

A CLOUD SECURE STORAGE PROTECTION BASED ON DATA ENCRYPTION, DECRYPTION AND DISPERSION

A PROJECT REPORT

Submitted by

DINESH A (910618104026)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



K.L.N. COLLEGE OF ENGINEERING, POTTAPALAYAM
(An Autonomous Institution, Affiliated to Anna University, Chennai)

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**A CLOUD SECURE STORAGE PROTECTION BASED ON DATA ENCRYPTION, DECRYPTION AND DISPERSION**” is the bonafide work of “**DINESH A (910619104026)**” who carried out the project work under my supervision.

SIGNATURE

**Dr. S. MIRUNA JOE AMALI,
HEAD OF THE DEPARTMENT**

Computer science and engineering
K.L.N. College of Engineering,
Pottapalayam,
Sivagangai-630 612.

SIGNATURE

**Mrs. S. GAYATHRI
SUPERVISOR
ASSISTANT PROFESSOR**
Computer science and engineering
K.L.N. College of Engineering,
Pottapalayam,
Sivagangai-630 612.

Submitted for the project viva-voce conducted on _____.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete, without mentioning the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We sincerely extend our gratitude to the founder of our institution **Late. Thiru. K.L.N. KRISHNAN** for making us march towards the glory of success.

We express our sincere thanks to our respected Principal **Dr.A.V.RAM PRASAD, M.E, Ph.D., MISTE, FIE**, for all the facilities offered.

We would like to express our deep gratitude and heartfelt thanks to **Dr. S. MIRUNA JOE AMALI, M.E, Ph.D**, Head of the Department of Computer Science and Engineering who motivated and encouraged us to do this outtrival project for this academic year.

Our profound, delightful and sincere thanks to our project guide **Mrs. S. GAYATHRI, M.E, ASSOCIATE PROFESSOR** and project coordinator **Mr. SULTHAN ALI KHAN, M.E** whose support was inevitable during the entire period of our work.

We thank our teaching staff for sharing their knowledge and view to enhance our project. We also thank our non-teaching staff for extending their technical support to us. We thank our parents for giving us such a wonderful life and our friends for their friendly encouragement throughout the project.

Finally, we thank the almighty for giving the full health to finish the project successfully.

ABSTRACT

Cloud storage service has shown its great power and wide popularity which provides fundamental support for rapid development of cloud computing. However, due to management negligence and malicious attack, there still lie enormous security incidents that lead to quantities of sensitive data leakage at cloud storage layer. From the perspective of protecting cloud data confidentiality, this work proposed a two schemes namely encrypted data storage and data dispersion. The encrypted data storage processed an DL based cryptoanalysis algorithm and data partition is performed by using a divide and conquer approach. The main purpose of this work is to provide idea of the combination of these two algorithms to provide double security to the data stored inside the cloud. Also it protects the data from an unauthorized users. The experimental results indicate that proposed mechanism is not only suitable for ensuring the data security at storage layer but also can store huge amount of cloud data effectively in a minimum time overhead.

TABLE OF CONTENTS

| CHAP TER NO. | TITLE | PAGE NO. |
|--------------------|--|-------------|
| | ABSTRACT | iv |
| | ABBREVIATIONS | viii |
| | LIST OF FIGURES | ix |
| | LIST OF SYMBOLS | x |
| 1 | INTRODUCTION | 1 |
| 1.1 | Cloud Computing | 1 |
| 1.2. | Deployment Models | 2 |
| 1.2.1 | Private Cloud | 2 |
| 1.2.2 | Public Cloud | 3 |
| 1.2.3 | Hybrid Cloud | 3 |
| 1.2.4 | Community Cloud | 4 |
| 1.2.5 | Distributed Cloud | 4 |
| 1.2.6 | Inter Cloud | 4 |
| 1.2.7 | Multi Cloud | 4 |
| 2 | LITERATURE SURVEY | 5 |
| 2.1 | Forward Secure Public Key Encryption with Keyword Search for Outsourced Cloud Storage | 5 |
| 2.2 | CAFE: A Virtualization-Based Approach to Protecting Sensitive Cloud Application Logic Confidentiality | 5 |
| 2.3 | Provably Secure and Lightweight Identity- Based Authenticated Data Sharing Protocol for Cyber-Physical Cloud Environment | 6 |

| | | |
|----------|---|-----------|
| 2.4 | Data Integrity Auditing without Private Key Storage for Secure Cloud Storage | 6 |
| 2.5 | Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing | 6 |
| 2.6 | Existing System | 7 |
| 2.6.1 | Disadvantages | 8 |
| 2.7 | Proposed System | 8 |
| 2.7.1 | Advantages | 8 |
| 3 | SYSTEM REQUIREMENTS | 9 |
| 3.1 | Introduction | 9 |
| 3.2 | Hardware and Software Requirements | 9 |
| 3.2.1 | Hardware Requirements | 9 |
| 3.2.2 | Software Requirements | 9 |
| 3.3 | Hardware and Software Specification | 10 |
| 3.3.1 | Software Specification | 10 |
| 3.4 | Software Description | 10 |
| 3.4.1 | Cloud Me Tool | 10 |
| 3.4.2 | Java Technology | 11 |
| | 3.4.2.1 Java Programming Language | 11 |
| 4 | SYSTEM DESIGN | 15 |
| 4.1 | System Architecture | 15 |
| 4.2 | Use case Diagram | 16 |
| 4.3 | Activity Diagram | 17 |
| 4.4 | Sequence Diagram | 18 |
| 5 | SYSTEM IMPLEMENTATION | 19 |
| 5.1 | List of Modules | 20 |
| 5.2 | Module Description | 20 |
| 5.2.1 | Data Owner | 20 |

| | | |
|-----------|---------------------------|-----------|
| 5.2.2 | User Module | 20 |
| 5.2.3 | Admin | 20 |
| 5.2.4 | Data Splitting | 21 |
| 5.2.5 | Encryption Module | 23 |
| 6 | SAMPLE CODE | 26 |
| 7 | SYSTEM TESTING | 28 |
| 7.1 | Unit Testing | 29 |
| 7.2 | Integration Testing | 29 |
| 7.3 | Validation Testing | 29 |
| 7.4 | Verification Testing | 30 |
| 8 | SCREENSHOTS | 31 |
| 9 | CONCLUSION | 35 |
| 10 | FUTURE ENHANCEMENT | 36 |
| 11 | REFERENCES | 37 |

ABBREVIATIONS

| | |
|--------|---|
| AI | Artificial Intelligence |
| CSP | Cloud Service Provider |
| BLM | Block Awareness Language Model |
| CSSM | Cloud Secure Storage Mechanism |
| DL | Deep Learning |
| HTML | Hyper Text Markup Language |
| SQL | Structured Query Language |
| DNN | Deep Neural Network |
| TCP/IP | Transmission Control Protocol / Internet Protocol |

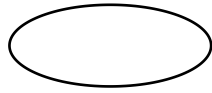
LIST OF FIGURES

| FIGURE NO. | TITLE | PAGE NO. |
|-------------------|-----------------------|-----------------|
| 1.1 | Cloud Computing Types | 2 |
| 2.1 | CSSM Architecture | 7 |
| 4.1 | System Architecture | 15 |
| 4.2 | Use case Diagram | 16 |
| 4.3 | Activity Diagram | 17 |
| 4.4 | Sequence Diagram | 18 |
| 8.1 | Registration | 31 |
| 8.2 | User Login | 31 |
| 8.3 | File Upload | 32 |
| 8.4 | Uploaded Files | 32 |
| 8.5 | Cloud Data Partition | 33 |
| 8.6 | Data Encryption | 33 |
| 8.7 | Data Request | 34 |
| 8.8 | Download | 34 |

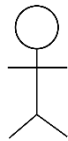
LIST OF SYMBOLS

SYMBOL

SYMBOL NAME



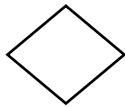
Use Case



Actor



Control flow



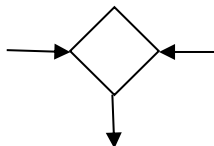
Decision



Start



Stop



Merge

CHAPTER 1

INTRODUCTION

1.1 CLOUD COMPUTING

In the evolution of computing technology, information science has moved from mainframes to private computers to server-centric computing to the online. Today, several organizations are seriously considering adopting cloud computing, consecutive major milestone in technology and business collaboration. Cloud computing has been outlined by NIST (National Informatics Science and Technology) as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that maybe quickly provisioned and discharged with lowest management effort or cloud provider interaction. Cloud Computing remains a work in progress. Although cloud computing 's edges are tremendous, security and privacy issues are the first obstacles to wide adoption. As a result of cloud service providers (CSPs) are separate administrative entities, moving to the business public cloud deprives users of direct management over the systems that manage their data and applications.

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers that may be located far from the user—ranging in distance from across a city to across the world.

1.2 DEPLOYMENT MODELS

1.2.1 Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management,[88] essentially "[lacking] the economic model that makes cloud computing such an intriguing concept.

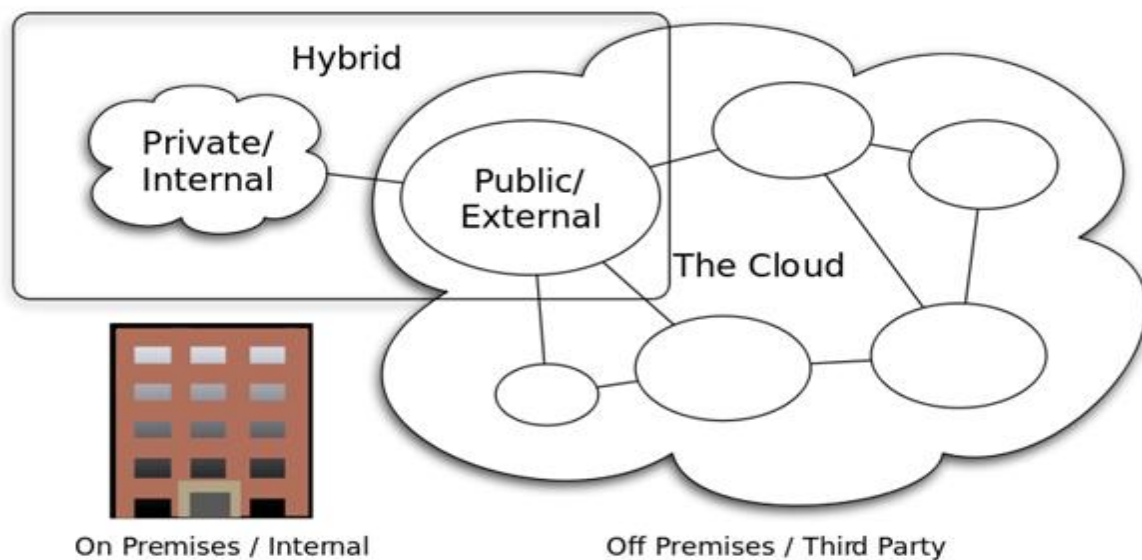


Fig 1.1:Cloud Computing Types

1.2.2 Public cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free. Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon Web Services (AWS), Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and "Azure ExpressRoute" respectively.

1.2.3 Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service. Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service.

1.2.4 Community cloud

Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party, and either hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.

1.2.5 Distributed cloud

A cloud computing platform can be assembled from a distributed set of machines in different locations, connected to a single network or hub service. It is possible to distinguish between two types of distributed clouds: public-resource computing and volunteer cloud. Public-resource computing. This type of distributed cloud results from an expansive definition of cloud computing, because they are more akin to distributed computing than cloud computing.

1.2.6 Intercloud

The Intercloud is an interconnected global "cloud of clouds" and an extension of the Internet "network of networks" on which it is based. The focus is on direct interoperability between public cloud service providers, more so than between providers and consumers (as is the case for hybrid- and multi-cloud).

1.2.7 Multicloud

Multicloud is the use of multiple cloud computing services in a single heterogeneous architecture to reduce reliance on single vendors, increase flexibility through choice, mitigate against disasters, etc. It differs from hybrid cloud in that it refers to multiple cloud services, rather than multiple deployment modes (public, private, legacy).

CHAPTER-2

LITERATURE SURVEY

2.1 Forward Secure Public Key Encryption with Keyword Search for Outsourced Cloud Storage

M. Zeng et al developed public key encryption with keyword search (PKSE) is considered to be a promising technique, since clients can efficiently search over encrypted data files. That is, a client first generates a search token when to query data files, the cloud server uses the search token to proceed the query over encrypted data files. However, a serious attack is raised when PKSE meets cloud. Formally speaking, the cloud server can learn the information of a newly added encrypted data file containing the keyword that previously queried by using the search tokens it has received, and can further discover the privacy information. To address this issue, we propose a forward secure public key searchable encryption scheme, in which a cloud server cannot learn any information about a newly added encrypted data file containing the keyword that previously queried.

2.2 CAFE: A Virtualization-Based Approach to Protecting Sensitive Cloud Application Logic Confidentiality

Park et al presented a system named CAFE for cloud infrastructures where sensitive software logic can be executed with high secrecy protected from any piracy or reverse engineering attempts in a virtual machine even when its operating system kernel is compromised. The key mechanism is the end-to-end framework for the execution of applications, which consists of the secure encryption and distribution of confidential application binary files, and the runtime techniques to load, decrypt, and protect the program logic by isolating them from tenant virtual machines based on hypervisor-level techniques. We evaluate applications in several software categories which are commonly offered in cloud marketplaces showing that strong confidential execution can be provided with only marginal changes (around 100-220 lines of code) and

minimal performance overhead. The results demonstrate the effectiveness and practicality of CAFE in cloud marketplaces.

2.3 Provably Secure and Lightweight Identity-Based Authenticated Data Sharing Protocol for Cyber-Physical Cloud Environment.

Karati et al presented a lightweight identity-based authenticated data sharing protocol to provide secure data sharing among geographically dispersed physical devices and clients. The proposed protocol is demonstrated to resist chosen-ciphertext attack (CCA) under the hardness assumption of decisional-Strong Diffie-Hellman (SDH) problem. We also evaluate the performance of the proposed protocol with existing data sharing protocols in terms of computational overhead, communication overhead, and response time.

2.4 Data Integrity Auditing without Private Key Storage for Secure Cloud Storage

W. Shen et al explored a new paradigm called data integrity auditing without private key storage and design such a scheme. In this scheme, we use biometric data (e.g., iris scan, fingerprint) as the user's fuzzy private key to avoid using the hardware token. Meanwhile, the scheme can still effectively complete the data integrity auditing. We utilize a linear sketch with coding and error correction processes to confirm the identity of the user. In addition, we design a new signature scheme which not only supports blockless verifiability, but also is compatible with the linear sketch. The security proof and the performance analysis show that our proposed scheme achieves desirable security and efficiency.

2.5 Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing

J. Shen et al developed novel block design-based key agreement protocol that supports multiple participants, which can flexibly extend the number of participants in a cloud environment according to the structure of the block design. Based on the proposed group data sharing model, we present general formulas for generating the common conference key IC for multiple participants. Note that by benefiting from the $(v, k + 1, 1)$ -block design, the computational complexity of the proposed protocol linearly increases.

2.6 EXISTING SYSTEM

A Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption

H. Song et al presented a Cloud Secure Storage Mechanism named CSSM. To avoid data breach at the storage layer, CSSM integrated data dispersion and distributed storage to realize encrypted, chunked and distributed storage. In addition, CSSM adopted a hierarchical management approach and combined user password with secret sharing to prevent cryptographic materials leakage. The experimental results indicate that proposed mechanism is not only suitable for ensuring the data security at storage layer from leakage, but also can store huge amount of cloud data effectively without imposing too much time.

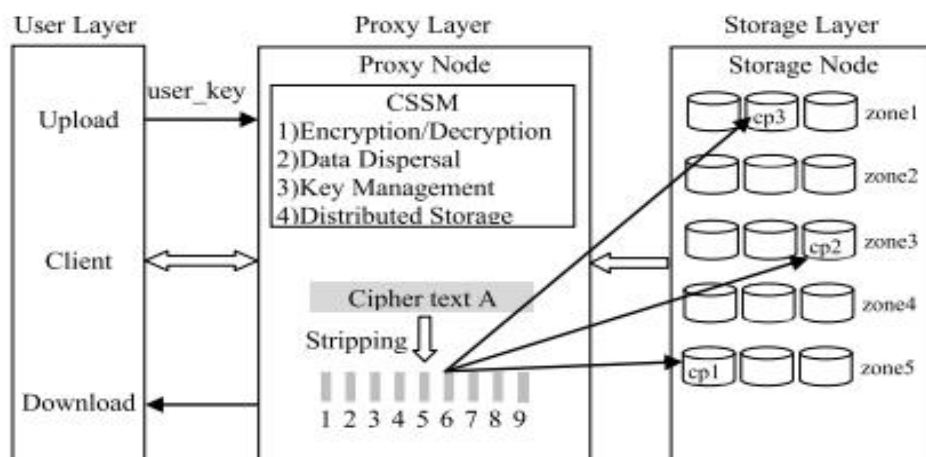


Fig 2.1: CSSM Architecture

2.6.1 DISADVANTAGES

- Lead unauthorized access to user data at storage layer.
- Requires skilled users to manage the edge servers.
- Expensive cost.
- Requires more time for processing.
- Inefficient Key management.

2.7 PROPOSED SYSTEM

Cloud storage service has shown its great power and wide popularity which provides fundamental support for rapid development of cloud computing. However, due to management negligence and malicious attack, there still lie enormous security incidents that lead to quantities of sensitive data leakage at cloud storage layer. From the perspective of protecting cloud data confidentiality, this work proposed a two schemes namely encrypted data storage and data partition. The encrypted data storage processed an DL based cryptoanalysis algorithm and partition is performed by using a divide and conquer approach. The main purpose of this work is to provide idea of the combination of these two algorithms to provide double security to the data stored inside the cloud.

2.7.1 ADVANTAGES

- Prevention of data leakage.
- Increased the attack difficulty.
- High protection in key management.
- Minimum time and cost overhead

CHAPTER-3

SYSTEM REQUIREMENT

3.1 INTRODUCTION

The system requirement is the first step in the requirements analysis process. It lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 SOFTWARE REQUIREMENTS

- Operating system : Windows XP/7.
- Coding Language : JAVA
- Database : MYSQL
- Tool used : CloudMe Tool

3.2.2 HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

3.3 HARDWARE AND SOFTWARE SPECIFICATION

3.3.1 SOFTWARE SPECIFICATION

- OS: window 7
- Software tool: Microsoft Visual Studio 2010
- Back end: SQL-Server 2005 and XML.
- Front end: net
- Documentation: MS- office.

3.4 SOFTWARE DESCRIPTION

3.4.1 CLOUDME TOOL

CloudMe is a file storage service operated by CloudMe AB that offers cloud storage, file synchronization and client software. It features a blue folder that appears on all devices with the same content, all files are synchronized between devices. The CloudMe service is offered with a freemium business model and provides encrypted SSL connection with SSL Extended Validation Certificate. CloudMe provides client software for Microsoft Windows, macOS, Linux, Android, iOS, Google TV, Samsung Smart TV, WD TV, Windows Storage Server for NAS and web browsers. As a cloud sync storage provider, CloudMe has a strong focus on the European market and differentiates itself from other storage providers with mobility and media features like Samsung SmartTV support.

Recently Novell announced support for the CloudMe service in their Dynamic File Services Suite. Novosoft Handy Backup version 7.3 also announced support for CloudMe. WinZip is also integrated with CloudMe. There are many third party mobile apps and software available for CloudMe, many using the WebDAV support of CloudMe. CloudMe features a Cloud storage and sync solution that allows the users to store, access and share their content, both with each other and with people outside the service. Sharing can be done by email, text messaging, Facebook and Google. Files can

be stored in a blue folder, which is synchronized to all connected computers and devices. A web desktop and cloud OS service called CloudTop.com is available that uses CloudMe as its internet file system.

CloudMe is a powerful & stylish multipurpose Cloud Storage & File-Sharing Services WordPress theme. It has a modern business design created for online presentation of cloud storage and file-sharing services. It's a fresh theme that's also perfect for all sorts of applications, web hosting business or any kind of technology websites. It includes custom post types that you can use for attracting clients, displaying the range of your services, team members, subscription plans and pricing, and many more.

CloudMe is fully responsive and 100% Retina Ready which makes the theme look splendid on any device.

- 4 beautiful premade homepages.
- Ready to use pages: About, Benefits, Resources, Pricing, Team, Services.
- Advanced Contact Forms.
- Outstanding short codes.

3.4.2 JAVA TECHNOLOGY

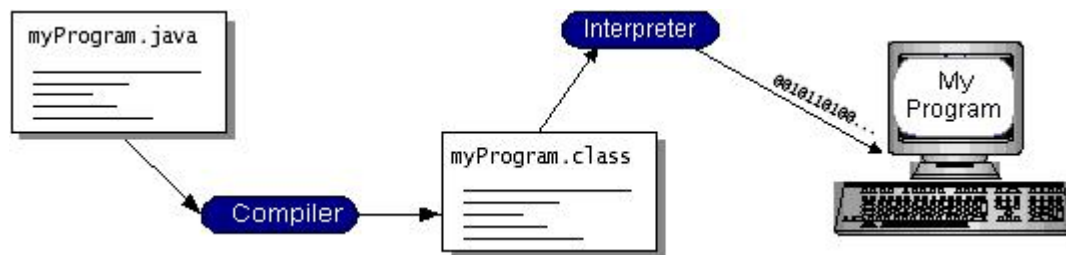
Java technology is both a programming language and a platform.

3.4.2.1 The Java Programming Language

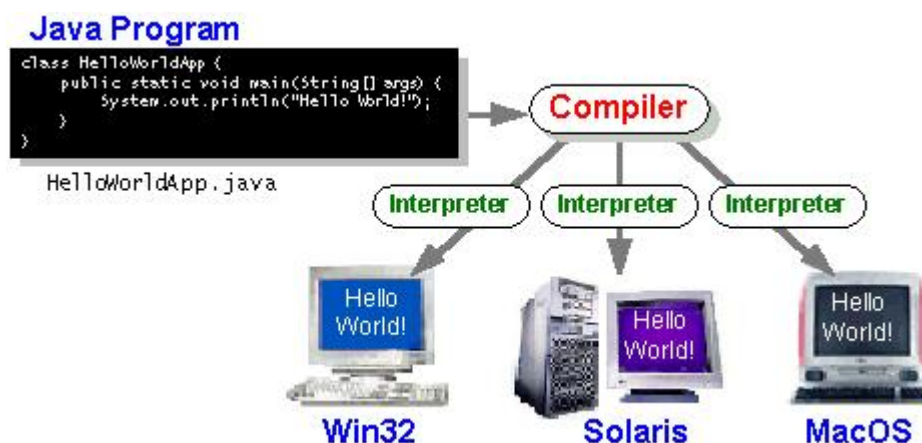
The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on

the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

- **The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets:** The set of conventions used by applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

- **Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components:** Known as JavaBeans™, can plug into existing component architectures.
- **Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™):** Provides uniform access to a wide range of relational databases.
- The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

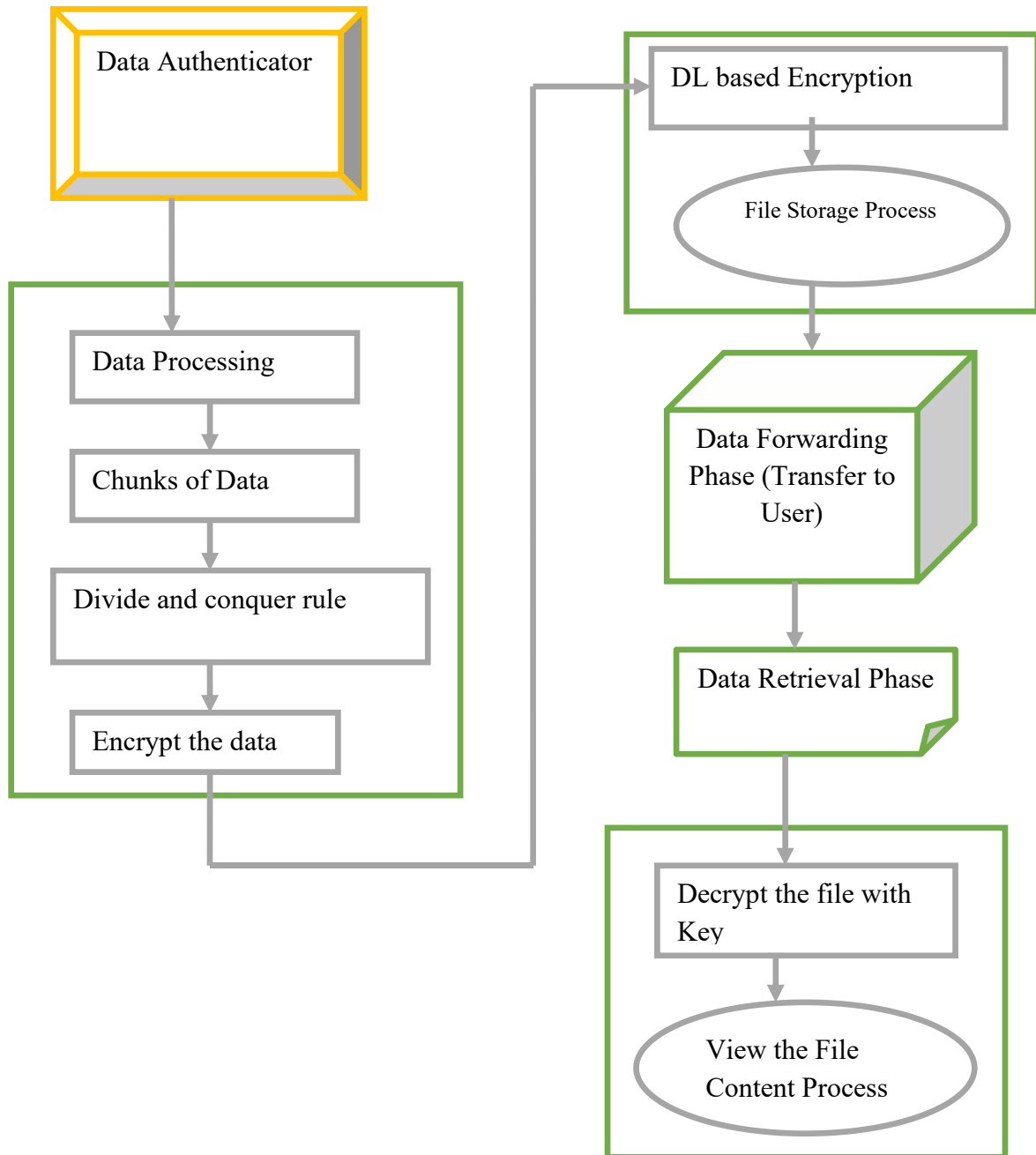


Fig no:4.1 SYSTEM ARCHITECTURE

4.2 USE CASE DIAGRAM:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor.

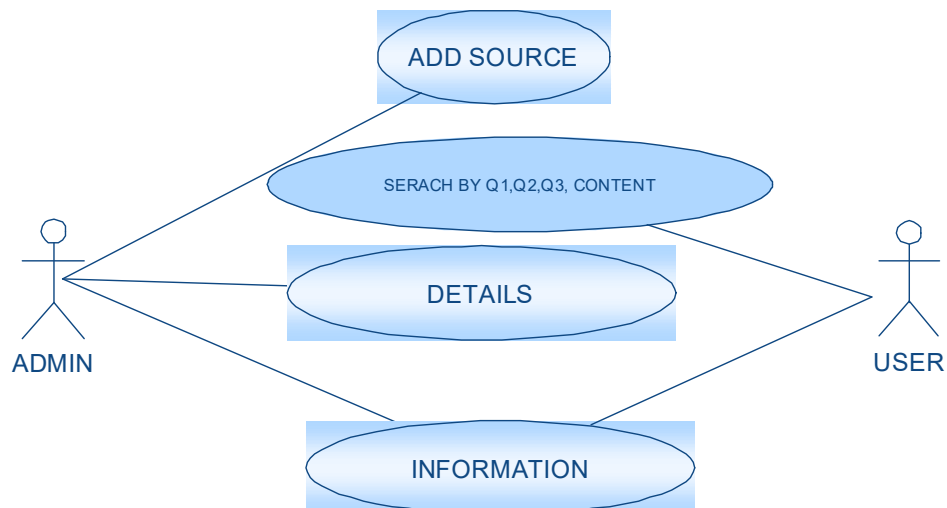


Fig no: 4.2 USE CASE DIAGRAM

4.3 ACTIVITY DIAGRAM:

Activity diagram is another important diagram in UML to describe aspects of the system. Activity diagram basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

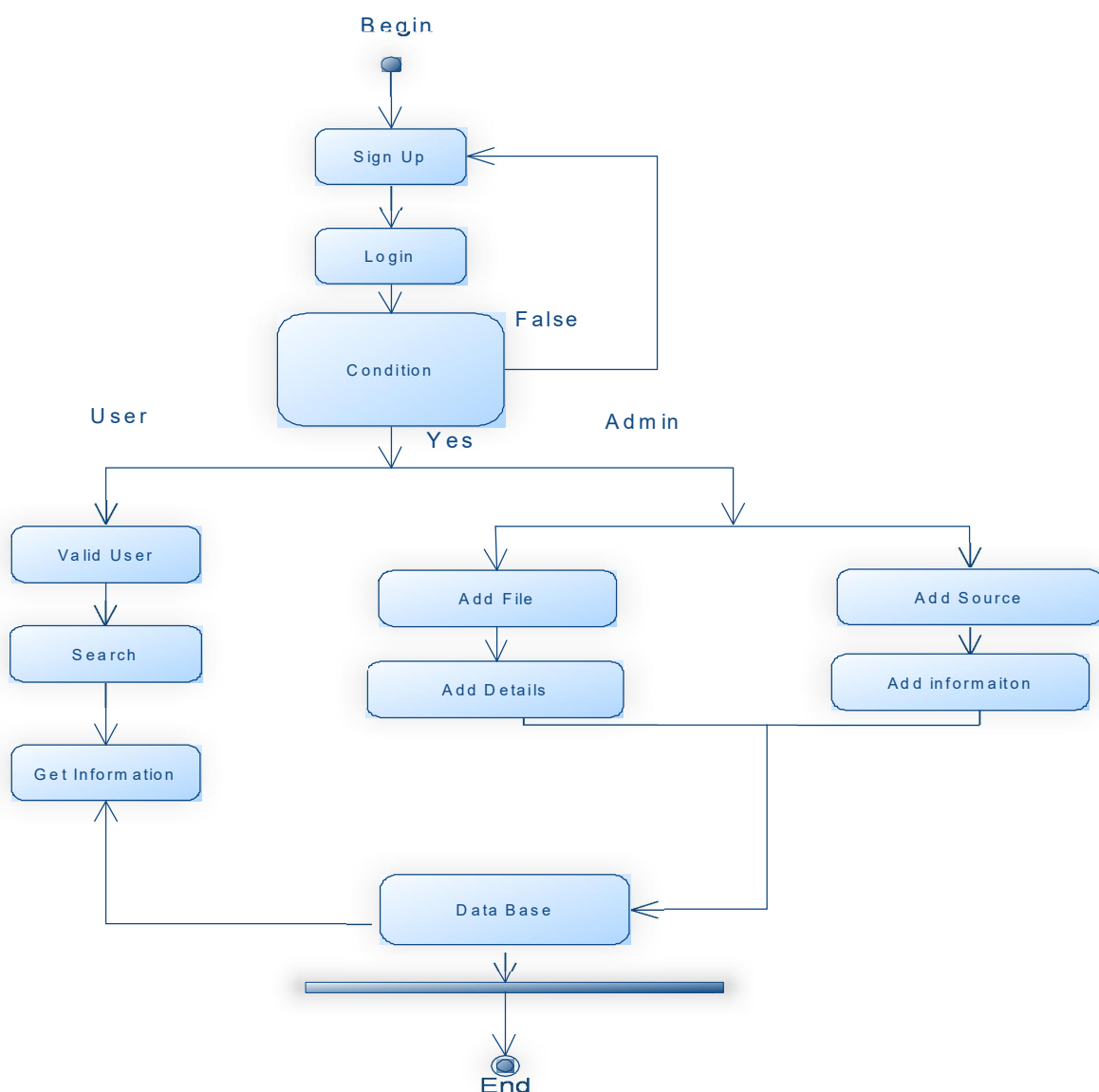


Fig no: 4.3 ACTIVITY DIAGRAM

4.4 SEQUENCE DIAGRAM:

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

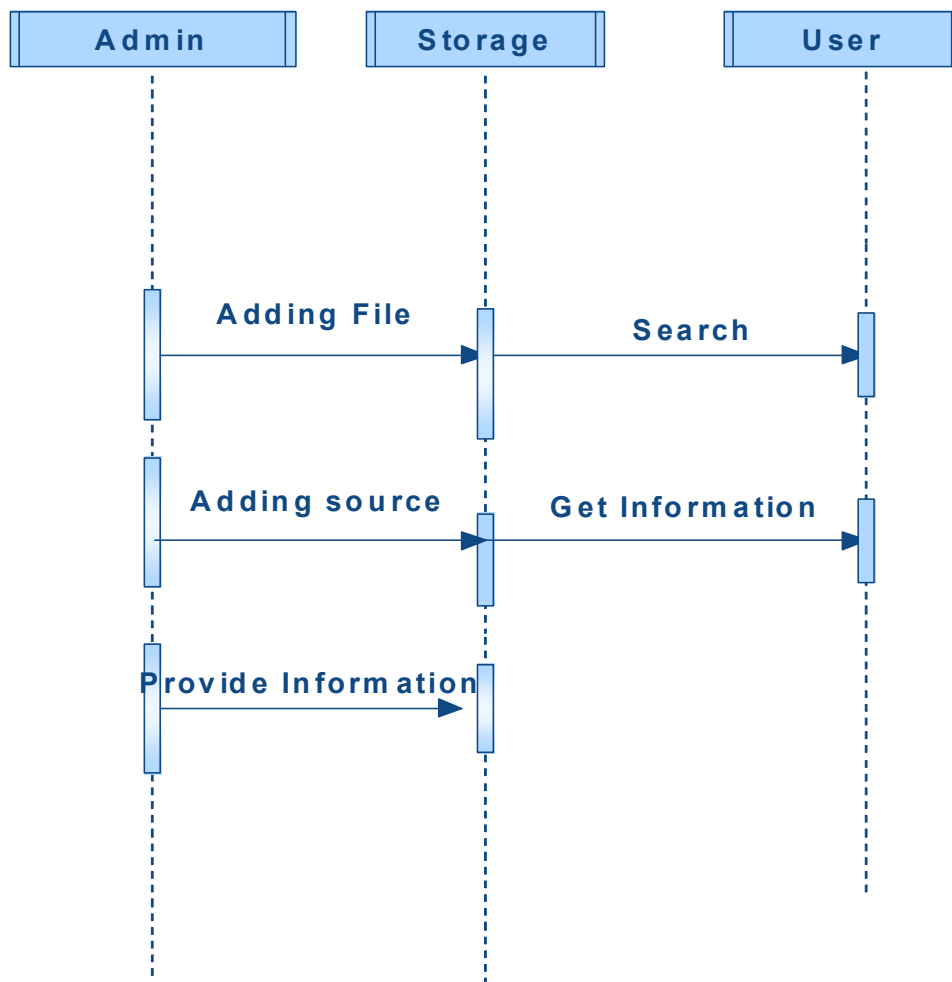


Fig no: 4.4 SEQUENCE DIAGRAM

CHAPTER 5

SYSTEM IMPLEMENTATION

Cloud storage service has shown its great power and wide popularity which provides fundamental support for rapid development of cloud computing. However, due to management negligence and malicious attack, there still lie enormous security incidents that lead to quantities of sensitive data leakage at cloud storage layer. From the perspective of protecting cloud data confidentiality, this work proposed a two schemes namely encrypted data storage and data partition. The encrypted data storage processed an DL based cryptanalysis algorithm and partition is performed by using a divide and conquer approach. The main purpose of this work is to provide idea of the combination of these two algorithms to provide double security to the data stored inside the cloud.

Initially DL-based cryptanalysis proposed a generic cryptanalysis model based on deep learning (DL), where the model tries to find the key of block ciphers from known plaintext-ciphertext pairs. We show the feasibility of the DL-based cryptanalysis by attacking on lightweight block ciphers such as simplified DES, Simon, and Speck. Next, the divide and conquer approach is performed. The divide-and-conquer, breaks a problem into subproblems that are similar to the original problem, recursively solves the subproblems, and finally combines the solutions to the subproblems to solve the original problem. Also, it protects the data from an unauthorized user. The experimental results indicate that proposed mechanism is not only suitable for ensuring the data security at storage layer but also can store huge amount of cloud data effectively in a minimum time overhead.

5.1 MODULES

- Data owner
- User modules
- Admin
- Data splitting
- Encryption Module

5.2 MODULE DESCRIPTION

5.2.1 DATA OWNER MODULE

- In this module, the data Owner information is registered. This information includes a username, password, mail id to login this system.
- The data owners details are verified and accepted by the admin.
- After the data owner registration, they can easily upload the data in the cloud and get a protection from this system.
- The admin can provide a security by protecting the owner's data by splitting and encrypting process

5.2.2 USER MODULE

- In this module, the user information is registered. This information includes a username, password, mail id to login this system.
- The user details are verified and accepted by the admin.
- After the user registration, they can easily access the data uploaded in the cloud without any malware.
- The admin can provide a user to access the file with a valid key.

5.2.3 ADMIN/SERVER MODULE

- The admin is an intermediate between a data owner and a data user. The overall system is controlled and monitored by an admin.

- The admin comprises all the details of owners and user and monitor the data processing status.
- The data is stored in the cloud and encrypted using a DL algorithm to secure the data
- The quality of services are improved by admin providing a better security and performance of data transaction.

5.2.4 DATA SPLITTING

- To improve the additional protection for the cloud data, the encrypted data is partitioned into several form using a Divide and conquer method.
- This method is used to split the single data file into three data file and provided a key for it.
- When the user put a valid key to access the data, then the partitioned data are merged and view as original file to the user.

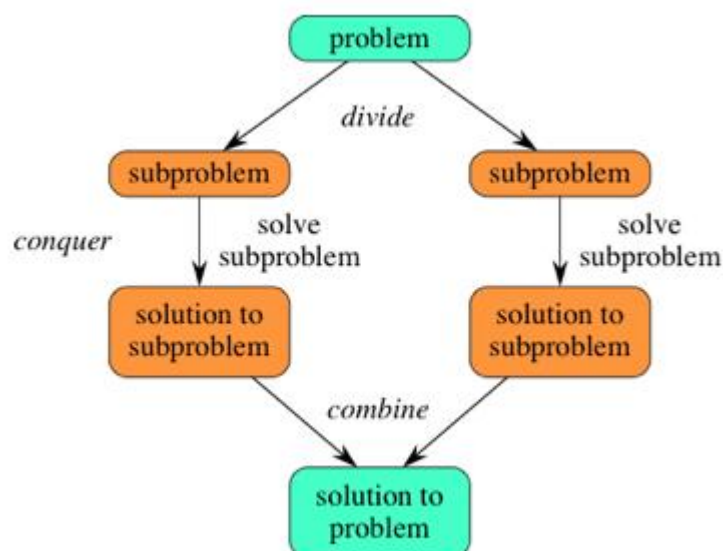
Divide and conquer algorithms

The two sorting algorithms we've seen so far, selection sort and insertion sort, have worst-case running times of $\Theta(n^2)$. When the size of the input array is large, these algorithms can take a long time to run. In this tutorial and the next one, we'll see two other sorting algorithms, merge sort and quicksort, whose running times are better. In particular, merge sort runs in $\Theta(n \lg n)$ time in all cases, and quicksort runs in $\Theta(n \lg n)$ time in the best case and on average, though its worst-case running time is $\Theta(n^2)$. Here's a table of these four sorting algorithms and their running times:

| Algorithm | Worst-case running time | Best-case running time | Average-case running time |
|----------------|-------------------------|------------------------|---------------------------|
| Selection sort | $\Theta(n^2)$ | $\Theta(n^2)$ | $\Theta(n^2)$ |
| Insertion sort | $\Theta(n^2)$ | $\Theta(n)$ | $\Theta(n^2)$ |
| Merge sort | $\Theta(n \lg n)$ | $\Theta(n \lg n)$ | $\Theta(n \lg n)$ |
| Quicksort | $\Theta(n^2)$ | $\Theta(n \lg n)$ | $\Theta(n \lg n)$ |

Divide-and-conquer

Both merge sort and quicksort employ a common algorithmic paradigm based on recursion. This paradigm, **divide-and-conquer**, breaks a problem into subproblems that are similar to the original problem, recursively solves the subproblems, and finally combines the solutions to the subproblems to solve the original problem. Because divide-and-conquer solves subproblems recursively, each subproblem must be smaller than the original problem, and there must be a base case for subproblems.



It should think of a divide-and-conquer algorithm as having three parts:

1. **Divide** the problem into a number of subproblems that are smaller instances of the same problem.
2. **Conquer** the subproblems by solving them recursively. If they are small enough, solve the subproblems as base cases.
3. **Combine** the solutions to the subproblems into the solution for the original problem.

5.2.5 Encryption Module

- The cloud data is secured in this system by generating a key between the owner and user using DL algorithm.
- The DNN based cyto-analysis is used the same secret key to both encrypt and decrypt messages.

DNN Learning Framework.

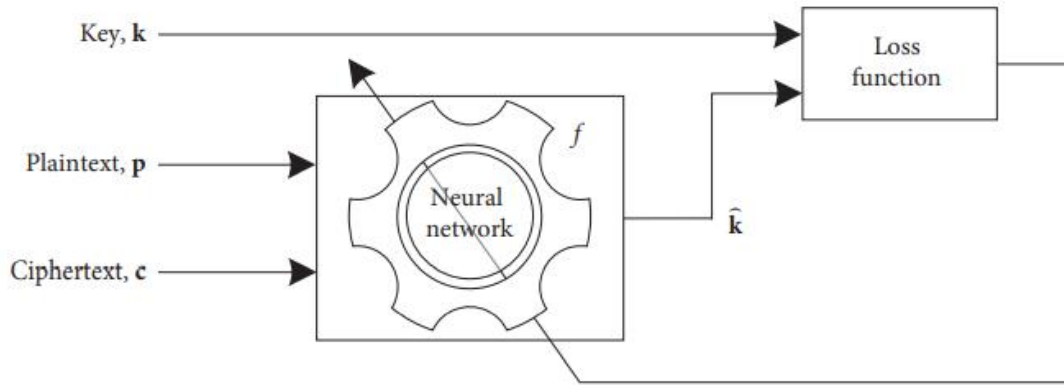
The modern term “DL” is considered as a better principle of learning multiple levels of composition, which uses multiple layers to progressively extract higher level features from the raw input [29]. In the DL area, a DNN is considered as one of the most popular generative models. As a multilayer processor, the DNN is capable of dealing with many nonconvex and nonlinear problems. The feedforward neural network forms a chain, and thus, the feedforward neural network can be expressed as

$$f(\mathbf{x}; \boldsymbol{\theta}) = f^{(L+1)}\left(f^{(L)}\left(\dots f^{(1)}(\mathbf{x})\right)\right),$$

where \mathbf{x} is the input, the parameter $\boldsymbol{\theta}$ consists of the weights \mathbf{W} and the biases \mathbf{b} , $f^{(l)}$ is called the l th layer of the network, and L is the number of hidden layers. Each layer of the network consists of multiple neurons, each of which has an output that is a nonlinear function of a weighted sum of neurons of its preceding layer. The output of the j th neuron at the l th layer can be expressed as

$$j^{(l)} = f^{(l)} \left(\sum_i w_{ij}^{(l)} u_i^{(l-1)} + b_j^{(l)} \right),$$

where $w_{ij}^{(l)}$ is the weight corresponding to the output of the i th neuron at the preceding layer and $b_j^{(l)}$ is the bias. We apply a DNN to find the key of lightweight block ciphers. The multilayer perception mechanism and special training policy promote the DNN to be a commendable tool to find affine approximations to the action of a cipher algorithm. We train the DNN by using N_r pairs of (p, c) randomly generated with different keys in order that the system f finds affine approximations to the action of a cipher, as shown in Figure 1. In Figure 1, the loss function can be the mean square error (MSE) between the encryption key, k , and the output of the DNN, \hat{k} . The performance of the trained DNN is evaluated by using N_t pairs randomly generated with different keys. Finally, given M known plaintexts, we find the key by using the trained DNN and the majority decision.



Schematic diagram of the DL-based cryptanalysis.

DNN Structure for the Cryptanalysis.

The structure of a DNN model for the cryptanalysis is shown in Figure 2. We consider a ReLU function, $f_{ReLU}(x) = \max(0, x)$, as the nonlinear function. The DNN has n_l neurons at the l th hidden layer, where $l = 1, \dots, L$. Each neuron at the input layer associates each bit of the plaintext and ciphertext; that is, the

ith neuron represents p_i , and the $(j + n - 1)$ th neuron represents c_j , where $i, j = 0, 1, \dots, n - 1$. The number of neurons at the input layer is $2n$. Each neuron at the output layer associates each bit of the key; that is, the output of the i th neuron corresponds to k_i , where $i = 0, 1, \dots, m - 1$. Hence, the number of neurons at the output layer is m . The output of the DNN, $\hat{\mathbf{k}}$, is a cascade of nonlinear transformation of the input data, $[\mathbf{p}, \mathbf{c}]$, mathematically expressed as

$$\hat{\mathbf{k}} = f([\mathbf{p}, \mathbf{c}]; \boldsymbol{\theta}) = f^{(L+1)}(f^{(L)}(\dots f^{(1)}([\mathbf{p}, \mathbf{c}]))$$

where L is the number of hidden layers and $\boldsymbol{\theta}$ is the weights of the DNN.

CHAPTER 6

SAMPLE CODE

LOGIN FORM

```
<!DOCTYPE html>
<%@page
import="java.text.SimpleDateFormat"
%>
<%@page
import="java.sql.ResultSet"%>
<%@include file="dbcon.jsp" %>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width,
initial-scale=1.0" name="viewport">
  <title>Cloud Security</title>
  <meta content=""
name="description">
  <meta content="" name="keywords">
  <!-- Favicons -->
  <link href="assets/img/favicon.png"
rel="icon">
  <link href="assets/img/apple-touch-
icon.png" rel="apple-touch-icon">
  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?
family=Open+Sans:300,300i,400,400i,6
00,600i,700,700i|Raleway:300,300i,400,
400i,500,500i,600,600i,700,700i|Poppi
ns:300,300i,400,400i,500,500i,600,600i,
700,700i" rel="stylesheet">
  <!-- Vendor CSS Files -->
```

```
  <link
href="assets/vendor/animate.css/animat
e.min.css" rel="stylesheet">
  <link
href="assets/vendor/bootstrap/css/boots
trap.min.css" rel="stylesheet">
  <link href="assets/vendor/bootstrap-
icons/bootstrap-icons.css"
rel="stylesheet">
  <link
href="assets/vendor/boxicons/css/boxic
ons.min.css" rel="stylesheet">
  <link
href="assets/vendor/glightbox/css/gligh
tbox.min.css" rel="stylesheet">
  <link
href="assets/vendor/swiper/swiper-
bundle.min.css" rel="stylesheet">
  <!-- Template Main CSS File -->
  <link href="assets/css/style.css"
rel="stylesheet">
  <!--
```

```
=====
=====
* Template Name: Green - v4.7.0
* Template URL:
https://bootstrapmade.com/green-free-
one-page-bootstrap-template/
* Author: BootstrapMade.com
* License:
https://bootstrapmade.com/license/
```

```

=====
=====
→
<%

if(request.getParameter("submit") !=
null)
{
    String
name=request.getParameter("name").to
String();
    String
pass=request.getParameter("pass").toStr
ing();
    ResultSet
rs=st.executeQuery("select * from reg
where      uname='"+name+"'      and
pass='"+pass+"'");
    if(rs.next())
    {
        out.println("<script>alert('Login
Successfully')</script>");
        out.println("<script>window.location='
uhome.jsp'</script>");
        session.setAttribute("un",name);
        session.setAttribute("email",rs.getString
("email"));
    }
    else

{
    out.println("<script>alert('Login
Failed')</script>");
    out.println("<script>window.location='l
ogin.jsp'</script>");
}
}
%>

```

```

</head>

<body>

    <!-- ===== Top Bar ===== -->
    <section id="topbar" class="d-flex
align-items-center">
        <div class="container d-flex justify-
content-center      justify-content-md-
between">
            <div class="contact-info d-flex
align-items-center">
                </div>
            <div class="social-links d-none d-
md-block">
                <a href="#" class="twitter"><i
class="bi bi-twitter"></i></a>
                <a href="#" class="facebook"><i
class="bi bi-facebook"></i></a>
                <a href="#" class="instagram"><i
class="bi bi-instagram"></i></a>
                <a href="#" class="linkedin"><i
class="bi bi-linkedin"></i></i></a>
            </div>
        </div>
    </section>

    <!-- ===== Header ===== -->
    <header id="header" class="d-flex
align-items-center">
        <div class="container d-flex align-
items-center">
            <h1 class="logo me-auto"><a
href="index.html">Cloud
Security</a></h1>
            <!-- Uncomment below if you
prefer to use an image logo -->

```

CHAPTER 7

SYSTEM TESTING AND STUDY

SYSTEM TESTING

When a system is developed, it is expected that it performs properly. In practice, however, some errors always occur. The main purpose of testing an information system is to find the errors and correct them. A successful test is one, which find an error. The main objectives of the system testing are

- To ensure during the operation that the system will perform as per specified in the design phase.
- To make sure that the system meets user requirements during operations.
- To verify that the controls incorporated in the system functions as intended.
- To see that if correct inputs are fed into the system, it provides perfect output.
- To verify that during operation incorrect input processing and output will be deleted.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. If the testing conducted successfully, it will uncover errors in the software. As a secondary benefit, testing demonstrates that the software functions appear to be working according to specification and that performance requirements appear to have been made.

The scope of the system test should include both manual operations and computer operations system testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls.

System testing is the process of checking if the developed system is working according to the original objectives and requirements.

This will ensure that the test coverage meets the requirement and that testing is done in semantic manner. System testing accounts for the largest percentage of technical effort in the software development phase.

There are three aspects of system testing:

- Unit testing
- Validation testing
- Sub System testing

7.1 Unit Testing

The application is tested in small units like functions so as to check that the function achieves the desired functionality when correct input is supplied. In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use.

Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object-oriented programming a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

7.2 Integration Testing

Integration testing is to test the project by connecting each module. The admin has to check whether the connection between the modules is correct.

7.3 Validation testing

The application is tested to check how it responds to various kinds of input given. The user should be intimated of any kind of exceptions in a more understandable manner so that debugging becomes easier.

At the culmination of the black box testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected. Next stage is the

validation testing and it can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in the manner that can be reasonably expected by the user. When an user enters incorrect inputs it should not display error messages, instead it should display helpful messages enabling user to use the tool properly. The tool is tested with test data as well as live data and has been found to work properly in the networked environment.

Validation is Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders. For example, when the user skips the username it shows an error message asking for username.

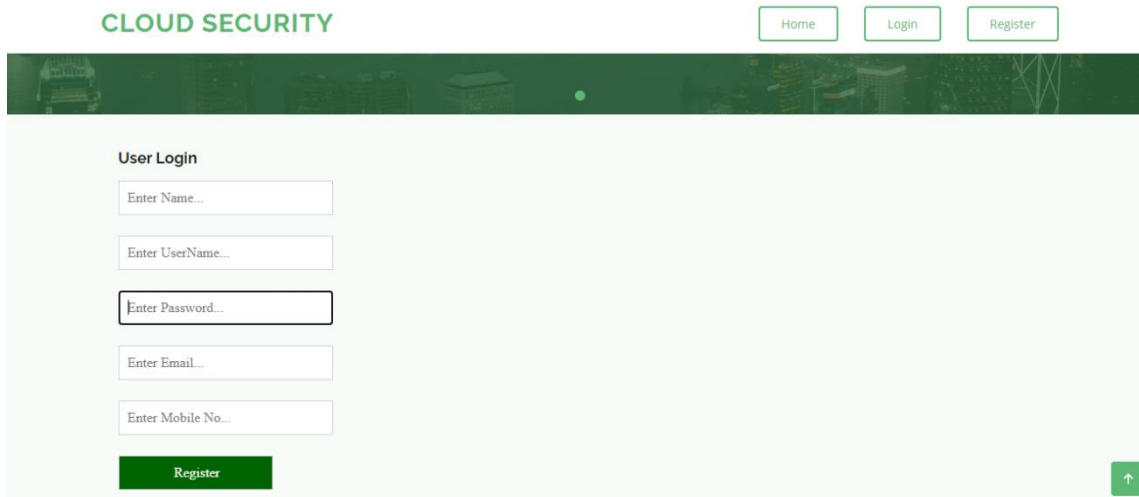
7.4 Verification Testing

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process. In the verification process, after giving the username and password the data is verified with the data in the data.

In most ways, System Security behaves in ways that are typical to rogue anti-virus programs. Infections typically occur by way of Trojan Horse, and the Trojan is hidden in some ordinary, harmless-looking thing you come across online. System Security infections are often the result of a Trojan hidden in a video codec, free download, or even a click-through pop-up screen on a website. That means that if your computer is infected with System Security, it is a surprise to you. This is all part of the scam's attempt to get you to think that System Security has some legitimate connection.

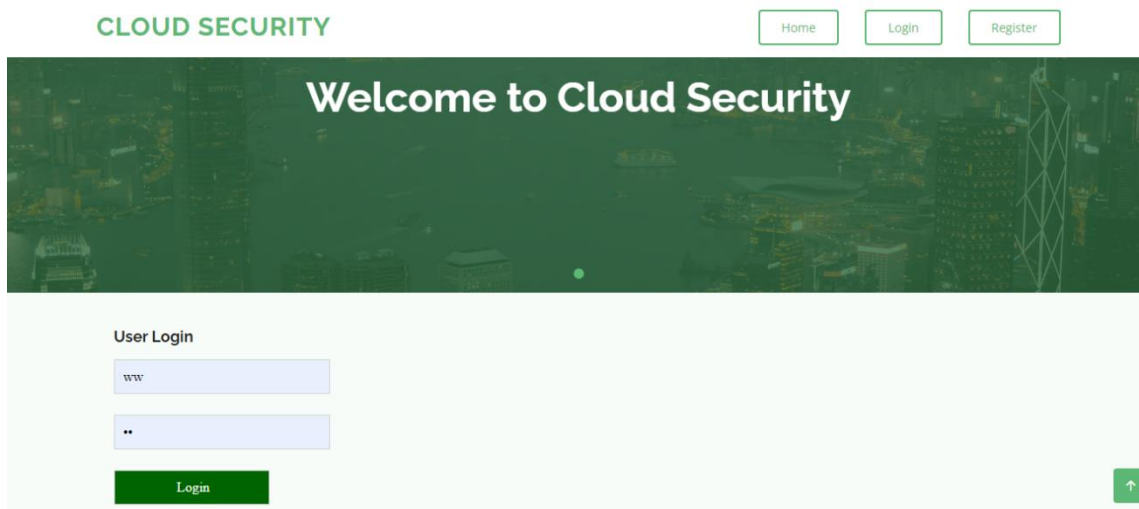
CHAPTER 8

SCREENSHOTS



The screenshot shows the 'Cloud Security' website's registration page. At the top, the text 'CLOUD SECURITY' is on the left, and three buttons labeled 'Home', 'Login', and 'Register' are on the right. Below this is a dark green banner with a cityscape background. The main content area is light green and contains a 'User Login' section. This section has five input fields: 'Enter Name...', 'Enter UserName...', 'Enter Password...', 'Enter Email...', and 'Enter Mobile No...'. Below these fields is a green 'Register' button. A small green button with an upward arrow is in the bottom right corner.

Fig: 8.1 Registration



The screenshot shows the 'Cloud Security' website's user login page. At the top, the text 'CLOUD SECURITY' is on the left, and three buttons labeled 'Home', 'Login', and 'Register' are on the right. Below this is a dark green banner with a cityscape background and the text 'Welcome to Cloud Security' in white. The main content area is light green and contains a 'User Login' section. This section has two input fields: the first contains 'ww' and the second contains '..'. Below these fields is a green 'Login' button. A small green button with an upward arrow is in the bottom right corner.

Fig: 8.2 User Login

CLOUD SECURITY

Home

Files

Other Files

Request

Download

Logout

User Upload

file1

Choose File

doc1.txt

Yes

Upload

Fig: 8.3 File upload

CLOUD SECURITY

Home

Files

Other Files

Request

View Uploaded Files

| S.NO | FILE NAME | EXT | UPLOADED DATE | UPLOADED BY |
|------|-----------|-----|---------------------|-------------|
| 1 | aa | | 2022:01:30 17:21:00 | a |
| 2 | aq | txt | 2022:01:30 18:41:57 | a |
| 3 | en | txt | 2022:02:01 19:52:05 | a |
| 4 | un | txt | 2022:02:01 19:54:59 | a |
| 5 | aan | txt | 2022:02:01 20:06:05 | a |
| 6 | qw | txt | 2022:02:01 20:09:46 | a |
| 7 | aa | txt | 2022:02:01 21:53:33 | a |
| 8 | al | txt | 2022:02:01 21:55:38 | a |
| 9 | aq | txt | 2022:02:01 21:56:31 | a |
| 10 | en | txt | 2022:02:01 22:00:06 | a |
| 11 | g | txt | 2022:02:01 22:02:45 | a |
| 12 | kk | txt | 2022:02:01 22:05:27 | a |
| 13 | aa | txt | 2022:02:01 22:06:44 | a |
| 14 | qq | txt | 2022:02:01 22:08:39 | a |
| 15 | a | txt | 2022:02:01 22:09:56 | a |
| 16 | z | txt | 2022:02:01 22:13:41 | a |

Fig: 8.4 Uploaded files








| > smart > CloudMe | | ▼ | ↺ |
|-------------------|---|---------------------|---|
| Personal | Name | Date modified | |
| |  desktop | 11-04-2022 02:32 PM | |
| |  fi11 | 22-04-2022 08:30 PM | |
| |  fi12 | 22-04-2022 08:30 PM | |
| |  fi13 | 22-04-2022 08:30 PM | |
| |  file11 | 01-05-2022 03:33 PM | |
| |  file12 | 01-05-2022 03:33 PM | |
| |  file13 | 01-05-2022 03:33 PM | |

Fig: 8.5 Cloud data partition

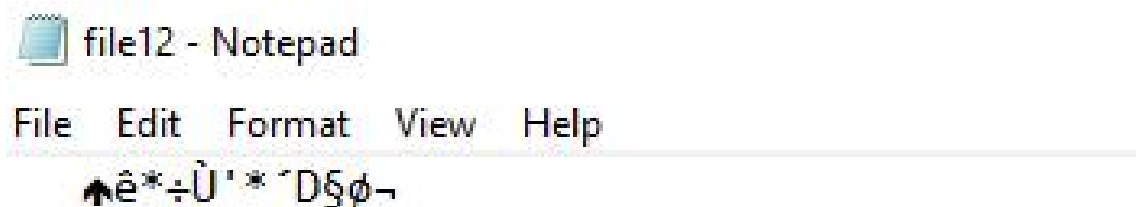


Fig: 8.6 Data encryption

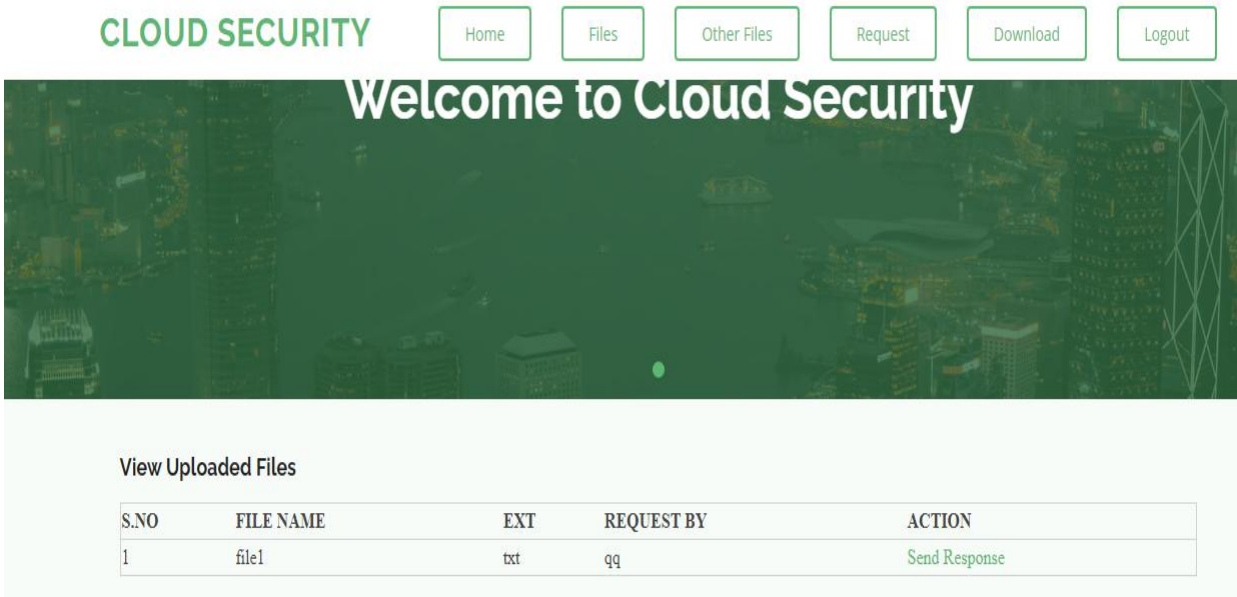


Fig: 8.7 Data request

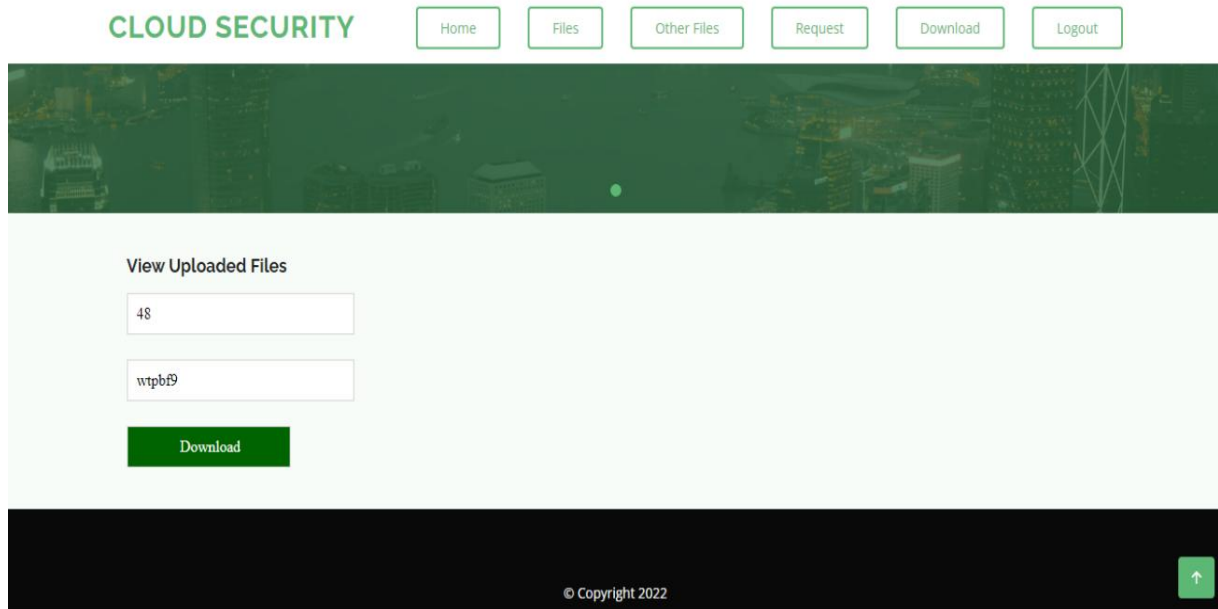


Fig: 8.8 Download

CHAPTER 9

CONCLUSION

This work proposed a two schemes namely encrypted data storage and data dispersion. The encrypted data storage processed an **DL based cryptanalysis** algorithm and partition is performed by using a **divide and conquer** approach. The main purpose of this work is to provide idea of the combination of these two algorithms to provide double security to the data stored inside the cloud. The **divide and conquer** method is used to split the single data file into three data file and provided a key for it. When the user put a valid key to access the data, then the partitioned data are merged and view as original file to the user. The DL-based cryptanalysis model and evaluated the performance of the DL-based attack on the S-DES, Simon32/64, and Speck32/64 ciphers. The accuracy of DL is improved, and the accuracy becomes more precise, thanks to the development of algorithms and hardware. Moreover, advanced data transformation that efficiently maps cryptographic data onto ML data will help the DL-based cryptanalysis to be performed without the keyspace restriction.

CHAPTER 10

FUTURE ENHANCEMENT

Gathering data and observing trends will always leave some unnoticed facts, some overlooked trends, and may very well include irrelevant data that should have been ignored. Having said that, making an educated guess will probably be more beneficial than choosing to remain completely indifferent and oblivious to the future. "It's our responsibility as software developers to anticipate what's to come and to deliver long term solutions accordingly." Web development, as a set of technologies, practices, and institutes, is formed mainly by two forces: hardware, or more specifically, consumer devices, and the same old eternal pursuit of financial gain. When taking the task of predicting the future of web dev, an investigation of these two should be our starting point.

If our aim is to get the maximum usage for our web apps, we must plan ahead and make them available to a large and unpredictable range of devices. That means, building them in such a way that would enable us to customize them easily to any existing device but also, to be prepared for the next new thing.

CHAPTER 11

REFERENCES

- [1] R. Amin, S. H. Islam and K. -K. R. Choo, "Provably Secure and Lightweight Identity-Based Authenticated Data Sharing Protocol for Cyber-Physical Cloud Environment," in *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 318-330, 1 Jan.-March 2021, doi: 10.1109/TCC.2018.2834405.
- [2] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan and L. Mostarda, "Capturing-the-invisible (CTI): Behavior-based attacks recognition in IoT-oriented industrial control systems", *IEEE Access*, vol. 8, pp. 104956-104966, 2020.
- [3] M. Kumar, A. Rani and S. Srivastava, "Image forensics based on lighting estimation", *Int. J. Image Graph.*, vol. 19, no. 3, Jul. 2019.
- [4] K. Lee, "Comments on "Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption"," in *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1299-1300, 1 Oct.-Dec. 2020.
- [5] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun and Y. Xiang, "Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing," in *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 996-1010, 1 Nov.-Dec. 2019, doi: 10.1109/TDSC.2017.2725953.
- [6] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu and J. Ma, "Data Integrity Auditing without Private Key Storage for Secure Cloud Storage," in *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1408-1421, 1 Oct.-Dec. 2021.
- [7] X. Yang, R. Lu, J. Shao, X. Tang and A. A. Ghorbani, "Achieving Efficient Secure Deduplication with User-Defined Access Control in Cloud," in *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 591-606, 1 Jan.-Feb. 2022, doi: 10.1109/TDSC.2020.2987793.
- [8] Y. Yang, X. Zheng, C. Rong and W. Guo, "Efficient Regular Language Search for Secure Cloud Storage," in *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, pp. 805-818, 1 July-Sept. 2020, doi: 10.1109/TCC.2018.2814594.
- [9] H. Yin, Z. Qin, J. Zhang, L. Ou and K. Li, "Achieving Secure, Universal, and Fine-Grained Query Results Verification for Secure Search Scheme Over Encrypted Cloud Data," in *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 27-39, 1 Jan.-March 2021, doi: 10.1109/TCC.2017.2709318.
- [10] Y. Zhang, Y. Mao, M. Xu, F. Xu and S. Zhong, "Towards Thwarting Template Side-Channel Attacks in Secure Cloud Deduplications," in *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1008-1018, 1 May-June 2021, doi: 10.1109/TDSC.2019.2911502.

