# Automatic Self Destruction of Data In Cloud

**Ms. Jenila L[1], Divya P[2], Priyanka C[3],Gajalakshmi E[4]**
[1]Assistant professor, Department of Computer Science and Engineering
[2,3,4] Final year UG Students, Department of Computer Science and Engineering

*Abstract*— **The process of Data sharing in cloud storage is the method of receiving substantial attention in information and communications technology as it can provide the users efficient and effective storage of services. The cryptographic techniques are usually applied in order to protect the confidentiality of the shared sensitive data. The data protection is posing significant challenges in cloud storage for sharing of data. The fundamental challenge is the protection and revoking of the cryptographic key. In order to tackle this, a new data protection mechanism for cloud storage is proposed in this project which holds the following properties. Initially the cryptographic key is protected the two factors. The secrecy of the cryptographic key is held only if one of the two factors works. The next property is that the cryptographic key can be revoked efficiently by integrating the proxy re-encryption and the key separation techniques. The attribute-based encryption technique is adopted finally to protect the data in a fine-grained way. Furthermore, the performance evolution and the security analysis show that this proposal is secure and efficient, respectively.**

***Index Terms***— **Cryptographic-key, Proxy Re-encryption, key separation, security analysis.**

## INTRODUCTION

The shared data in cloud servers contains user's sensitive information (e.g., personal profile, financial data, health records, etc.) that are needed to be well protected. The cloud servers may migrate users's data to other cloud servers in outsourcing or share them in cloud searching as the ownership of the data is separated from the administration of` them. It is a big challenge to protect the privacy of those shared data in cloud in cross-cloud and big data environment. Once the user-defined time is expired the shared data should be self-destroyed. The storage of data as a common encrypted form is one of the methods to alleviate the problems. The user cannot share his/her encrypted data at a fine-grained level is the major disadvantage of encrypting data. The significant advantage based on the tradition public key encryption instead of one-to-one encryption is achieved through the Attribute-based encryption (ABE). Both data security and fine-grained access control can be achieved through ABE(Attribute-based encryption) scheme. The cipher-text is labelled with set of descriptive attributes by the key-policy ABE (KP-ABE) scheme. The encryption service has been provided by the Timed-release encryption (TRE) where an encryption key is associated with a predefined release time, and a receiver can only construct the corresponding decryption key in this time instance Time-Specific Encryption (TSE) scheme on the basis, which is used to specify a suitable time interval such that the cipher-text can only be decrypted in this interval. The ABE is applied to the shared data that will introduce several problems with regard to time specific constraint and self-destruction and applying TSE, will introduce problems with regard to fine-grained access control. This paper attempt to solve these problems by using KPABE and adding a constraint of time interval to each attribute in the set of decryption attributes.

## EXISTING SYSTEM

Sharing data among users is perhaps one of the most engaging features that motivate cloud storage. There are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owner's anonymity. The problem arises when a file is shared to multiple users.

## PROPOSED SYSTEM

A key-policy attribute-based encryption with time-specified attributes (KP-TSABE), a novel secure data Autolysis of Data scheme in cloud computing is proposed here. Every ciphertext is labeled with a time interval while private key is associated with a time instant in the KP-TSABE scheme. if both the time instant is in the allowed time interval and the attributes associated with the ciphertext satisfy the key's access structure the ciphertext can only be decrypted. Secure Data Sharing in Clouds (SeDaSC) methodology that provides is proposed here: 1) data confidentiality and integrity; 2) access control; 3) data sharing (forwarding) without using compute-intensive re encryption; 4) insider threat security; and 5) forward and backward access control 6) One time download 7) Share Time Expire 8) Secret Key Management.

## MODULES

**Modules:**

1. **Authentication and authorization**
2. **File Encryption and Decryption**
3. **File Sharing**
4. **File Decryption and Download**
5. **File Autolysis of Data and Access Control**

**1. Authentication and authorization:** In this module, the User has to register first, and then only he or she can access the data base. Once the registration is complete, the user can login to the site. The authorization and authentication process is used to facilitate the system to protect itself and it also protects the whole mechanism from unauthorized usage. The Registration involves in getting the details of the users who wants to use this application.

**2. File Encryption and Decryption**: In this module, the user uploads the files which he/she wants to share. Initially, the uploaded files will be stored in the Local System. Next the user will upload the file to the real Cloud Storage (In this application, Dropbox is used). While uploading the file to the Cloud, the file gets encrypted by using the AES (Advanced Encryption Standard) Algorithm and the Private Key will be generated. Then the Encrypted Data will be converted to Binary Data for Data security and will be stored in Cloud.

**3. File Sharing:** The uploaded files will be shared to the friends or users in this module. The Data Owner will set the time to expire the data in Cloud. The Private Key of the Shared Data is sent through Email.

**4. File Decryption and Download:** In this Module, the user can download the data by decrypting technique. This can be done by using the AES (Advanced Encryption Standard) Algorithm. The users must give the corresponding Private Keys in order to decrypt the data. If the user enters the Wrong Private Key for Three times then the Data will be deleted. An intimation email will be sent to the Data owner, if the file gets deleted. The Downloaded Data is stored in the Local Drive.

**5. File Autolysis of Data and Access Control:** In this module, the Data will be deleted automatically if the User does not download the file successfully within the time given by the data owner. The File Autolysis will be disabled if the user downloads the data. An intimation Email will be sent to the Data Owner if the file gets deleted by the File Autolysis Scheme. If any malicious is attached to the shared file then the shared user will receive an intimation i.e., to block the backward access in the website. Example: If a user to logout account then can't go back our previous page.

## 5) SYSTEM ARCHITECTURE

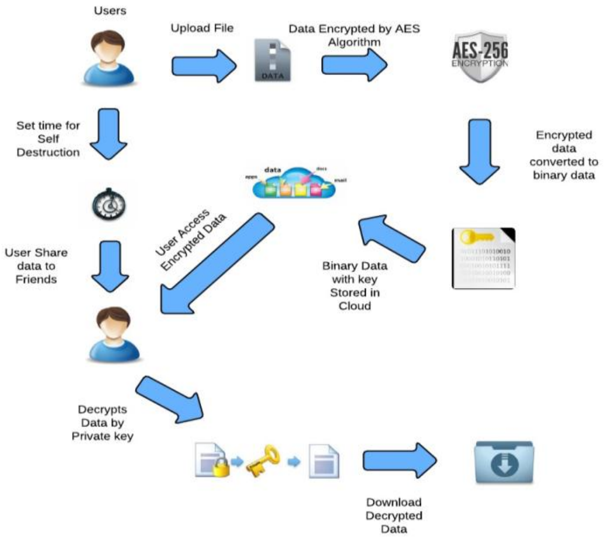

Users
Upload File
Data Encrypted by AES Algorithm
AES-256 ENCRYPTION
DATA
Set time for Self Destruction
Encrypted data converted to binary data
data
apps
docs
mail
User Access Encrypted Data
Binary Data with key Stored in Cloud
User Share data to Friends
Decrypts Data by Private key
Download Decrypted Data

*Figure* (1): System Architecture

## 6) SYSTEM CONFIGURATION

***Hardware Requirements:***

| | | |
|---|---|---|
| ➢ Processor | : | Pentium –III |
| ➢ Speed | : | 1.1 GHz |
| ➢ RAM | : | 256 MB (min) |
| ➢ Hard Disk | : | 20 GB |
| ➢ Floppy Drive | : | 1.44 MB |
| ➢ Key Board | : | Standard Windows Keyboard |
| ➢ Mouse | : | Two or Three Button Mouse |
| ➢ Monitor | : | SVGA |

***Software Requirements:***

| | | |
|---|---|---|
| ➢ Operating System | : | Windows |
| ➢ Front End | : | Java JDK1.8.2 |
| ➢ Front End Tool | : | NetbeansIDE8.2 |
| ➢ DatabaseConnectivity | : | Mysql5.0 |
| ➢ Database Tool | : | SQLyog |

## 7) SYSTEM STUDY

### A) Feasibility Study

The analysis of a problem to determine if it can be solved effectively is called the feasibility study. The results will determine whether the solution should be implemented or not. This activity will take place during the project initiation phase and will be made before significant expenses are engaged. An evaluation of a proposal, designed to determine the difficulty in carrying out a designated task can be called as feasibility study. A feasibility study precedes development in technical field and implementation of project. Usually, a feasibility study looks at the viability of an idea with an emphasis on identifying potential problems.

### B) Java

Initially the java language was called as "oak" but it was renamed as "java" in early 1995. Java is a platform independent language. The primary motivation of the java language was the need for a platform-independent language also called as architecture neutral language that could also be used to create software to be embedded in various consumer electronic devices.

- Java is a programming language
- Java is consistent and cohesive
- Java gives the programmer, full control
- Java is for Internet Programming where c is for System Programming.

### C) Servlets

Java Servlets are the programs that run on a Web or an Application server and it acts as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server. Servlets can be used to collect input from the users through web page forms, present records from a database or another source, and create web pages dynamically. Java Servlets serve as programs that are implemented using the Common Gateway Interface (CGI). Servlets offer many advantages in the comparison with the CGI in which the performance is significantly better. Servlets are executed within the address space of a Web server. A separate process is not necessarily to be created to handle each client request. Servlets are platform-independent as they are written in Java. The Java security manager on the server enforces a set of restrictions to protect the resources on a server machine so the servlets are trusted. The full functionality of the Java class libraries are available to a servlet. The Servlets can communicate with the applets, databases, or other software via the sockets and RMI mechanisms.

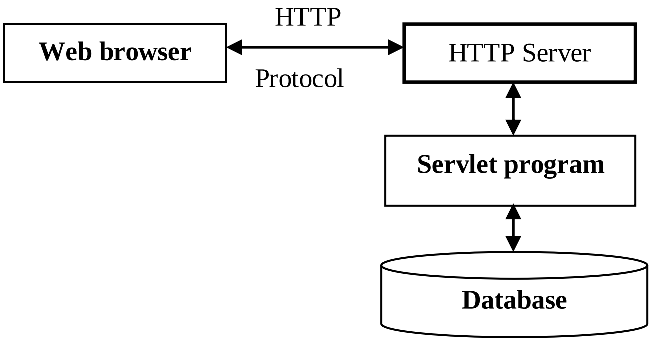


*Figure (2):* Servlet Architecture

## 8) SOFTWARE TESTING

This testing approach document is majorly designed for Information and Technology Services upgrades to PeopleSoft. It contains an overview of the testing activities to be performed when an upgrade or enhancement is made, or a module is added to an existing application. There are various tests that need to be conducted again in the system testing as follows:

- Test Plan
- Test Case
- Test Data

### A) Unit Testing

The first and the most basic level of Software Testing, in which a single unit is examined in isolation from the remaining source code is called Unit Testing. It is done to verify whether a unit is functioning properly or not. It checks the smallest units of code and proves that the particular unit can work perfectly in isolation.

### B) Integration Testing

Integration testing is performed after Unit Testing in which the software components are clubbed together in large aggregates and tested, to verify the proper functioning, performance and reliability between units, and expose any defect in the interface.

### C) System Testing

System Testing is done after identifying the functional bugs at the Unit and Integration testing level to scrutinize the entire software system. The major objective of system testing is to verify the non-functional part of the software like speed, security, reliability and accuracy. System Testing can also be done to ensure that the software meets the customer's functional and business requirements.

### D) Planning Phase

At the planning phase, the product will be defined by the team of Engineers, Marketers and Sales Staff. Hence, testing at this phase is majorly focused on scrutinizing the idea rather than source codes.

### E) Design Stage

At the design stage, the designers work to figure out on providing planned competencies of the product. It is divided into two forms as Internal Design which explains the internal working of the product and the External Design which describes the product from user's perspective.

- **Walkthrough**: the program is simulated to show how it will work, how different pieces of system will work, etc. to highlight the redundancies and missed details.
- **Inspection**: this technique focuses on handling errors, conforming with the standards and other defined areas, etc.
- **Technical Review**: in this review meeting, Testers discuss the issues related to the program and list down the problems that needs to be changed and redesigned.

### F) Development Stage

In the development, stage software construction, defect diagnosis and prevention strategies are synchronized to reduce development risk, time and cost. The main objective is to eliminate errors, increase quality as well as efficiency of the software, before it proceeds to the Quality Assurance phase. Development Testing can be classified into different types, based on the Company's expectations as data flow analysis, metrics analysis, code coverage analysis, Static Code Analysis, Traceability, Peer Code Review, etc.

### G) Testing Stage

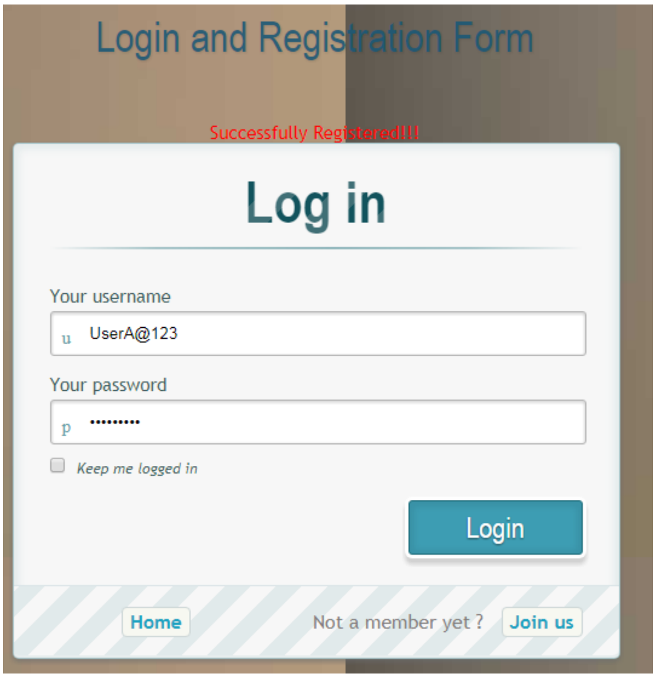
Testing Phase includes the following tests listed below

- Acceptance Test
- Control Flow
- Data Flow and Integrity Test
- Stress Test
- User Interface
- Regression Test
- Performance Test
- Beta Test
- Release Test

## 9) RESULTS AND DISCUSSIONS

In this project, the SeDaSC methodology, which is a cloud storage security scheme for group data is proposed. The proposed methodology provides data confidentiality, secure data sharing without re-encryption, access control for malicious insiders, and forward and backward access control. The SeDaSC methodology provides assured deletion by deleting the parameters required to decrypt a file.

*Figure (3)***:** Home Page



Login and Registration Form

Successfully Registered!!!

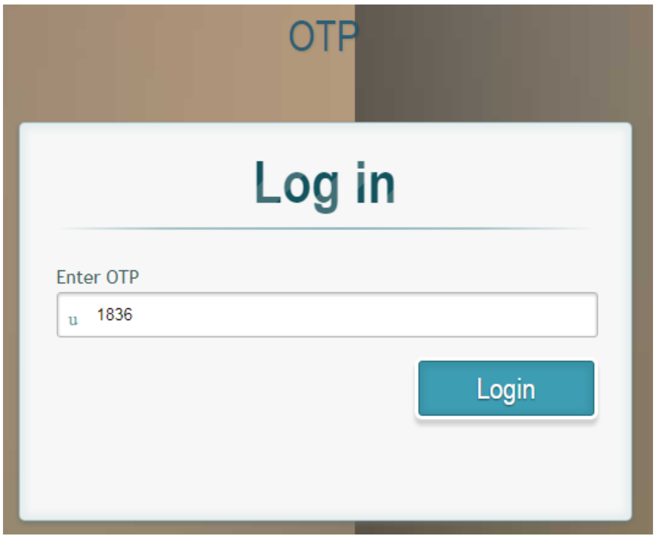
Log in

Your username: u UserA@123

Your password: p •••••••••

□ Keep me logged in

*Figure (4):* Login Page

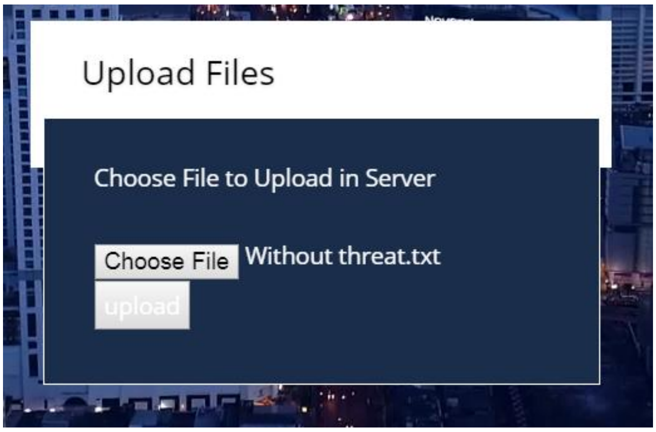


OTP

Log in

Enter OTP: u 1836

*Figure (5):* OTP Verification

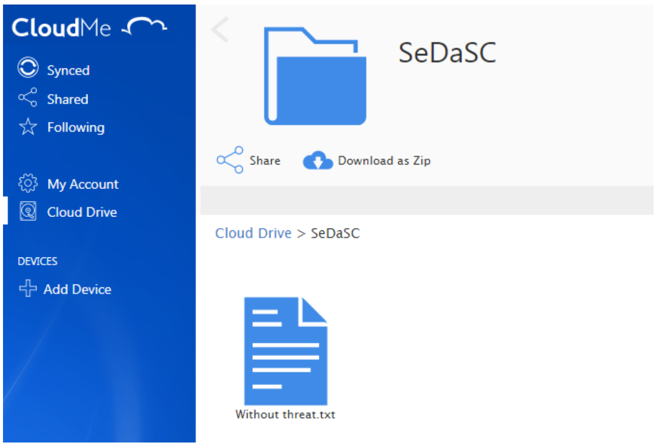


Upload Files

Choose File to Upload in Server

Without threat.txt

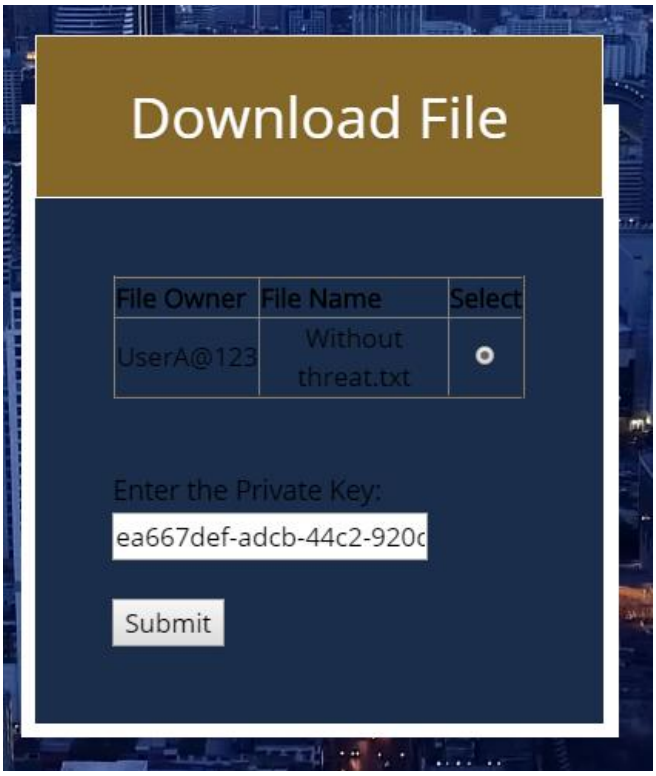
*Figure* (6): Uploading File



Cloud Drive > SeDaSC

Without threat.txt

*Figure (7):* Private cloud file

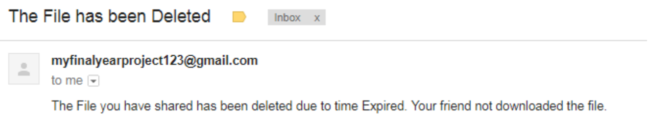


Download File

| File Owner | File Name | Select |
|---|---|---|
| UserA@123 | Without threat.txt | ● |

Enter the Private Key: ea667def-adcb-44c2-920c

*Figure (8):* Downloading file



The File has been Deleted Inbox x

myfinalyearproject123@gmail.com
to me

The File you have shared has been deleted due to time Expired. Your friend not downloaded the file.

*Figure (9):* Time Expired

## 10) CONCLUSION

In this project, the SeDaSC methodology, which is a cloud storage security scheme for group data is proposed. The proposed methodology provides data confidentiality, secure data sharing without re-encryption, access control for malicious insiders, and forward and backward access control. The SeDaSC methodology provides assured deletion by deleting the parameters required to decrypt a file.Since this project is only about sharing files to friends perform computer actions this project has been designed keeping in mind the future scopes. What this project aimed and achieved creating is not a product but a tool to a better automotive environment, a tool can be used to shape many things in the future. Thus this project will give rise to many future modifications forking in all directions. Some of the near future scopes of this project are: There are few interesting problems to be continued to study for the future work. One of them is the user can share a file to multi users at a time. The AES (Advanced Encryption Scheme) to encrypt the Data is used in this project. In future this application can be developed using different types of advanced algorithm for Encryption.

## 11) REFERENCES

1) B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditingfor shared data in the cloud," *Cloud Computing, IEEE Transactions on*, vol. 2, no. 1, pp. 43–56, 2014.

2) J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, and J. Ma, "Priam: Privacy preserving identity and access management scheme in cloud," *KSII Transactions on Internet and Information Systems (TIIS),* vol. 8, no. 1, pp. 282–304, 2014.

3) J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. S. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," *Peerto- Peer Networking and Applications*.[Online].Available:http://dx.doi.org/10.1007/s12083-014-0295-x

4) P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *Cloud Computing, IEEE Transactions on*, vol. 1, no. 2, pp. 142–157, 2013.

5) R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *Network, IEEE*, vol. 28, no. 4, pp. 46–50, 2014.

6) X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *International Journal of Network Security*, vol. 16, no. 4, pp. 351–357, 2014.

7) A.SahaandB.Waters,"Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, ser. LNCS, vol. 7371. Springer, 2005, pp. 457–473.

8) V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and Communications Security*. ACM, 2006, pp. 89–98.

9) A. F. Chan and I. F. Blake, "Scalable, server-passive, useranonymous timed release cryptography," in *Proceedings of the International Conference on Distributed Computing Systems*. IEEE, 2005, pp. 504–513.

10) K. G. Paterson and E. A. Quaglia, "Time-specific encryption," in *Security and Cryptography for Networks*. Springer, 2010, pp. 1–16.

11) Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Large universe decentralized key-policy attribute-based encryption," *Security and Communication Networks*, 2014.[Online].Available:http://dx.doi.org/10.1002/sec.997

12) J.Bethencourt,A.Sahai,andB.Waters,"Ciphertext-policy attribute-based encryption," in *Proceedings of the 28th IEEE Symposium on Security and Privacy*. IEEE, 2007, pp. 321– 334.

13) L. Cheung and C. C. Newport, "Provably secure ciphertext policy abe," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 456– 465.

14) B.Waters,"Ciphertext-policyattribute-based encryption: An expressive, efficient, and provably secure realization," *Public Key Cryptography–PKC 2011*, pp. 53–70, 2011.

15) A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

**AUTHORS**

**Ms. Jenila L** Assistant Professor, Department of Computer Science and Engineering at Loyola Institute of Technology

**Divya P** Final year UG student, Department of Computer Science and Engineering, Loyola Institute of Technology

**Priyanka C** Final year UG student, Department of Computer Science and Engineering, Loyola Institute of Technology

**Gajalakshmi E** Final year UG student, Department of Computer Science and Engineering, Loyola Institute of Technology