**Safe Access System**

BY

Samarth Patel

21C23502

A REPORT

SUBMITTED TO

ITM SLS Baroda University

in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

Computer Science & Engineering (with specialization in Cyber Security and Network

School of Computer Science Engineering & Technology

April 2024

**DECLARATION OF ORIGINALITY**

I declare that this report entitled “**Safe Access System**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or award.

|  |  |  |
| --- | --- | --- |
| Signature | : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
| Name | : | Samarth Patel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
| Date | : | 19th April, 2024\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

The co-team member of the project work include,

1. Meet Shah (20C23017)
2. Jaydeep Roy ()

**ACKNOWLEDGEMENTS**

I extend my heartfelt gratitude to my esteemed supervisor, Professor Gaurav Kulkarni, whose guidance and mentorship have illuminated my path in crafting the Safe Access System. His belief in my abilities has empowered me to delve into the realm of innovative security solutions.

A debt of appreciation is owed to the entire faculty and staff of ITM SLS Baroda University’s School of Computer Science Engineering & Technology. Their unwavering support and encouragement have been the cornerstone of this project, propelling me forward with zeal and determination.

I am also indebted to the users and testers who volunteered their time and feedback, helping refine and enhance the functionality of the Safe Access System.

**ABSTRACT**

The "Safe Access System" is an innovative security solution designed to enhance user authentication in digital environments. In this system, users undergo a multi-level password setup during registration, comprising conventional text-based passwords, image-based passwords, and graphical passwords.

During login, users are required to authenticate themselves by successfully passing through all three levels of password authentication. The system validates each level of password against stored data in the database, ensuring robust security measures.

The implementation of multiple authentication layers aims to fortify access control, safeguarding sensitive information from unauthorized access. Through this project, we endeavor to contribute to the advancement of secure access technology, paving the way for safer digital interactions in various domains.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **TITLE** | | i |
| **DECLARATION OF ORIGINALITY** | | ii |
| **ACKNOWLEDGEMENTS** | | iii |
| **ABSTRACT** | | iv |
| **TABLE OF CONTENTS** | | v |
| **LIST OF FIGURES** | | vi |
| **LIST OF ABBREVIATIONS** | | vii |
|  | |  |
| **CHAPTER 1 INTRODUCTION** | | 1 |
| 1-1 | Problem Statement and Motivation | 1 |
| 1-2 | Objectives | 1 |
| 1-3 | Project Scope and Direction | 2 |
| **CHAPTER 2 LITERATURE REVIEW** | | 7 |
| 2-1 | Methodology | 7 |
| 2-2 | Summary | 9 |
| **CHAPTER 3 SYSTEM REQUIREMENTS** | | 10 |
| 3-1 | Introduction | 10 |
| 3-2 | Software and Hardware Requirement | 11 |
| 3-3 | Summary | 13 |
| **CHAPTER 4 SYSTEM DESIGN** | | 14 |
| 4-1 | Introduction | 14 |
| 4-2 | Proposed System | 15 |
| 4-3 | Data Flow Diagram | 16 |
| 4-4 | Summary | 18 |
| **CHAPTER 5 IMPLEMENTATION** | | 19 |
| 5-1 | Introduction | 19 |
| 5-2 | System Design | 20 |
| 5-3 | Algorithm | 23 |
| 5-5 | Feature Extraction | 30 |
| 5-6 | Package/Libraries Used | 31 |
| **CHAPTER 6 SYSTEM TESTING** | | 32 |
| 6-1 | Introduction | 32 |
| 6-2 | Test Cases | 33 |
| 6-3 | Result | 35 |
| 6-4 | Performance Evaluation | 37 |
| 6-5 | Summary | 39 |
| **CONCLUSION** |  | 40 |
| **REFERENCES** |  | 41 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Title** | **Page** |
| Figure 2-1-F1 | IC Design Flow | 8 |
| Figure 2-2-F1 | Full-custom flow for a microprocessor | 10 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Forms** |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| PHP | HyperText Pre-Processor |
| HTTP | HyperText Transfer Protocol |
| MySQL | My Structured Query Language |
| UI | User Interface |
| DOM | Document Object Model |

**CHAPTER 1 INTRODUCTION**

**1-1 Problem Statement and Motivation**

The increasing reliance on digital platforms for various activities necessitates robust security measures to protect sensitive information from unauthorized access. However, traditional password-based authentication systems often fall short in providing adequate security against evolving cyber threats. Therefore, there is a pressing need to develop an advanced authentication system that offers multi-level security to ensure safe access to digital resources.

The proposed project aims to address this challenge by designing and implementing a "Safe Access System" that integrates three levels of password authentication: conventional text-based passwords, image-based passwords, and graphical passwords. During user registration, individuals will be prompted to set up passwords across these three levels, enhancing the complexity and strength of their authentication credentials.

Upon login, users will undergo a comprehensive authentication process, wherein each level of password will be verified against stored data in the system's database. This multi-layered approach to authentication seeks to fortify access control and mitigate the risk of unauthorized access to sensitive information.

By developing and deploying the "Safe Access System," this project endeavors to contribute to the advancement of secure access technology, addressing the critical need for enhanced security measures in digital environments.

**1-2 Objectives**

1. Develop a user-friendly interface for seamless registration and login processes within the Safe Access System.
2. Implement a secure database system to store user authentication credentials and ensure data confidentiality.
3. Design and integrate algorithms for generating and verifying conventional text-based passwords during registration and login.
4. Implement a graphical password system that allows users to create and authenticate passwords using graphical elements.
5. Establish a robust authentication framework that seamlessly integrates all three levels of password authentication (text-based, image-based, and graphical) during the login process.

**1-3 Project Scope and Direction**

The project scope encompasses the design, development, and implementation of the Safe Access System, an innovative multi-level authentication solution aimed at enhancing security in digital environments. The system will offer users the ability to set up and authenticate passwords across three levels: conventional text-based passwords, image-based passwords, and graphical passwords.

The project direction focuses on:

1. Prioritizing security: Ensuring that the Safe Access System provides robust protection against unauthorized access and cyber threats.
2. Enhancing user experience: Striving to create a seamless and user-friendly authentication process to encourage widespread adoption.
3. Adapting to evolving technologies: Remaining flexible to incorporate advancements in authentication methods and security protocols to ensure the system's relevance and effectiveness in the long term.

**CHAPTER 2 LITERATURE REVIEW**

2-1 Methodology

2-2 Summary

**CHAPTER 3 SYSTEM REQUIREMENTS**

**3-1 Introduction**

The successful implementation of the Safe Access System relies heavily on the utilization of various software tools and technologies.

These tools form the foundation upon which the system is built, facilitating the development of a robust and secure authentication solution.

In this section, we outline the software requirements essential for the creation and deployment of the Safe Access System, encompassing a range of programming languages, frameworks, and server environments.

**3-2 Software Requirement**

The development of the Safe Access System necessitates proficiency in a diverse set of programming languages and frameworks, including HTML, CSS, JavaScript, PHP, Python, and jQuery. These languages serve as the building blocks for creating the system's user interface, implementing functionality, and handling server-side operations.

Additionally, the system relies on the use of web servers such as the Apache HTTP Server to host and serve web pages, ensuring seamless communication between clients and the system's backend. Furthermore, the deployment environment is based on the Windows Server operating system, providing a stable and secure platform for hosting the Safe Access System.

To streamline database management tasks, the system leverages phpMyAdmin, a web-based administration tool for MySQL databases. This enables efficient storage and retrieval of user authentication credentials, ensuring data integrity and confidentiality.

Moreover, the inclusion of UI frameworks such as Bootstrap facilitates the creation of responsive and visually appealing user interfaces, enhancing the overall user experience of the Safe Access System.

**3-3 Summary**

In conclusion, the software requirements outlined above form the essential toolkit for the development, deployment, and maintenance of the Safe Access System.

By leveraging a combination of programming languages, frameworks, web servers, and operating systems, we are equipped to build a secure and user-friendly authentication solution tailored to the needs of modern digital environments.

Moving forward, adherence to these software requirements will be crucial in ensuring the successful implementation and ongoing functionality of the Safe Access System, as it continues to provide a safe and reliable means of accessing digital resources.

**CHAPTER 4 SYSTEM DESIGN**

**4-1 Introduction**

The design phase of the Safe Access System is a critical stage where the blueprint for the system’s architecture and functionality is meticulously crafted. It involves a systematic approach to conceptualizing the system’s structure, defining its components, and delineating the flow of data and interactions between various modules. In this section, we delve into the intricacies of the system design process, shedding light on the proposed system architecture, data flow diagram, and key design considerations.

**4-2 Proposed System**

The Safe Access System represents an innovative multi-level authentication solution aimed at enhancing security and user experience in digital environments. At its core, the system incorporates three distinct levels of password authentication: conventional text-based passwords, image-based passwords, and graphical passwords. Users engage with the system through an intuitive and user-friendly interface, where they can seamlessly register and log in using their preferred authentication methods.

The architecture of the system adheres to the client-server model, with client-side components developed using HTML, CSS, and JavaScript, while server-side functionality is implemented using PHP and Python. For data storage and management, the system relies on a robust relational database management system (MySQL), complemented by the user-friendly phpMyAdmin administration tool for efficient database management tasks.

**4-3 Data Flow Diagram**

INSERT DIAGRAM

**4-4 Summary**

In summary, the system design phase serves as the cornerstone for the development and implementation of the Safe Access System. By meticulously defining the system architecture, delineating the flow of data, and outlining interaction patterns, we ensure that the authentication solution is not only robust and scalable but also intuitive and user-friendly. The design considerations outlined in this section will serve as guiding principles throughout the implementation process, facilitating the successful realization of the Safe Access System's objectives in securing digital access environments.

**CHAPTER 5 IMPLEMENTATION**

**5-1 Introduction**

The implementation of the Safe Access System involves several key components, including user interface design, database management, password generation and verification algorithms, and the development of a robust authentication framework. Below, we provide an overview of the implementation process, including sample code snippets and algorithm descriptions.

**5-2 System Design**

[each protocol diagram here]

**5-3 Algorithm**

**Text based authentication (Login)**

* Begins with session initialization and includes a connection to a database (not provided in the code snippet).
* Defines several functions used for password hashing, decryption, and key generation.
* Checks if the login form has been submitted using ⁠ isset($\_POST["submit"]) ⁠.
* Retrieves the entered username, passphrase, and password using ⁠ $\_POST ⁠.
* Validates that all fields are not empty.
* Queries the database to retrieve stored salt and hashed password associated with the entered username.
* Decrypts the stored salt using the passphrase provided by the user.
* Iteratively hashes the provided password with the decrypted salt.
* Compares the hashed password obtained with the stored hashed password to authenticate the user.
* If authentication is successful, redirects the user to a new page (e.g., ⁠ loggraphical.php ⁠).
* If authentication fails, displays an alert message indicating "User not found."

**Text based Authentication (Sign Up)**

* Begins with session initialization and includes a connection to a database (not provided in the code snippet).
* Defines several functions used for password hashing, salt generation, encryption, and key generation.
* Checks if the sign-up form has been submitted using ⁠ isset($\_POST["submit1"]) ⁠.
* Retrieves the entered username, email, passphrase, and password using ⁠ $\_POST ⁠.
* Validates that all fields are not empty.
* Queries the database to check if the entered UserID already exists.
* If the UserID does not exist, proceeds with the sign-up process:
* Generates a unique salt for password hashing using ⁠ generateSalt() ⁠.
* Hashes the provided password with the generated salt using ⁠ hashPassword() ⁠.
* Encrypts the generated salt using the passphrase provided by the user.
* Inserts the user's details (username, email, hashed password, encrypted salt) into the database.
* Redirects the user to a new page upon successful registration.
* If the UserID already exists, displays an alert message indicating "UserID already exists. Please choose a different UserID."

**Pattern based Authentication**

* Start the session to store user data.
* Retrieve session variables for the username and passphrase if they exist.
* Define the HTML structure for the sign-up page, including the title, stylesheets, and background video.
* Create a text container for the page title.
* Define a container for the graphical authentication buttons and input field.
* Attach click event handlers to three buttons (⁠ #red\_box ⁠, ⁠ #blue\_box ⁠, ⁠ #green\_box ⁠).
* When clicked, each button appends a specific pattern to the input field (⁠ #input ⁠).
* Define a form with a password input field and a submit button.
* When the submit button is clicked, the form data is sent to the server using POST method to ⁠ $\_SERVER["PHP\_SELF"] ⁠.
* Include jQuery library for DOM manipulation.
* Implement click event handlers for the graphical authentication buttons to append patterns to the input field.
* Include a PHP block to handle form submission (⁠ isset($\_POST["submit"]) ⁠).
* Retrieve the graphical password entered by the user from ⁠ $\_POST['graph'] ⁠.
* Validate that the graphical password is not empty.
* Retrieve the stored salt and graphical password hash from the database based on the logged-in user (⁠ $loguname ⁠).
* Decrypt the stored salt using the passphrase (⁠ $key1 ⁠) provided during login.
* Generate a new salt for the graphical password and append it to the stored salt.
* Hash the graphical password using the combined salt.
* Encrypt the combined salt and update the database with the new graphical password.
* Display a success message and redirect the user to the next authentication level upon successful update.
* Close the database connection.

**Image based authentication**

* The HTML defines the structure of the sign-up page.
* It includes elements for displaying the background video, title, image, and input fields for selecting an image and entering a pattern.
* The `loadImage()` function is triggered when a user selects an image file. It displays the selected image on the page.
* The `image\_crop()` function is invoked when the "Crop" button is clicked. It crops and displays the selected image into nine smaller images in a 3x3 grid format.
* Event handlers are attached to each of the smaller images. When clicked, they append specific patterns to the password input field (`#input`) and change the opacity of the clicked image to indicate selection.
* When the "Save" button is clicked, the form data (including the selected image pattern) is submitted to the server using the POST method to the same PHP script (`$\_SERVER["PHP\_SELF"]`).
* The PHP script processes the form data upon submission.
* The PHP script retrieves session variables (`$id` and `$key`) for user identification and decryption.
* It defines functions for generating salt, hashing passwords, and encrypting data.
* Upon form submission (`isset($\_POST["submit"])`), it retrieves the user's stored salt from the database.
* It generates a new salt for the picture password and hashes the password using the combined salt.
* The combined salt is encrypted using the user's passphrase (`$key`) and stored in the database along with the hashed password.
* Upon successful update, a success message is displayed, and the user is redirected to the index page.

**5-4 Package/Libraries Used**

To enhance the user interface, the Safe Access System leverages several libraries and packages, including:

JavaScript Library - jQuery 3.6.4:

jQuery is a fast, small, and feature-rich JavaScript library that simplifies HTML document traversal and manipulation, event handling, and animation. It provides a convenient way to interact with the Document Object Model enabling developers to streamline the development process and enhance the functionality of web applications.

Example of including jQuery library in HTML:

<scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>

CSS Library - all.min.css:

The all.min.css library is a comprehensive collection of pre-built CSS stylesheets, including popular frameworks like Bootstrap and custom styles for common UI elements. It offers a convenient way to apply consistent styling across web pages, reducing development time and ensuring a polished user interface.

Example of including all.min.css in HTML:

<link rel="stylesheet" href="path/to/all.min.css">

Google Fonts (fonts.googleapis.com):

Google Fonts is a vast collection of free, open-source fonts that can be easily integrated into web projects. By using Google Fonts, developers can enhance the typography of their web applications, improving readability and visual appeal.

Example of including Google Fonts in CSS:

@import url('https://fonts.googleapis.com/css?family=Roboto');

By leveraging these libraries and packages, the Safe Access System is equipped with the necessary tools to streamline feature extraction and enhance the user interface, ultimately contributing to a seamless and intuitive authentication experience for users.

**CHAPTER 6 SYSTEM TESTING**

**6-1 Introduction**

System testing is a critical phase in the development lifecycle of the Safe Access System, ensuring that the implemented solution meets the specified requirements and functions as intended. This phase involves the execution of various test cases to evaluate the system's performance, reliability, and security features. In this section, we outline the test cases designed to assess the effectiveness of the Safe Access System in securing user data and mitigating common security threats.

**6-2 Test Cases**

[pending]

**6-3 Result**

[pending]

**6-4 Summary**

In summary, the system testing phase confirms the effectiveness of the Safe Access System in securing user data and mitigating common security threats. Through comprehensive test cases, the system demonstrates its ability to prevent brute force attacks, SQL injection, and directory traversal, thereby ensuring the confidentiality, integrity, and availability of user information. Moving forward, ongoing testing and refinement will be necessary to maintain the system's security posture and adapt to evolving cyber threats.

**CONCLUSION**

In conclusion, the development and implementation of the Safe Access System represent a significant advancement in the realm of secure access technology. Through the integration of multi-level password authentication, including conventional text-based passwords, image-based passwords, and graphical passwords, the system offers enhanced security measures while prioritizing user convenience and usability.

The implementation process involved the design and development of various components, including the user interface, database management system, password generation and verification algorithms, and the authentication framework. By leveraging technologies such as HTML, CSS, JavaScript, Node.js, and database systems like MySQL or MongoDB, we were able to create a robust and scalable authentication solution.

Throughout the implementation process, careful consideration was given to security best practices, ensuring that user authentication credentials are securely stored and processed. The use of bcrypt or similar hashing algorithms for text-based password hashing, along with the development of secure image and graphical password verification mechanisms, contributes to the overall strength and reliability of the system.

Moving forward, further testing and refinement of the Safe Access System will be necessary to ensure its effectiveness and reliability in real-world scenarios. Additionally, ongoing research and development efforts will be essential to keep pace with evolving security threats and technological advancements, ensuring that the system remains at the forefront of secure access technology.

**REFERENCES**

[1] Rohi tkumar Kolay, Animesh Vora, Vinaykumar Yadav, “Graphical Password Authentication Using Image Segmentation” , e-ISSN: 2395-0056, p-ISSN:2395-0072, Mar-2017

[2] Dr.K.Gowsic, K.Manojekumar, R.Mithunraj, K.Pravin, R.Sanjaykumar, “Secure Login Using

One Time Authentication Password System”, Volume 10, ISSN: 2320-2882, Volume 10, Issue 6

June-2022

[3] Ankesh Khandelwal, Shashank Singh, Niraj Satnalika, “User Authentication by Secured Graphical Password Implementation”, Volume 1 – No 25

[4] Prof. Akansha Bondade, Mr. Suyash Janbandhu, Mr. Bhushan Mungantiwar, Mr. Piyush Dhunde, Mr. Sumit Hajare, Mr. Suraj Solanke, “Three Level Security System Using Image Based Authentication” , ISSN: 2395-4396, Vol-8 Issue-3 2022

[5] Charanjeet Singh, Dr. Tripat Deep Singh, “A 3-Level Multifactor Authentication Scheme For Cloud Computing” , ISSN Print: 0976-6367, ISSN Online: 0976-6375, January-February 2019

[6] N Chaitra, Pratibha, Dr. Rajashree V Biradar, “Three Level Password Authentication System” , e-ISSN: 2580-5208, July-2022

[7] Ragini Sharma, Dr. Umarani Chellapandy, “Three Level Password Authentication System Mechanism”, ISSN: 2321-9653, Apr-2022

[8] A Balamurli, M V R Harsha, V Sai Hitesh, A Sai Chaitanya, “Graphical Password by Image

Segmentation”, ISSN: 2278-3075, Volume-8, Issue-6S4, April 2019