## ACKNOWLEDGEMENT

We thank our God for all his blessings showered on us and also for giving us good knowledge and strength in enabling us to finish our project. Our deep gratitude goes to our founder **(Late) D. SELVARAJ M.A., M.L.A.** for his patronage in the completion of our project.

We take this opportunity to thank our honorable chairperson our beloved madam **Dr.S.NALINI SELVARAJ M.Com., M.Phil.** and our honorable director **Mr.S.AMIRTHARAJ B.E.,M.B.A**for their support given to us to finish our project successfully. We wish to express our sincere thanks to our principal **Dr.A.V.MAYAKANNAN Ph.D.** for his kind encouragement and his interest towards us.

We are extremely grateful to Head of Computer Science and Engineering department **DR. D C JULLIE JOSEPHINE Ph.D.,** for her valuable suggestions, guidance and encouragement. I wish to express my dear sense of gratitude and sincere thanks to my supervisor who help us for the successful completion of our project **Mr. B. MUTHAZHAGAN. M. Tech., (Ph.D.)** Associate professor, Department of Information Technology

We express our sincere thanks to my parent, friends and staff members who have helped and encouraged us during the entire course of completing this project work successfully.

## ABSTRACT: -

User activity logs can be a valuable source of information in cloud forensic investigations; hence, ensuring the reliability and integrity of such logs is crucial. Most existing solutions for secure logging are designed for conventional systems rather than the complexity of a cloud environment. In this paper, we propose the Cloud Log Assuring Soundness and Secrecy (CLASS) process as an alternative scheme for the securing of logs in a cloud environment. In CLASS, logs are encrypted using the individual user's public key so that only the user is able to decrypt the content. In order to prevent unauthorized modification of the log, we generate proof of past log (PPL) using Rabin's fingerprint and Bloom filter. Such an approach reduces verification time significantly. Findings from our experiments deploying CLASS in OpenStack demonstrate the utility of CLASS in a real-world context.

|  |  |  |
| --- | --- | --- |
|  | **LIST OF THE FIGURES** |  |
| **S.NO** |  | **Page NO:** |
| 1.1 | System architecture | 5 |
| 5.2 | Sequence diagram | 21 |
| 5.3 | Use case diagram | 22 |
| 5.4 | Class diagram | 23 |
| 5.5 | Activity diagram | 24 |
| 5.6 | Dataflow diagram | 25 |
| 8.3 | Working of java program | 37 |
| 8.5 | Execution diagram | 39 |
| 9.6 | Implementation Of API | 45 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**List of abbreviations**

**UML** Unified Modeling Language

**RSA**    Rivets, Adi Shamir, and Leonard Adleman

**JAVA-VM** JAVA VURTUAL MACHINE

**ODBC** Open Database Connectivity

**JDBC**  Java Database Connectivity

**SQL** Structured Query Language

**API** Application Programming Interface

**ISAM**  Indexed Sequential Access Method

**List of contents**

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT** |  |
|  | **List of figures** |  |
|  | **List of abbreviations** | **(V)** |
|  |  |  |
| **S. NO:** | **TITLE** | **P. NO:** |
| **1.** | **1.1 Introduction** | **2** |
|  | **1.2 Object** | **4** |
| **2.** | **Literature Survey** | **6** |
| **3.** | **System Analysis** | 15 |
|  | 3.1 Exiting System | 16 |
|  | 3.1.1 Disadvantage | 16 |
|  | 3.2 Proposed System | 16 |
|  | 3.2.1 Advantage | 16 |
| **4.** | **System Requirement** | **17** |
|  | 4.1 Introduction | 18 |
|  | 4.2 Hardwar Requirement | 18 |
|  | 4.3 Software Requirement | 18 |
| **5.** | **SYSTEM DESIGN** | **19** |
|  | 5.1 UML Diagram | 20 |
|  | 5.1.1 Introduction | 20 |
|  | 5.2 Sequence Diagram | 21 |
|  | 5.3 Use case Diagram | 22 |
|  | 5.4 Class Diagram | 23 |
|  | 5.5 Activity Diagram | 24 |
|  | 5.6 Dataflow Diagram | 25 |
| **6.** | **SYSTEM IMPLEMENTATION** | **26** |
|  | **6.1 MODULE DISCRIPTION** | **27** |
|  | **6.1.1** Admin Login andUser Login | 27 |
|  | 6.1.2 Searching the Keyword | 27 |
|  | 6.1.3 Key Generation | 28 |
|  | 6.1.4 View the Patient and User Details | 28 |
|  | **6.1.4 Decryption the Data** | **28** |
|  | **6.2 Algorithm** | **29** |
| **7.** | **TESTING** | **30** |
|  | 7.1 Software Testing | 31 |
|  | * + 1. Introduction | 31 |
|  | **7.2Validation &Verification** | **31** |
|  | **7.3 TYPES OF TESTING** | **32** |
|  | * + 1. Unit Testing | 32 |
|  | * + 1. Functional Testing | 33 |
|  | 7.3.3 System Testing | 33 |
|  | 7.3.4 White Box Testing | 33 |
|  | 7.3.5 Black Box Testing | 34 |
|  | 7.3.6 Integration Testing | 34 |
|  | 7.3.7 Validation | 34 |
| **8.** | **SOFTWARE DECRYPTION** | **35** |
|  | 8.1 Java Technology | 36 |
|  | 8.2 Java Programming Language | 36 |
|  | 8.3 Working of A Java Program | 37 |
|  | 8.4 Java Platform | 38 |
|  | 8.5 Components | 38 |
|  | 8.6 Implementation of Java VM | 39 |
|  | 8.6.1 ODBC | 40 |
|  | 8.6.2 JDBC | 41 |
| **9.** | **DOMAIN** | **42** |
|  | **9.1 My SQL** | **42** |
|  | 9.1 Introduction | 42 |
|  | 9.2 Overview | 42 |
|  | 9.3. History | 43 |
|  | 9.4. Features | 43 |
|  | 9.5. Limitations | 44 |
|  | 9.6. API | 45 |
|  | 9.6.1. Client | 46 |
|  | 9.6.2. Server | **46** |
|  | 9.7. My SQL | 46 |
|  | 9.8. Memory Allocations | 46 |
|  | 9.8.1. GLOBAL | 46 |
|  | 9.8.2. SESSION | 46 |
|  | 9.8.3 Connection/Thread Handling | 47 |
|  | 9.8.4 Parser | 47 |
|  | 9.9 Optimizer | 47 |
|  | 9.10.1 Query Cache | 47 |
|  | 9.10.2 Key Cache | 47 |
|  | 9.11 Storage engine | 48 |
| **10.** | **RESULTS AND DISCUTIONS** | **49** |
|  | 10.1 TEST RESULTS | 50 |
|  | 10.2 CONCLUSION | 50 |
|  | 10.3 FUTURE ENHANCEMENT | 51 |
| **11.** | **APPENDIX-I** |  |
| **12** | **APPENDIX-II** |  |
| **13** | **APPENDIX-II** |  |
| **14** | **REFERENCE** |  |