Name	Attempt Details of file
Description	Testing Attempt Details button on the top right side menu bar.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Click on " Attempt Details " Button
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button. • Select the 'Access Details button. • Select the 'Attempt Details button.
Expected output	Screen with the name of a person appears who have tried to access the file as many times as shown.
Actual output	Screen with the name of a person appears who have tried to access the file as many times as shown.
Status	Test case passed successfully.

Test Case Number	07
Test Case Name	See the Location of people who have accessed.
Description	Location can accessed either through in the form of MAC/IP addresses or Access Location button.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Click on "Attempt Details "Button.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button. • Select the 'Access Details button. • Select the 'Attempt Details button, MAC/IP addresses appears. Or • Select the "Location" Button.
Expected output	Screen with the MAC and IP addresses appears of the systems who have tried to access the file as many times as shown.
Actual output	Screen with the MAC and IP addresses appears of the systems who have tried to access the file as many times as shown.
Status	Test case passed successfully.

Test Case Number	08
Test Case Name	To Lock/Unlock File for a person
Description	Testing the Locking/Unlocking the file functionality.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Modified file should be generated.
Input Values	Click on "Permission" checkbox.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button. • Select the 'Access Details button. • Click on "Permission" checkbox if you want to unlock file that particular person, otherwise uncheck it.
Expected output	Screen with display message appears showing that particular person is allowed/not allowed to view the file.
Actual output	Screen with display message appears showing that particular person is allowed/not allowed to view the file.
Status	Test case passed successfully.

Test Case Number	09
Test Case Name	Add to Allowed/ Disallowed list
Description	<p>Maintaining a list of Users that will be only allowed/ Disallowed for a particular file instead of doing for each.</p> <p>In allowed case, all people will be not allowed except those in list, whereas in disallowed case, all people will be allowed except those in list.</p>
Testing Technique	Component testing, Black Box Testing
Preconditions	Modified file should be generated. You have to select Allowed/Disallowed option over there.
Input Values	Click on "Add to Allowed" button.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button. • Click on "Add to Allowed/Disallowed" button.
Expected output	Success Message appears.
Actual output	Success Message appears.
Status	Test case passed successfully.

Test Case Number	10
Test Case Name	Import CSV file.
Description	Directly import CSV file and set their permission to allow instead of adding each person to Add to allowed list. Suitable for an Organization.
Testing Technique	Component testing, Black Box Testing
Preconditions	Modified file should be generated. You have to upload CSV file.
Input Values	Click on "Upload CSV" button.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button. • Click on "Upload CSV" button.
Expected output	Success Message appears.
Actual output	Success Message appears.
Status	Test case passed successfully.

Test Case Number	11
Test Case Name	Download Original/Modified file from server
Description	Downloading Original/Modified file from server but it's up to the user where he wants to place the file on server or not.
Testing Technique	Component testing, Black Box Testing
Preconditions	Modified file should be generated. You have to check the upload Original on server checkbox incase if you want it to download later otherwise uncheck it.
Input Values	Click "Download" icon.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the 'View All Files' button.
Expected output	Success Message appears.
Actual output	Success Message appears.
Status	Test case passed successfully.

Test Case Number	12
Test Case Name	Add User
Description	Testing “Add User” functionality. User will request the admin and admin will add him if he is legitimate.
Testing Technique	Component testing, Black Box Testing
Preconditions	Application should be opened.
Input Values	Click “Add User” button.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the ‘Add User button.
Expected output	Success Message appears.
Actual output	Success Message appears.
Status	Test case passed successfully.

Test Case Number	13
Test Case Name	Remove/Update User Profile
Description	Testing “Remove/Update User Profile” functionality.
Testing Technique	Component testing, Black Box Testing
Preconditions	User should be added whose profile is going to update.
Input Values	Click “Action” icon next to that particular user.
Steps	<ul style="list-style-type: none"> • Open the application • Main Screen is open. • Select the “View All User” button. • Click “Action” icon next to that particular user.
Expected output	Screen with multiple options appears; update it and Success Message would appear.
Actual output	Screen with multiple options appears; update it and Success Message would appear.
Status	Test case passed successfully.

This Feature is only for Organizational

Test Case Number	14
Test Case Name	Add Employee
Description	Testing “Add Employee” button. Either Employee can be added manually or organization CSV file can be imported.
Testing Technique	Component testing, Black Box Testing
Preconditions	Application should be opened.
Input Values	Click “Add Employee” button
Steps	<ul style="list-style-type: none">• Open the application• Main Screen is open.• Select the “Add Employee” button.• Then Click” Add Employee from CSV file” if you don’t it manually.
Expected output	Screen with multiple options appears; update it and Success Message would appear.
Actual output	Screen with multiple options appears; update it and Success Message would appear.
Status	Test case passed successfully.

Test Case Number	15
Test Case Name	Remove/Update Employee Profile
Description	Testing “Remove/Update Employee Profile” functionality.
Testing Technique	Component testing, Black Box Testing
Preconditions	Employee should be added whose profile is going to update.
Input Values	Click “Action” icon next to that particular Employee.
Steps	<ul style="list-style-type: none">• Open the application• Main Screen is open.• Select the “View All Employee r” button.• Click “Action” icon next to that particular Employee.
Expected output	Screen with multiple options appears; update it and Success Message would appear.
Actual output	Screen with multiple options appears; update it and Success Message would appear.
Status	Test case passed successfully.

CHAPTER: 7
FUTURE WORK

7 FUTURE WORK

Any software of this kind always needs more and more work to evolve. There are a lot of possible changes and additions that can be done to the system to improve its performance and functionalities. The system has been made in a modular fashion which enables integrating new features very easy.

Furthermore, the practical usage of the system can be increased by adding more and more services to software.

CHAPTER: 8
CONCLUSION

8 CONCLUSION

7.1 Overview

The Secure Digital File is aimed for public and private sector organizations. The main purpose of this project is the development of an application that would allow the users to give you remote access control (Read/Write/Execute) of files via the network.

It can be utilized in military institutions and universities. Especially for data control and security. The software will provide desktop application with user friendly Interface. Aim &

Objectives

CHAPTER:9
BIBLIOGRAPHY

9 BIBLIOGRAPHY

1. Abdullah Z.H., Udzir N.I., Mahmud R., Samsudin K. (2011) *File Integrity Monitor Scheduling Based on File Security Level Classification*.
2. Zain J.M., Wan Mohd W.M., El-Qawasmeh E. (eds) Software Engineering and Computer Systems. ICSECS 2011. *Communications in Computer and Information Science*, vol 180. Springer, Berlin, Heidelberg
3. Dworkin, M., 2010. *Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices*, s.l.: NIST Special Publication 800-38E
4. Scarfone, K., Spuppaya, M. & Sexton, M., 2007. *Guide to Storage Encryption Technologies for End User Devices*, s.l.: NIST Special Publication 800-111.
5. NIST, 2001. *Announcing the Advanced Encryption Standard (AES)*, s.l.: Federal Information Processing Standards- 197.
6. NIST, 2001. *Announcing the Advanced Encryption Standard (AES)*, s.l.: Federal Information Processing Standards- 197.

APPENDIX A

USER MANUAL

Table of Contents

APPENDIX A	81
USER MANUAL	81
1 GENERAL INFORMATION.....	84
1.1 System Overview:	85
1.2 Organization of the manual:.....	85
2 SYSTEM SUMMARY	86
2.1 System Configuration:.....	87
2.2 User Access Levels:	87
2.3 Contingencies:	87
3 GETTING STARTED	88
3.1 Installation:.....	89
4 USING THE SYSTEM:	90
4.1 Introduction Wizard	91
4.2 Ceate Modified file.....	91
4.3 Locate The File.....	92
4.4 File Reports	92
4.5 Define Access Control for file	92

4.6 User Management 92

Table of Figures:

Figure 3-1 Main Screen 91

General Information

1 GENERAL INFORMATION

This section explains in general terms the system **Secure Digital File** and the purpose for which it is intended.

1.1 System Overview:

Secure Digital File is an XTS-AES based disk encryption solution. This allows creating encrypted file container and encrypted non-system partitions. It provides confidentiality and protects static data. It's a GUI based solution which can be deployed on computers with OS Windows 7/8/8.1/10.

1.2 Organization of the manual:

The user's manual consists of five sections: General Information, System Summary, Getting Started, Using the System.

1. **General Information** section explains in general terms the system and the purpose for which it is intended.
2. **System Summary** section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.
3. **Getting Started** section explains how to setup the system and configure it for the first time. The section presents briefly system's settings.
4. **Using the System** section provides a detailed description of system functions.

2.0 System Summary

2 SYSTEM SUMMARY

System Summary section provides a general overview of the system. The summary outlines the uses of the system's information and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

2.1 System Configuration:

Secure Digital File does not need special requirements. It can work on Windows 8/8.1/10.

System does need network connection.

2.2 User Access Levels:

The System will be available to user within their defined privileges.

2.3 Contingencies:

Schedule Risk:

The project might get behind schedule so in order to complete the project in time we will need to increase the hours/day that the project is being worked on.

Operational Risks:

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

Technical risks:

Technical risks will be eliminated by keeping the once defined requirements constant.

Programmatic Risks:

In case of a programmatic risk the scope of the project will be limited in order to stay inside the constraints of the project.

3.0 Getting Started

3 GETTING STARTED

This section explains how to install application and information regarding Secure Digital File.

3.1 Installation:

Secure Digital File application can be installed easily in few steps.

Then run Secure Digital File.exe to install it.

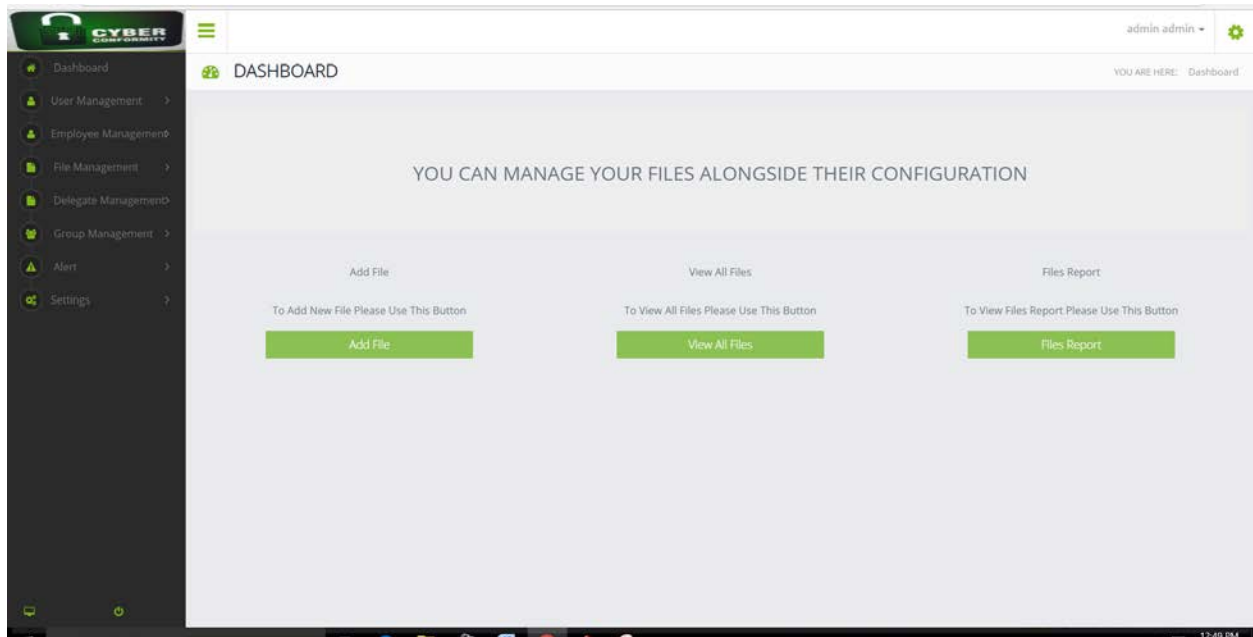


Figure 3-1 Main Screen

4.0 Using the System

4 USING THE SYSTEM:

This section provides a description of system functions and features.

4.1 Introduction Wizard

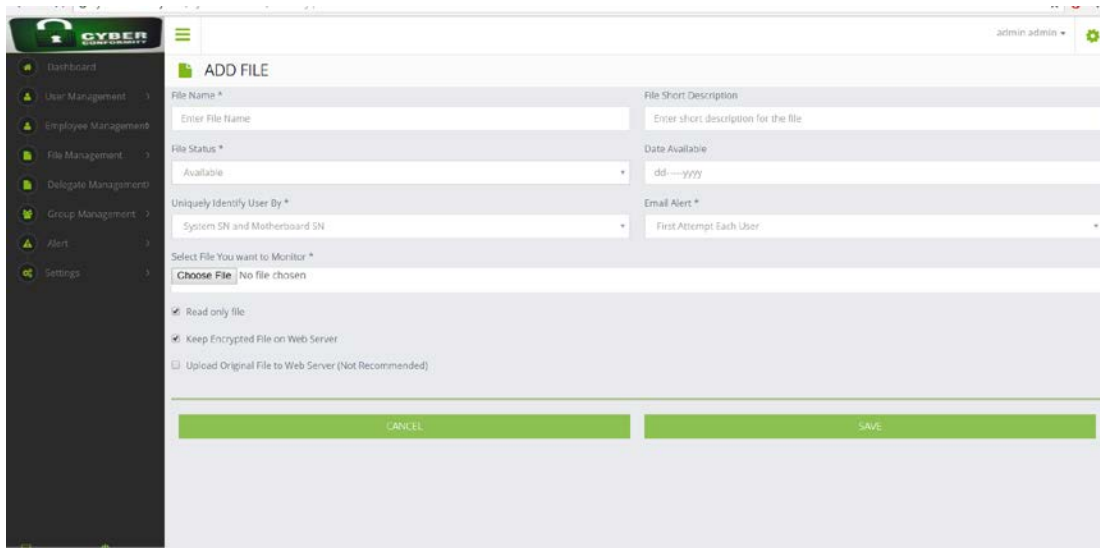
This gives user options to navigate through the system and use its different functions. Includes following options to proceed.

- Create Modified File
- Locate it
- File Reports
- Define Access Control for a File.
- User Management

The user will click on a button of his choice which displays new wizard which will have steps to proceed further in Secure Digital File.

4.1.1 Create Modified File

New wizard opens when Add file button is clicked. This provides a modified file.



The screenshot shows the 'ADD FILE' wizard interface. On the left is a dark sidebar with a 'CYBER' logo and a menu containing: Dashboard, User Management, Employee Management, File Management, Delegate Management, Group Management, Alert, and Settings. The main content area is titled 'ADD FILE' and contains the following fields and options:

- File Name ***: A text input field with the placeholder 'Enter File Name'.
- File Short Description**: A text input field with the placeholder 'Enter short description for the file'.
- File Status ***: A dropdown menu with 'Available' selected.
- Date Available**: A date input field with the placeholder 'dd----yyyy'.
- Uniquely Identify User By ***: A dropdown menu with 'System SN and Motherboard SN' selected.
- Email Alert ***: A dropdown menu with 'First Attempt Each User' selected.
- Select File You want to Monitor ***: A file selection area with a 'Choose File' button and the text 'No file chosen'.
- Read only file**: A checked checkbox.
- Keep Encrypted File on Web Server**: A checked checkbox.
- Upload Original File to Web Server (Not Recommended)**: An unchecked checkbox.

At the bottom of the form are two large green buttons: 'CANCEL' and 'SAVE'.

4.1.2 **Locate the file**

- New wizard opens when File Management button is clicked.
- File Management -> View Files-> File Access log

4.1.3 **File Reports**

- New wizard opens when File Management button is clicked.
- File Management -> File Reports

4.1.4 **Define Access control for a file**

New wizard opens when File Management button is clicked.

- Either you can define control access while adding a file ,
- Or You can update control access to the already added files

File Management -> View Files-> Click on Edit icon in Action column

4.1.5 **User Management**

New wizard opens when user Management button is clicked. This provides two options

- Add user
- View all user