

B.Sc. in Computer Science and Engineering Thesis

# **Image Based Password Authentication System**

Submitted by

Sheikh Ashraf Ali

201414008

Sanjida Akter Sharna

201414021

Md. Abdullah Al Mamun

201414037

Supervised by

Colonel Nishith Kumar Datta

Professor

Department Of Computer Science & Engineering

Military Institute of Science & Technology



**Department of Computer Science and Engineering**  
**Military Institute of Science and Technology**

December 2017

# **CERTIFICATION**

This thesis paper titled "**Image Based Password Authentication System**", submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in January 2018.

## **Group Members:**

1. Sheikh Ashraf Ali
2. Sanjida Akter Sharna
3. Md. Abdullah Al Mamun

## **Supervisor:**

---

**Colonel Nishith Kumar Datta**

Professor, CSE Department

Military Institute of Science & Technology

# CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper, titled, "**Image Based Password Authentication System**", is the outcome of the investigation and research carried out by the following students under the supervision of Colonel Nishith Kumar Datta, Professor, CSE Department, Military Institute of Science & Technology.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

---

Sheikh Ashraf Ali  
201414008

---

Sanjida Akter Sharna  
201414021

---

Md. Abdullah Al Mamun  
201414037

# ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, **Colonel Nishith Kumar Datta**, Professor, CSE Department, Military Institute of Science & Technology, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka

December 2017 .

1. Sheikh Ashraf Ali
2. Sanjida Akter Sharna
3. MD. Abdullah Al Mamun

# ABSTRACT

Preservation of information and computer security is broadly dependent on the secured authentication system which is underpinned by password. Text based password is a commonly used and available system for authentication. But it bears many limitations like shoulder surfing, dictionary attack, guessing the password by using various permutation and combination of alphanumeric numbers, brute force attack etc. In order to overwhelm this vulnerabilities of ancient textual password many graphical or image based password authentication system has been introduced form last few years. But none of this graphical system is considered as enough adventurous to keep pace with this issues. Here we have proposed an image based authentication system which is more efficient and can cope up with every vulnerabilities of recent password authentication system. In our system we are only allowing user to provide user name for registration as our system will generate a unique key number for user and this key will be used for regarding login procedure. The user name and key will be encrypted through AES algorithm and saved in a file to prevent database hacking. There will be a random image grid in our system which will be used for login purpose. A user doesnt need to enter any textual password for authentication in our recent module and hence combination of all this features improve the security, usability and user friendliness of our system.

# Contents

<i><b>CERTIFICATION</b></i>	<b>ii</b>
<i><b>DECLARATION</b></i>	<b>iii</b>
<i><b>ACKNOWLEDGEMENT</b></i>	<b>iv</b>
<i><b>ABSTRACT</b></i>	<b>1</b>
<b>1 Introduction</b>	<b>8</b>
1.1 Overview . . . . .	8
1.2 Objective . . . . .	8
1.3 Thesis Organization . . . . .	9
<b>2 Password scheme</b>	<b>10</b>
2.1 Textual password . . . . .	10
2.2 System Assigned password . . . . .	10
2.3 Graphical Password . . . . .	11
<b>3 Literature review</b>	<b>12</b>
3.1 Prior works . . . . .	12
3.1.1 A Shoulder surfing resistant graphical password scheme- WIW . . .	12
3.1.2 A shoulder surfing resistant image based authentication system with temporal indirect image selection . . . . .	13
3.1.3 A Review on Two Level Authentication Using Image Selection and Voice Recognition . . . . .	13

3.1.4	Image based authentication system . . . . .	15
3.1.5	A Secure Graphical Password Authentication System . . . . .	16
3.2	Observation . . . . .	16
<b>4</b>	<b>Paramilitary Ideas</b>	<b>17</b>
4.1	Textual Password . . . . .	17
4.2	Graphical Password . . . . .	17
4.3	Password encryption . . . . .	17
4.4	Password description . . . . .	17
4.5	Strong Password . . . . .	18
4.6	Password strength . . . . .	18
4.7	Eidetic memory . . . . .	18
4.8	Hamming distance of images . . . . .	18
4.9	Plain text . . . . .	18
4.10	Plain text . . . . .	19
4.11	Cipher text . . . . .	19
4.11.1	Substitution Cipher . . . . .	19
4.11.2	Poly alphabetic Substitution Cipher . . . . .	19
4.11.3	Transposition Cipher . . . . .	19
4.11.4	Permutation Cipher . . . . .	19
4.11.5	Private-key Cryptography . . . . .	19
4.11.6	Public-key Cryptography . . . . .	20
4.12	steg image . . . . .	20
4.13	Cryptography . . . . .	20
4.14	AES . . . . .	20
4.15	Image grid . . . . .	20
4.16	Authentication . . . . .	20
4.17	Pass image/ pass object . . . . .	20

4.18	shoulder surfing . . . . .	21
<b>5</b>	<b>Security Discussion</b>	<b>22</b>
5.1	Shoulder Surfing . . . . .	22
5.2	Password Guessing . . . . .	22
5.3	Dictionary Attack . . . . .	22
5.4	Password Cracking . . . . .	22
5.5	Hash guessing . . . . .	23
5.6	Rainbow Table . . . . .	23
5.7	Password Sniffing . . . . .	23
5.8	Brute Force Attack . . . . .	23
5.9	Hybrid password guess . . . . .	24
5.10	Password Resetting . . . . .	24
<b>6</b>	<b>Steganographic Image Generation</b>	<b>25</b>
6.1	Encryption . . . . .	25
6.1.1	AES . . . . .	25
6.1.2	Byte jumping series . . . . .	25
6.1.3	Zero case series . . . . .	26
6.2	Decryption . . . . .	26
<b>7</b>	<b>Methodology</b>	<b>27</b>
7.1	Proposed system . . . . .	27
7.2	Registration phase . . . . .	27
7.2.1	Key Number . . . . .	27
7.2.2	AES encryption . . . . .	27
7.3	Log in phase . . . . .	28
7.3.1	Enter user name . . . . .	28
7.3.2	10*10 image grid . . . . .	28



7.3.3	Shoulder surfing resistance password . . . . .	28
7.4	Problems Faced And Encountered . . . . .	30
<b>8</b>	<b>System Design</b>	<b>32</b>
8.1	System Architecture . . . . .	32
8.1.1	Registration system architecture . . . . .	32
8.1.2	Login system architecture . . . . .	33
<b>9</b>	<b>Implementation</b>	<b>34</b>
9.1	Java . . . . .	34
9.2	Java swing . . . . .	34
9.3	MATLAB . . . . .	34
<b>10</b>	<b>Result And Analysis</b>	<b>35</b>
10.1	Result . . . . .	35
10.1.1	Screen shot of registration phase . . . . .	35
10.2	Login phase . . . . .	37
10.3	Analysis . . . . .	41
<b>11</b>	<b>Future Work</b>	<b>42</b>
11.1	Stego image . . . . .	42
11.1.1	Replace username for login . . . . .	42
11.1.2	Use pen drive to take username from stago image . . . . .	42
11.2	AES key randomization . . . . .	43
11.3	Prevent brute force attack . . . . .	43
11.4	Transform the whole system into prototype . . . . .	43
<b>12</b>	<b>Advantage and Limitation</b>	<b>44</b>
12.1	Advantage . . . . .	44
12.2	Limitation . . . . .	44

<b>13 Conclusion</b>	<b>45</b>
13.1 Contribution . . . . .	45
13.2 Future work . . . . .	45
<b>References</b>	<b>46</b>
<i>APPENDIX A</i>	<b>47</b>
<i>APPENDIX B</i>	<b>50</b>
<i>APPENDIX C</i>	<b>53</b>
<i>APPENDIX D</i>	<b>54</b>

# List of Figures

3.1	Working procedure of shoulder surfing resistant image based authentication system with temporal indirect image selection . . . . .	13
3.2	Work flow two level authentication using image selection and voice recognition . . . . .	14
7.1	10x10 image grid . . . . .	29
8.1	Block Diagram Of Registration System Architecture . . . . .	32
8.2	Block Diagram Of Log In System Architecture . . . . .	33
10.1	Registration module . . . . .	35
10.2	Registration procedure . . . . .	36
10.3	AES encrypted file . . . . .	37
10.4	Login Module ( system ask for username ) . . . . .	38
10.5	Clickable Image grid( system ask to enter key number by clicking this grid)	39
10.6	Login Module ( system shows that login is successful) . . . . .	40

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

Authentication system is of great importance from the raise of information and technology for the confidentiality of data, information and many other things of individuals or any organizations. From the ancient era textual based authentication system has been used for this purpose. But now with the spread of technology and advancement hackers have become smart enough to break to any type of textual password. They have found many ways like password guessing, dictionary attack, hash guessing, rainbow table, password sniffing ,brute force attack and many other efficient way to break any strong password. For this reasons the information and data under this type of authentication system are at great risk of disclosing. To get rid of this problem graphical authentication system has been introduced a few years ago. There are many types and categories of graphical authentication systems which has been launched to preserve confidentiality of information. But none of this systems are enough efficient to protect data fully. Now a days shoulder surfing is the main obstacle to this graphical authentication system. If any observer , observe the graphical login system for some time then he could probably guess the pattern or type of graphical password which is the main failure of this authentication system . Recalling all this obstacles we have design such a system which is shoulder surfing registrant and not textual. Our system will provide user a key number which is required for login. While login user will provide the user name and the key number will be taken using special technique .It will be taken by clicking the separate images in a 10\*10 image grid. The user can also make his own key number by adding the current date to it which will be discussed broadly in methodology section.

### 1.2 Objective

Thought out the authentication system, the main objective is to develop such a way which is not textual so that no hackers can easily break any type of password using the latest technique of password breaking. Another prime objective is that to get rid of the failure( shoulder surfing) of graphical authentication system. Comparing with others system we have reached to the conclusion that our system is almost completely shoulder surfing resistant and hence

achieved our primary goal. Further we will use stenography tool to make our system more strong and two factored which is our secondary goal.

### **1.3 Thesis Organization**

Chapter 2 will fall light upon password scheme Chapter 3 will demonstrate about literature review of prior works Chapter 4 will describe about preliminary ideas Chapter 5 will describe about security discussion Chapter 6 will describe about Stenographic Image generation

Chapter 7 will describe about Methodology in detail

Chapter 8 will describe about System Design

Chapter 9 will describe about Implementation and the platform

Chapter 10 will describe about result and analysis Chapter 11 will describe about scope of future work Chapter 12 will describe about advantage and limitation Chapter 13 will conclude the whole work with future work

## **CHAPTER 2**

### **PASSWORD SCHEME**

#### **2.1 Textual password**

A password is a word or string of characters used for user authentication to prove identity or access approval to gain access to a resource (example: an access code is a type of password), which is to be kept secret from those not allowed access . [6] An ideal password is easy to memorize but difficult to hack by hackers through various password cracking tools like brute force attack, password guessing etc. From ancient time text based password have been used for authentication purposes. Now along with password, user name, DOB various details are required. User use password for various authentication like ATM, e-mail, accessing application, computer or database etc. Recently some surveys show that user choose very poor password format like they use only alphabetic letters or numerical numbers. It makes the password strength very low and poor. It is also noticed that user make their close friends or relative or special person name their password and they do not use any numerical number or special character. It is very easy to break. In some survey it is shown that people use the password as their password. They often disclose their password with their friend, family or use same password for multiple sector which creates many vulnerabilities. In the process of generating a password user either neglects the security issue or neglect their ability to memorize.

#### **2.2 System Assigned password**

In this scheme, the main idea is this the password is generated in a arbitrary way . They may be anything like arbitrary words, pass phrases or any pronounceable password which can be uttered easily . A random password may be any word or any random string which is difficult to remember and do not make any sense . To make it remember able the concept of pass phrase is introduced. A pass phrase is a sequence of words or other text used to control access to a computer system, program or data. A pass phrase is similar to a password in usage, but is generally longer for added security. Pass phrases are often used to control both access to, and operation of, cryptography programs and systems, especially those that derive an encryption key from a pass phrase. The origin of the term is by analogy with password.

The modern concept of pass phrases is believed to have been invented by Sigmund N. Porter in 1982. [7] In this password system, words are arbitrarily selected from English dictionary set. But in some studies it is found that user do not like this kind of system assigned or pass phrase passwords.

## **2.3 Graphical Password**

A graphical password is an authentication system that works by having the user select from images, in a specific order, presented in a graphical user interface (GUI). For this reason, the graphical-password approach is sometimes called graphical user authentication (GUA). A graphical password is easier than a text-based password for most people to remember. [8] Graphical password scheme is divided into four main sections. They are 1. recognition based 2. recall-based 3. cued-recall 4. cued-recognition The main concept of recognition based password scheme is that, user needs to recognize all the image sequence while login which he/she had chosen previously for registration purpose. In recall based scheme, user must be able to recall the password by thyself without any help or clue. It is tough for user to recall such password. So recognition based password is easier than recall based in comparison. In cued-recall scheme it can be described in such a way that, user provided personal information and details are helped to recall or memorize. It is better in sense that, it ensures decent memorability and usability than recall based password scheme. In cued-recognition scheme, various cues like words, visual, scenes are provided to user to make it easy for them to recognize. User can choose the cues to recognize the password which he thinks are alike to his password .

## CHAPTER 3

### LITERATURE REVIEW

#### 3.1 Prior works

##### 3.1.1 A Shoulder surfing resistant graphical password scheme- WIW

Recently a shoulder resistant scheme was introduced where they describe their system as follows: In this system, they mainly used four factors. A user need to choose  $h$  number of images to make his password. it is variable depending on each user. They can choose it according to their wish. While log in the content of this  $h$  images will change randomly. This images are said scenes here For each scene there will be  $n$  objects which is fix ranging from 250-300. Then  $k$  is defined as pass-objects which is only chosen by the user and it is his part of password. Then another factor  $m$  is defined as perturbation i.e the appearance of each pass object in each scene. While log in the user must be able to recognize those pass objects. One image is stand for one letter, which is determined by the system. Now user need to identify the appearances and location of pass objects in each scenes. This combination spells a letter for each pass object. This way user need to identify all the combination which spells some letter. Its the main task to identify all the letters for successful log in. Each time of log in this pass objects and non pass objects and their appearances will be randomized. For the attacker it is difficult to identify those letters and recognize the pattern as each time it is randomized. The main problem with this system is that, if the value of  $h$ ,  $k$  and  $m$  is smaller than its quite easy for a user to memorize those factors but then the attackers may recognize the pattern while studying for some days .its weakens the system. If the user choose the value  $h$ ,  $k$  and  $m$  bigger, then while log in the user need to memorize lots of factors like all the pass objects form each scene accurately. In case the user failed to choose the pass objects correctly then log in process will b unsuccessful. So here the whole system is not at all user friendly or not usable for ordinary users. In our system the user has to face no difficulties to form his password, he just need to provide his user name which is easy for him to memorize. For the user our system will generate a key number and provide him, as its generated by the system ,its highly secured. Then while log in the user just need to enter the key number from image grid. [1].



### 3.1.2 A shoulder surfing resistant image based authentication system with temporal indirect image selection

In this system it displays images in sequence like slide show and an user need to select slide show which includes his pass image among several slides show. The system presents several images to users. user select several pass images  $p$ , which is all to gather used as his password. And the user must remember this pass images. In login phase the system present  $N$  slide shows and  $M$  is the number of images in each slide show. Now the user need to recognize in which slide show the pass image is included, then he need to select the exact pass image from each slide show which changes with a fix interval of time  $1/v$  per image. By choosing the correct sequence the login will be successful. [2].

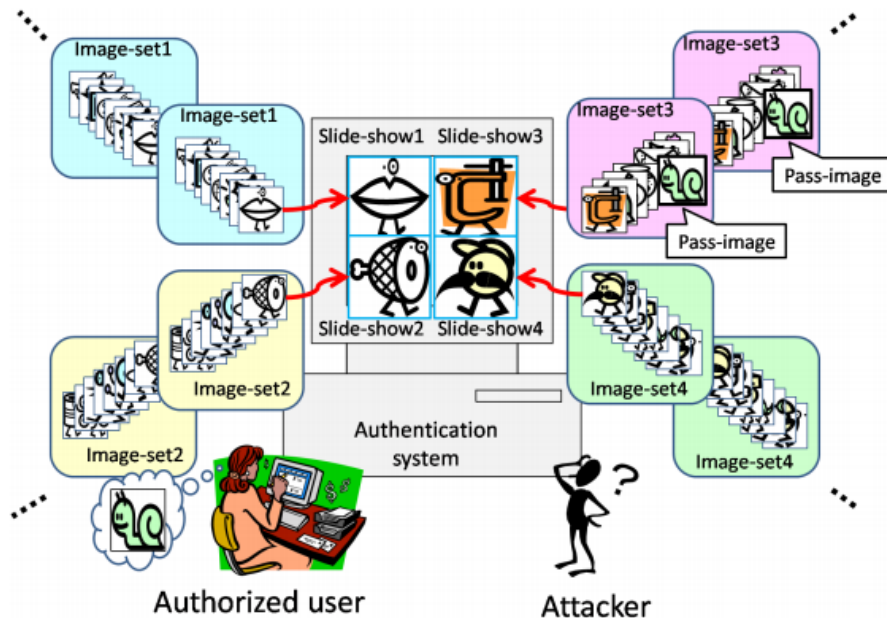


Figure 3.1: Working procedure of shoulder surfing resistant image based authentication system with temporal indirect image selection

### 3.1.3 A Review on Two Level Authentication Using Image Selection and Voice Recognition

In this system image and voice is used for authentication. During registration user has to select a image and enter a watermark. This image with the watermark is one level of authentication. Then user needs to give the voice summon which is the 2nd level of authentication. In this system they used voice input which is not very much user friendly. Different voice may occur during recognition.

Our system is very much user friendly. We just need to remember a 4 digit code to log in.

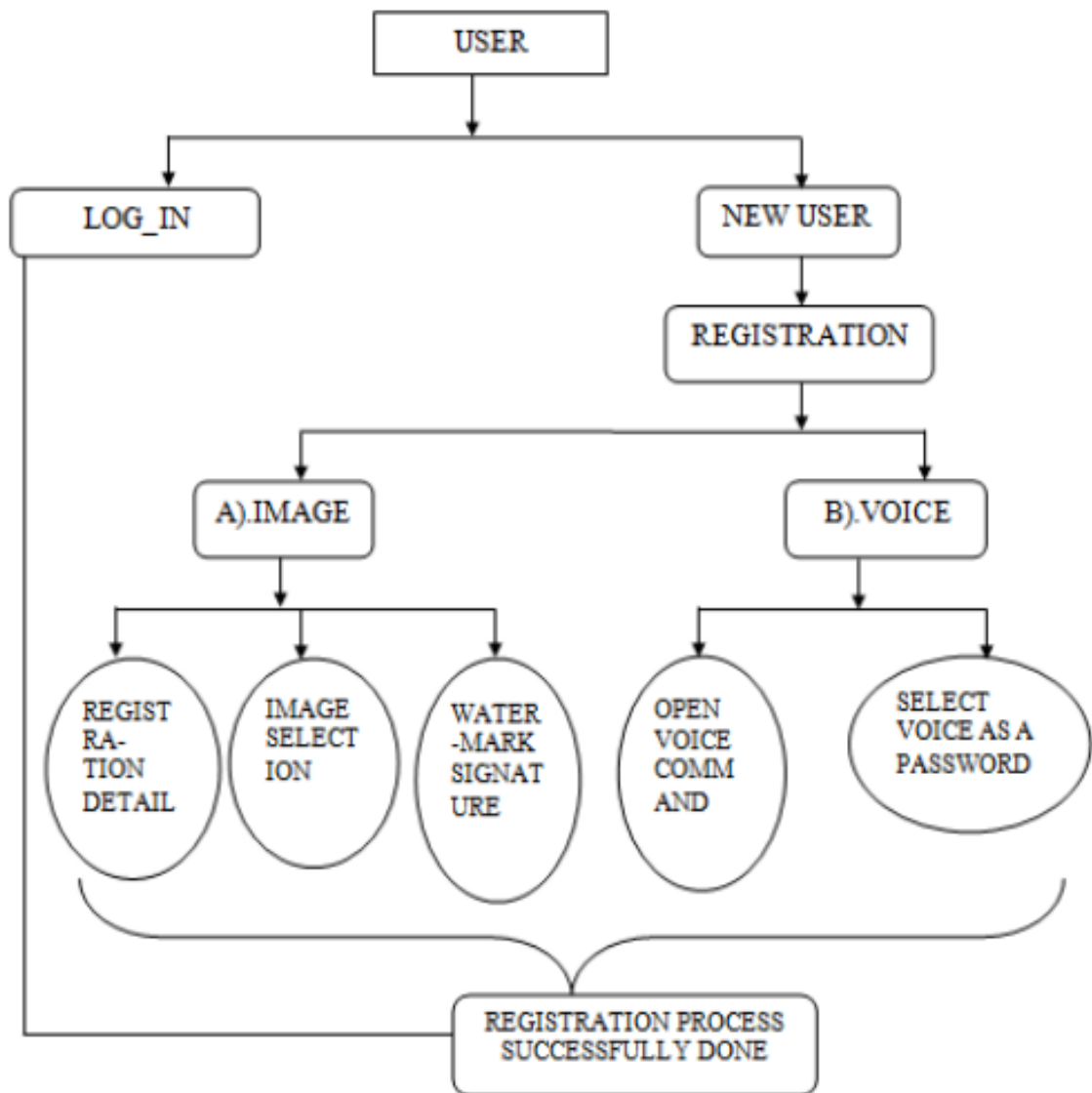


Figure 3.2: Work flow two level authentication using image selection and voice recognition

In this system there is a high possibility of shoulder surfing while selecting the image. User need to select just one image to log in. So it is easy for attacker to know the required image by shoulder surfing. Our system is highly preventable form shoulder surfing problem. We used a random image grid and a shoulder surfing resistant key which is described previously to prevent this problem. [3]

#### **3.1.4 Image based authentication system**

In this system, both password and image are used to overcome problems. Like other authentication system, it also consists of two phases that is registration and log in. In registration phase user need to fulfill basic personal requirements like name, DOB, e-mail address etc. Next user need to enter a password which must maintain their constraint like minimum 8 character password, minimum one uppercase and one numerical number and one special character . By full filling this criteria, user can choose their desired password. After this stage user must select one image as his pass image otherwise the registration will be unsuccessful. There is a image category selection option, from where user need to select his desired category like natural scenarios, animals, flowers etc. Every time this categories will be randomized by refreshing the page. Now if user select 3 images, then he/she need to select the category, then from his chosen category a 3\*3 image grid of same type but discrete images will appear. From this grid user will choose some of his pass images. Then he need to choose his 2nd category. By choosing 2nd category again 3\*3 grid of this category will appear. Again user need to choose pass image from this category. In the same manner user need to choose the 3rd category and pass images. User must memorize this pass images. In log in phase, user must provide user name, password and pass images. Pass images will be randomly displayed in log in round. In each log in round there may be all pass images or some images. It is also possible that none of pass images appear in any round. User need to choose the pass images as the exact sequence he choose them through registration phase. By choosing the correct sequence from log in rounds, log in will be successful.

In their system they cannot prevent offline dictionary attack and the system is slower comparing traditional textual authentication system for displaying image grid several time. They do not use any encryption system to encrypt the password. If a user hack the database then he can easily get the password and user name and it can make one step easier for the hacker to hack the system. And in their system shoulder surfing is prevented but not in fully manner. Comparing with this system, our system almost prevent offline dictionary attack and database hack .As in our system the user name and password which is a key number generated by our system) will be encrypted through AES and will be save in database. For this if hacker hack the database they wont get the exact password. As they dont know the key, they cannot decrypt it .Our image grid appear only once, and randomize in every click.

For this reason it is faster than their system. And our for advanced and randomized image grid, the possibilities of shoulder surfing is almost prevented which is explained broadly in methodology section. [4] .

### **3.1.5 A Secure Graphical Password Authentication System**

This system allows the user to create a graphical password by first selecting an image from a collection of available pictures . In the selected image user has to select one grid as the password. The selected image is watermarked with a cover image using Generic Visible Watermark Embedding technique. The method is based on the use of deterministic one-to-one compound mappings of image pixel values for overlaying a variety of visible watermarks of arbitrary sizes on cover images. During login, after entering the user details a QR Code is generated in the computer. User has to scan the QR code using his mobile phone. After scanning, a collection of images will be appeared in the screen of the phone. User has to select the image. After choosing correct image, the watermarked image will be appeared on the screen. User has to choose the correct grid position that he has already registered in the watermarked image. [5]

## **3.2 Observation**

Studying several papers we have reached to the conclusion that they have presented different graphical authentication system but neither of them is enough beneficial to fight against the recent problems like shoulder surfing, guessing the pattern etc. Comparing with this systems, our system is much more efficient to secure data and information and fulfill the requirement of secure authentication system.

# **CHAPTER 4**

## **PARAMILITARY IDEAS**

### **4.1 Textual Password**

Textual password is a password system where user use alphanumeric and special character as password. This password is stored in the database in general form or in encrypted form. Different system use different scheme for this password. Such as password should be at least contains 8 characters with at least one upper case and at least one digit.

### **4.2 Graphical Password**

A graphical password is an authentication system that works by having the user select from images, in a specific order, presented in a graphical user interface (GUI). For this reason, the graphical-password approach is sometimes called graphical user authentication (GUA).

### **4.3 Password encryption**

The translation of data into a secret code. Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called plain text ; encrypted data is referred to as cipher text.

There are two main types of encryption: asymmetric encryption (also called public-key encryption) and symmetric encryption.

### **4.4 Password description**

Decryption is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form. In decryption, the system extracts and converts the garbled data and transforms it to texts and images that are easily understandable not only by the reader but also by the system. Decryption may be accomplished manually or

automatically. It may also be performed with a set of keys or passwords.

## **4.5 Strong Password**

A strong password consists of at least six characters (and the more characters, the stronger the password) that are a combination of letters, numbers and symbols if allowed. Passwords are typically case-sensitive, so a strong password contains letters in both uppercase and lowercase.

## **4.6 Password strength**

Password strength is a measure of the effectiveness of a password against guessing or brute-force attacks. In its usual form, it estimates how many trials an attacker who does not have direct access to the password would need, on average, to guess it correctly. The strength of a password is a function of length, complexity, and unpredictability.

## **4.7 Eidetic memory**

It is a special type of memory. It is an ability of someone vastly to recall images one by one without using any type of mnemonic device with just a few instances of manifestation. A person has an almost faithful mental image snapshot or photograph of an event in their memory. However, eidetic memory is not limited to visual aspects of memory and includes auditory memories as well as various sensory aspects across a range of stimuli associated with a visual image.

## **4.8 Hamming distance of images**

Using of similar types, genres, and categories images which are very difficult to differentiate with one another to puzzle anyone. The less hamming distance the images are more vulnerable to differentiate.

## **4.9 Plain text**

In computing, plain text is the data (e.g. file contents) that represent only characters of readable material but not its graphical representation nor other objects (images, etc.). It may

also include a limited number of characters that control simple arrangement of text, such as line breaks or tabulation characters. Plain text is different from formatted text. According to The Unicode Standard Plain text is a pure sequence of character codes; plain Ue-encoded text is therefore a sequence of Unicode character codes. Styled text, also known as rich text, is any text representation containing plain text completed by information such as a language identifier, font size, color, hypertext links.

## **4.10 Plain text**

## **4.11 Cipher text**

### **4.11.1 Substitution Cipher**

This offers an alternative to the plain text. It is also known as Caesar cipher.

### **4.11.2 Poly alphabetic Substitution Cipher**

In this cipher, a mixed alphabet is used to encrypt the plain text, but at random points it would change to a different mixed alphabet which indicates the change with an uppercase letter in the Cipher text.

### **4.11.3 Transposition Cipher**

This cipher is also known as Rail Fence Cipher and is a permutation of the plain text.

### **4.11.4 Permutation Cipher**

The positions held by plain text are shifted to a regular system in this cipher so that the cipher text constitutes a permutation of the plain text.

### **4.11.5 Private-key Cryptography**

In this cipher, even the attacker is aware of the plain text and corresponding cipher text. The sender and receiver must have a pre-shared key. The shared key is kept secret from all other parties and is used for encryption as well as decryption. DES and AES algorithms are examples of this type of cipher. This cryptography is also known as "symmetric key algorithm".

#### **4.11.6 Public-key Cryptography**

In this cipher, two different keys - public key and private key - are used for encryption and decryption. The sender uses the public key to perform encryption, whereas the receiver is kept in the dark about the private key. This is also known as asymmetric key algorithm.

#### **4.12 steg image**

Using stenography process any text is hidden in a image. This image is called steg image

#### **4.13 Cryptography**

Cryptography is process in which information or data is encoded so that hackers in a communication channel are unable to disclose it.

#### **4.14 AES**

It is a cryptography technique basically which is Advanced Encryption Standard.

#### **4.15 Image grid**

It is mainly a arrangement of images align in horizontal and vertical line. It can be different in sizes and shape and its user responsive.

#### **4.16 Authentication**

The process or system of proving something genuine or exact.

#### **4.17 Pass image/ pass object**

In image based authentication system user has to choose to sequence of images which all to gather make the pass word . Each distinct images are called pass image/pass object.



#### **4.18 shoulder surfing**

it is the practice or habit of observing or spying on someone to know the pattern of their password or to know the personal information without the victim persons knowledge.

# CHAPTER 5

## SECURITY DISCUSSION

### 5.1 Shoulder Surfing

Shoulder surfing is using direct observation techniques, such as looking over someone's shoulder, to get information. Shoulder surfing is an effective way to get information in crowded places because it's relatively easy to stand next to someone and watch. [9]

### 5.2 Password Guessing

The most common type of attack is password guessing. Attackers can guess passwords locally or remotely using either a manual or automated approach. Password guessing isn't always as difficult as you'd expect. Most networks aren't configured to require long and complex passwords, and an attacker needs to find only one weak password to gain access to a network. Not all authentication protocols are equally effective against guessing attacks. Many tools can automate the process of typing password. [10]

### 5.3 Dictionary Attack

Dictionary attacks work on the assumption that most passwords consist of whole words, dates, or numbers taken from a dictionary. Dictionary attack tools require a dictionary input list. [10]

### 5.4 Password Cracking

Password cracking is the process of taking a captured password hashing and converting it to its plaintext original. To crack a password, an attacker needs tools such as extractors for hash guessing, rainbow tables for looking up plaintext passwords, and password sniffers to extract authentication information. [10]

## **5.5 Hash guessing**

Some password cracking tools can both extract and crack password hashes, but most password crackers need to have the LM password hash before they can begin the cracking process. The most popular Windows password hash extractor is the PW dump family of programs. Many password cracking tools accept PW dump-formatted hashes for cracking. Such tools usually begin the cracking process by generating some guesses for the password, then hashing the guesses and comparing those hashes with the extracted hash. [10]

## **5.6 Rainbow Table**

These days, password crackers are computing all possible passwords and their hashes in a given system and putting the results into a lookup table called a rainbow table. When an attacker extracts a hash from a target system, he or she can simply go to the rainbow table and look up the plaintext password. Some crackers (and Web sites) can use rainbow tables to crack any LM hashes in a couple of seconds. One can purchase very large rainbow tables, which vary in size from hundreds of megabytes to hundreds of gigabytes, or generate own using Rainbow Crack. Rainbow tables can be defeated by disabling LM hashes and using long, complex passwords. [10]

## **5.7 Password Sniffing**

Some password crackers can sniff authentication traffic between a client and server and extract password hashes or enough authentication information to begin the cracking process. [10]

## **5.8 Brute Force Attack**

A brute force attack is a trial-and-error method used to obtain information such as a user password. In a brute force attack, automated software is used to generate a large number of consecutive guesses as to the value of the desired data. Brute force attacks may be used by criminals to crack encrypted data. Automated password guessing programs and crackers use several different approaches. The most time consuming and most successful attack method is the brute-force attack, in which the attacker tries every possible combination of characters for a password. [10]

## **5.9 Hybrid password guess**

Hybrid password guessing attacks assume that network administrators push users to make their passwords at least slightly different from a word that appears in a dictionary. Hybrid guessing rules vary from tool to tool, but most mix uppercase and lowercase characters, add numbers at the end of the password, spell the password backward or slightly misspell it, and include characters. [10]

## **5.10 Password Resetting**

Attackers often find it much easier to reset passwords than to guess them. Many password cracking programs are actually password resitters. In most cases, the attacker boots from a floppy disk or CD-ROM to get around the typical Windows protections. Most password resitters contain a bootable version of Linux that can mount NTFS volumes and can help to locate and reset the Administrator's password. [10]

# CHAPTER 6

## STEGANOGRAPHIC IMAGE GENERATION

### 6.1 Encryption

The username will be encrypted AES encryption algorithm. Then convert the result in to binary form. This binary value will be put in image pixel by maintaining a predefined sequence called byte jumping series which is provided by our system. System change the least significant bit of color value of the pixels according to the predefined sequence. If the sequence is 1457 then system will change the least significant bit of color value of 1st of image then 5th pixel then 10th pixel and then 17th pixel according to the binary value of encrypted password. If the LSB value of a pixel is 0 and in encrypted password it is 1 then change the LSB to 1 else left unchanged. System provide this image to download for login purpose.

#### 6.1.1 AES

AES is based on a design principle known as a substitution-permutation network, a combination of both substitution and permutation, and is fast in both software and hardware.[11] Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits. The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, the third mixes columns. The last transformation is a simple exclusive or (XOR) operation performed on each column using a different part of the encryption key – longer keys need more rounds to complete.

#### 6.1.2 Byte jumping series

By this series system gets the positions of byte array of image where next byte of cipher text to be inserted. Using its value we get the decimal value of the number of the MSB of that byte. According to the decimal value with the current positions column we get the next byte

where insertion will take place.

### **6.1.3 Zero case series**

System uses zero case series in a special case. There may be a special case the decimal value of the MSB is zero. In this case we will get the next byte using zero case series.

## **6.2 Decryption**

To decrypt the data from the image we use the key. Then we have collected the LSB value using the counter part of encryption technique. Then using AES decryption we will get the our data. Some password cracking tools can both extract and crack password hashes, but most password crackers need to have the LM password hash before they can begin the cracking process. The most popular Windows password hash extractor is the PW dump family of programs. Many password cracking tools accept PW dump-formatted hashes for cracking. Such tools usually begin the cracking process by generating some guesses for the password, then hashing the guesses and comparing those hashes with the extracted hash.

# **CHAPTER 7**

## **METHODOLOGY**

### **7.1 Proposed system**

In our system, user just need to provide user name to register. He/she even dont need to generate the password as our system will generate a unique key number to the user which will be used further for lo gin procedure. From a random image grid the user will enter the key number for log in purpose

### **7.2 Registration phase**

The 1st phase in our authentication system is registration. In this phase like another authentication system user need to provide his user name which is alphanumeric. Here we r providing an extra benefit which is a user does not require to provide or create any password by thyself.

#### **7.2.1 Key Number**

After providing user name our system will generate a unique key number which consist of 4 digit decimal number. User just need to memorize this key number for login. In another word we can say this key number can be regarded as his password.

#### **7.2.2 AES encryption**

The generated key and user name will be encrypted using the cryptography method AES-128. After that this encrypted key and user name will be saved in a file so that if any hacker hacked the database, he will not get the actual user name and key number. As he dont know the AES key, he cannot easily decrypt the user name and key number.

## **7.3 Log in phase**

### **7.3.1 Enter user name**

If a user wish to log in, then the system will initially ask for his user name. a window will appear where user need to enter the correct user name. If the user name is similar to his previously provided user name, then the system will let him to proceed to the next step. In next step a 10\*10 image grid will appear.

### **7.3.2 10\*10 image grid**

In next step a 10\*10 image grid will appear. The image grid contains 25 original images and 3 copies of each image in total 100 images. 1st row of the grid is designed as 10 different images will be chosen randomly from the original images. Images of this grid are clickable except the images of 1st row. This 10 different images in the 1st row represent the value according to its index number which is numbered as 0 to 9. The duplicates of this 10 images represent the same value of original images in the 1st row. The rest of the 15 original images at the same time and its copies represent garbage value. The images and 1st row of the image grid will randomize while each log-in session. When the user will enter the 4 digit key number, by each click the whole image grid including the 1st row will be shuffled randomly. Thus while log in the image grid will be shuffled 4 times quickly so that any person or observer who is trying to assume the key number feel great difficulties. As the grid will shuffle in each click the images will change randomly and the observer also possesses an idiotic memory/photographic memory, though it will be very tough for him to recall every image correctly and he/she will be puzzled. Hence they will be fail to guess the key number.

### **7.3.3 Shoulder surfing resistance password**

If a user wish to enter his own key number except the generated one by the system, then he/she can do so. To this this user need to follow this few instructions which is set by us. User has to add the decimal digits of the date of the current day in which he/she is log in to the system in such a way that it will form a single digit ( 0-9).like if the current date is 27 then the sum of this two digit is (2+5=7). By repeating this single digit 4 times (i,e 7777) it will form a 4 digit number .Then user has to add this repeated 4 digit number with the original 4 digit key. If any time summation is garter than 10 only least significant digit will be taken. As example if the original key is 1241 then after adding 7777 with it the

1 2 4 1



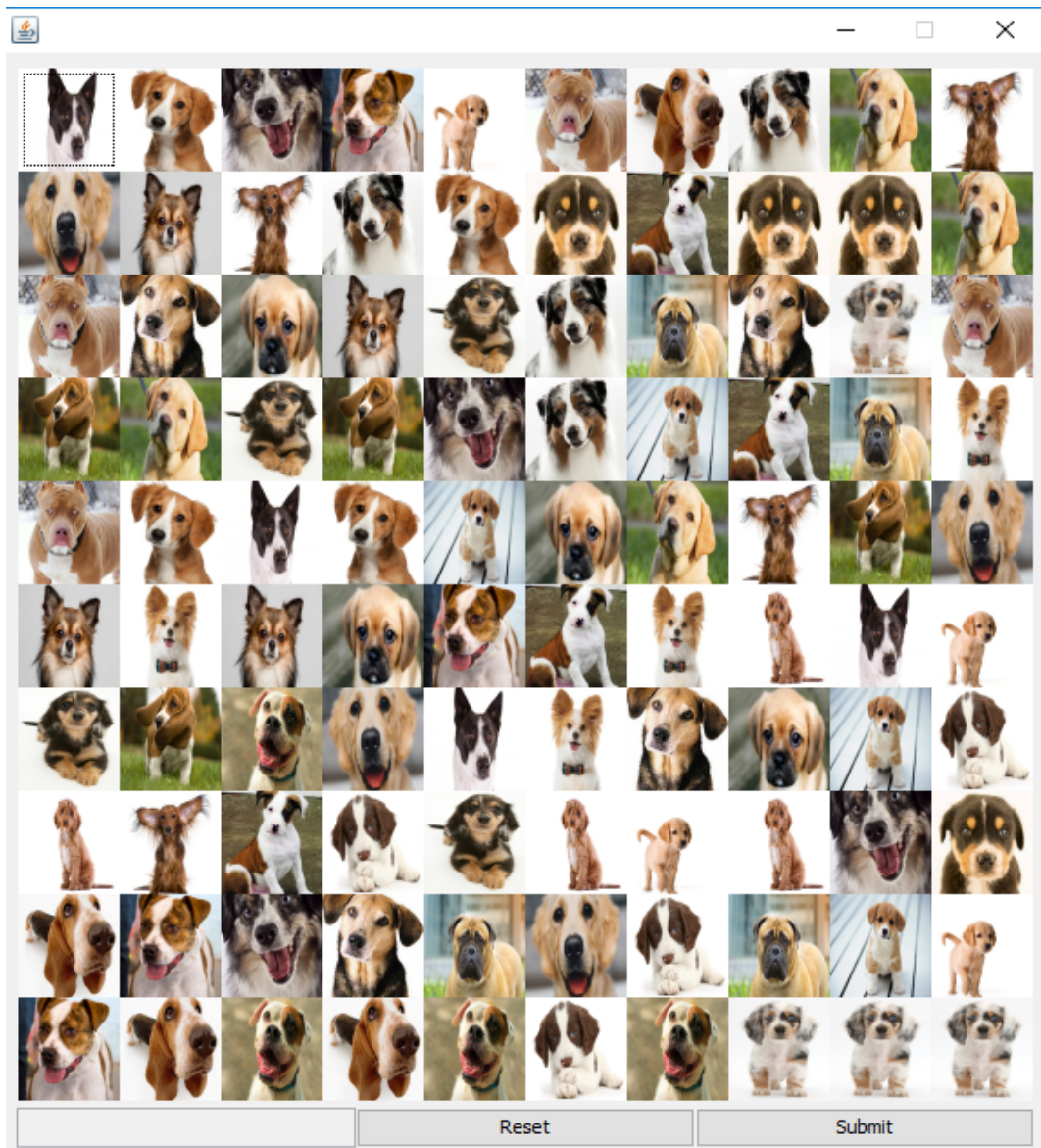


Figure 7.1: 10x10 image grid

+7 7 7 7  
8 9 1 8

8 9 1 8 = New key

Then this will be the new key number which can also be used for log in purpose to avoid shoulder surfing. System decode the original key by subtracting 4 digit repeated number which is formed from current date as same procedure during encoding. If any digit of the new key (minuend) is less than the 4 digit repeated number (subtrahend) the system will add 10 with the minuend and then will subtract it.

8 9 1 8  
-7 7 7 7

1 2 4 1 = Gets the key

Here the 3rd digit 1 is less than 7 so system adds 10 with 1 ( 1+10= 11) and 11-7= 4 is taken. By this system generates the original key and this key is cross checked with the key saved in file( In future it be replaced by database).

## 7.4 Problems Faced And Encountered

1. In our system we had to run the java code from MATLAB. For this system we want to save the output of the java code into a variable of MATLAB. For doing this, we had faced a lot of problems. 2. For writing in the file we had faced problems. 3. For reading from the file we had faced problems. 4. In our system we used AES encryption. After occurring AES encryption we used to save the output in a file. The outputs are characters. But when we read from the file for matching username or password, one or two characters were changed every time. 5. For continuous processing of java and MATLAB we can not fetch the right value from the file.

### Solution

1. For saving the output from the java code, we had used a file where the output was written. And we had fetched the value from the file in MATLAB and saved in a variable. 2. Instead of using w for writing in file, we had used wt. So, the problem is solved. 3. Instead of using r for reading from file, we had used rt. So, the problem is solved. 4. For solving AES encryption problem, we saved the double value of the output or characters in the file. So,

when we had fetched from the file. It had given the right value. 5. For continuous processing of java and MATLAB we had created a loop. This loop fetches the output of java code from the file when the file does not contain 0000. So, by this we had got the right value.

# CHAPTER 8

## SYSTEM DESIGN

### 8.1 System Architecture

Our system has two phases which interact between user and system. One is registration phase another is log in phase. Both phase has a individual architecture. One is registration architecture another is log-in architecture.

#### 8.1.1 Registration system architecture

The block diagram of registration system architecture is given below- In registration archi-

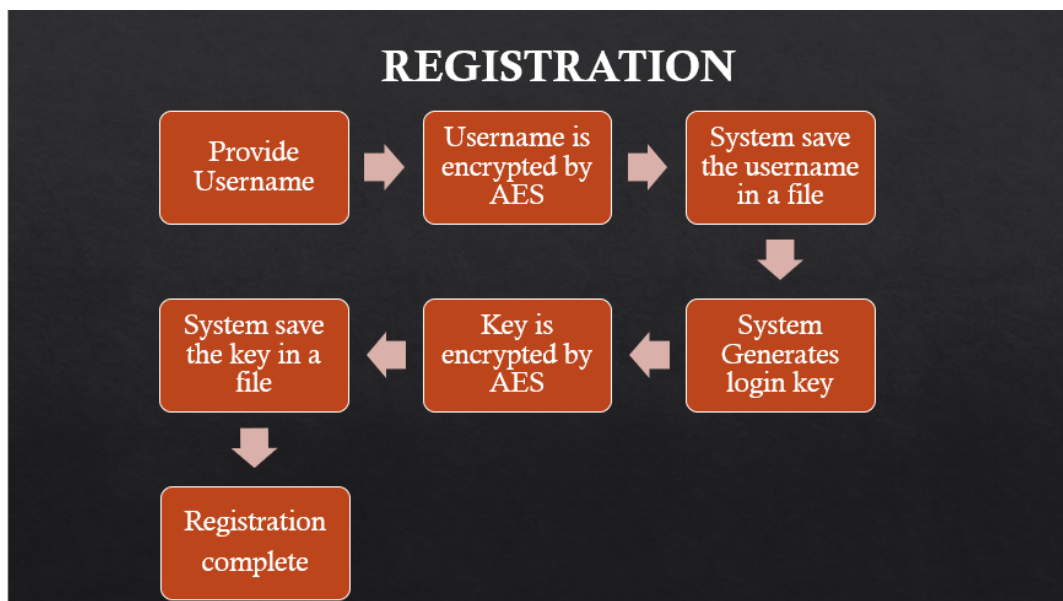


Figure 8.1: Block Diagram Of Registration System Architecture

architecture system provides a window which contains a text box for username purpose. System save this username in a text document or database. For database we used MySQL server . In database there is a table with three column id, username and password. Then system provides another textbox with a button below . By pressing the button by the user system generates a 4 digit key which is displayed in the text box. User has to remember this key for login purpose. After generating this key system displays a notification box registration completed.

### 8.1.2 Login system architecture

The block diagram of login system architecture is given below-

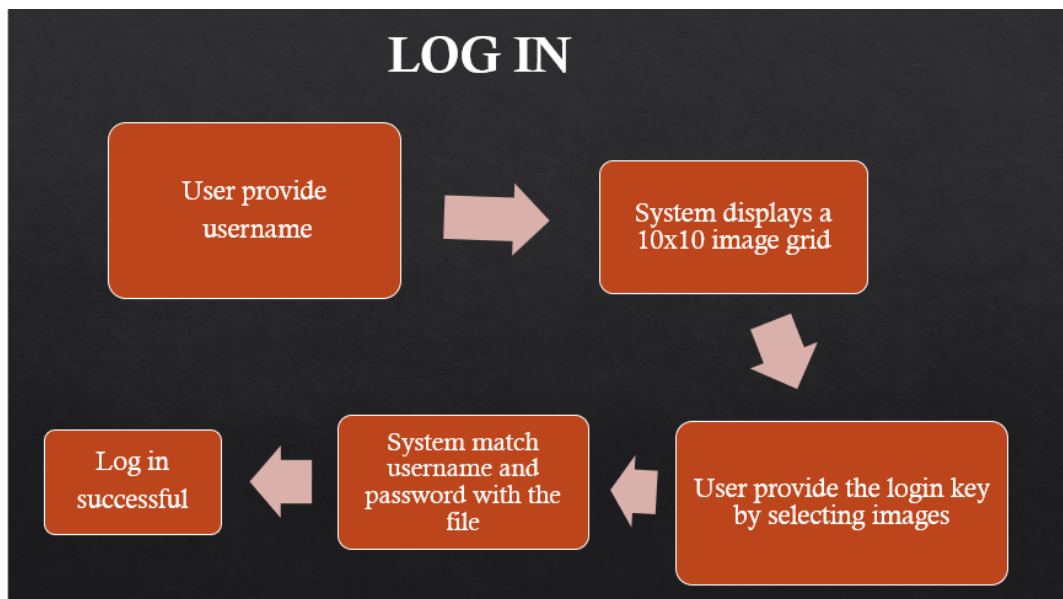


Figure 8.2: Block Diagram Of Log In System Architecture

In login architecture system provides a window with a text box for username. User has to put his username for next step. Then system displays a 10x10 image grid which contains a password box below and a reset button. By pressing the images the password box is filled up. User can reset his/her password by the reset button. If username and password is matched a notification box is displayed Login Successful. User has to press ok in the notification box to log in to the system.

# **CHAPTER 9**

## **IMPLEMENTATION**

### **9.1 Java**

Java is a object oriented programming language which is simpler to use . to create a complete application java can be used which can be run on a single computer or be distributed among servers and clients in a network. Java programming needs JDK ( Java Development Kit) which includes JRE ( Java Runtime Environment). Java use java as a interpreter and javac as a compiler. Java document generator Javadoc and archiver jar also can be used. For our system we used jdk 7.0.1 and Net Beans software as code editor.

### **9.2 Java swing**

To create window based application java swing is used . Java swing is a part of java foundation classes which is written in java. Java foundation classes are a set of GUI components which simplify the development of desktop application. Java swing provides platform independent and lightweight components. The javax.swing API such as JButton, JTextField, JTextarea, JRadioButton etc.

### **9.3 MATLAB**

We do our total project in MATLAB. MATLAB is very much usefull for proof of concept.

# CHAPTER 10

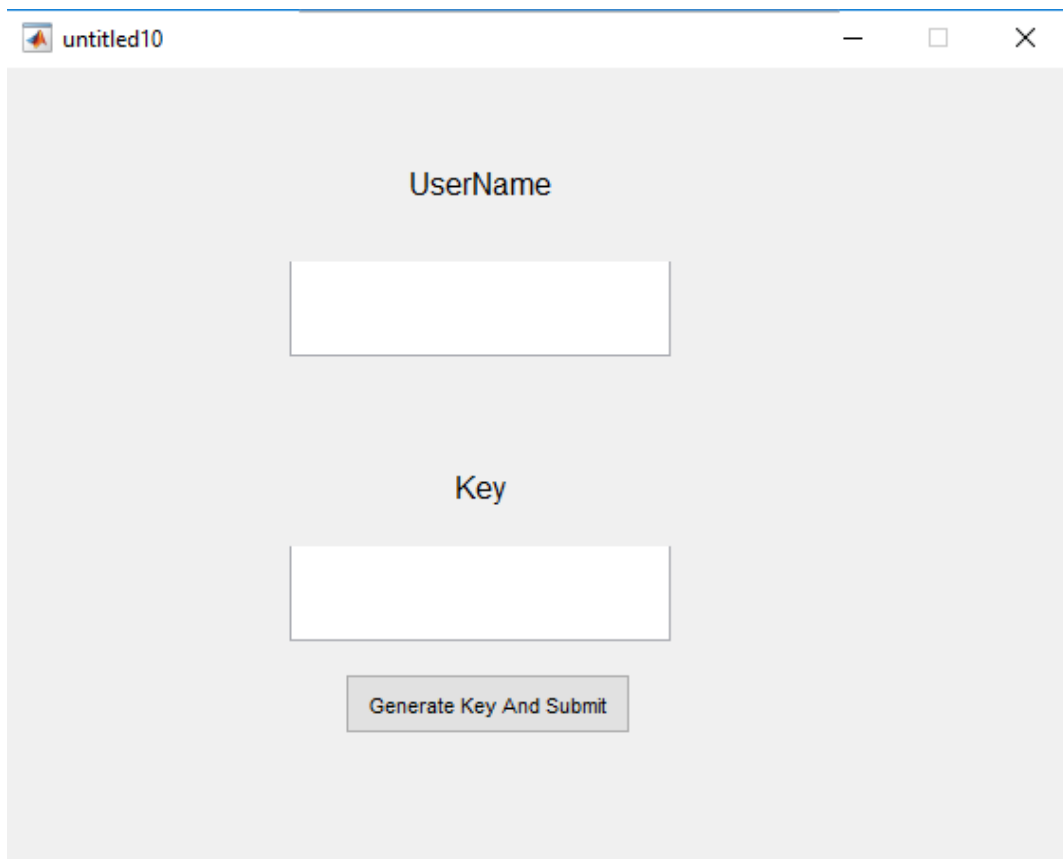
## RESULT AND ANALYSIS

### 10.1 Result

In this section we will discuss the result of our implemented system i.e registration and log in procedure

#### 10.1.1 Screen shot of registration phase

Here a user need to enter his user name. Then by clicking the generate key and submit button 4 digit key will generate. After generating the key, the registration procedure will be



The screenshot shows a window titled "untitled10" with a light gray background. It contains two text input fields. The first field is labeled "UserName" and the second is labeled "Key". Below the "Key" field is a button labeled "Generate Key And Submit".

Figure 10.1: Registration module

completed and user need to memorize the key number.

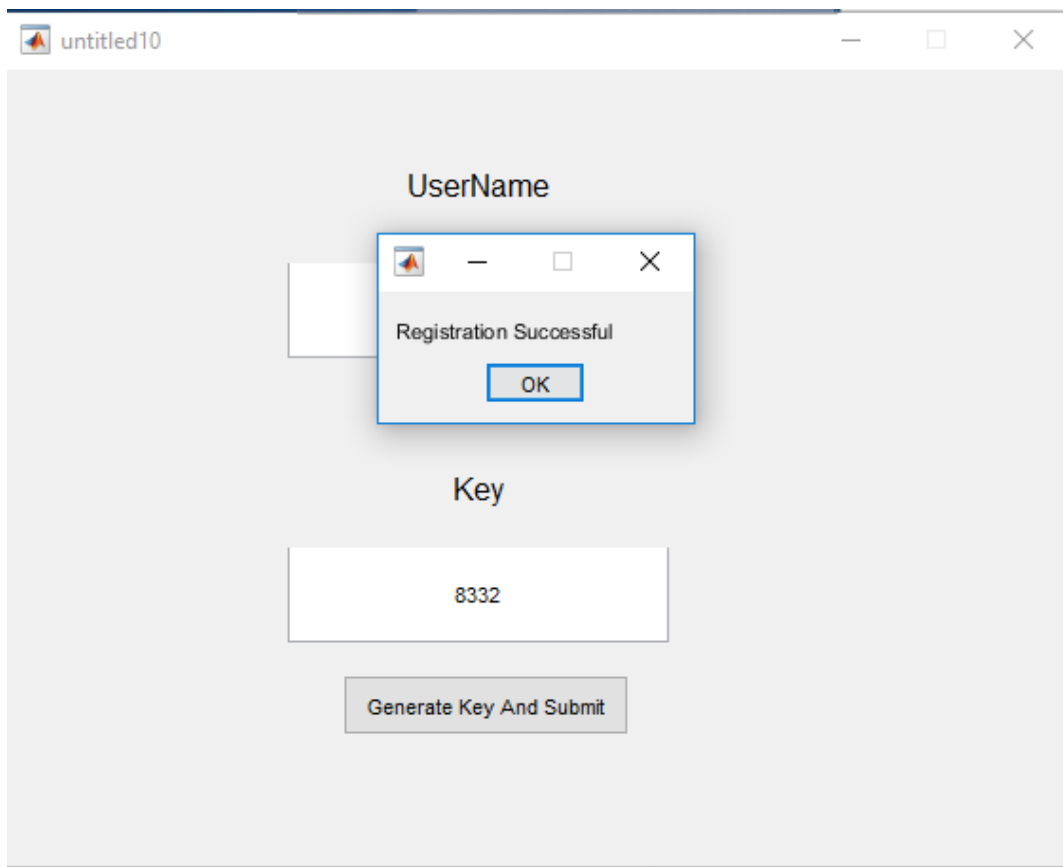


Figure 10.2: Registration procedure



User name and key encrypted by AES and saved in a file

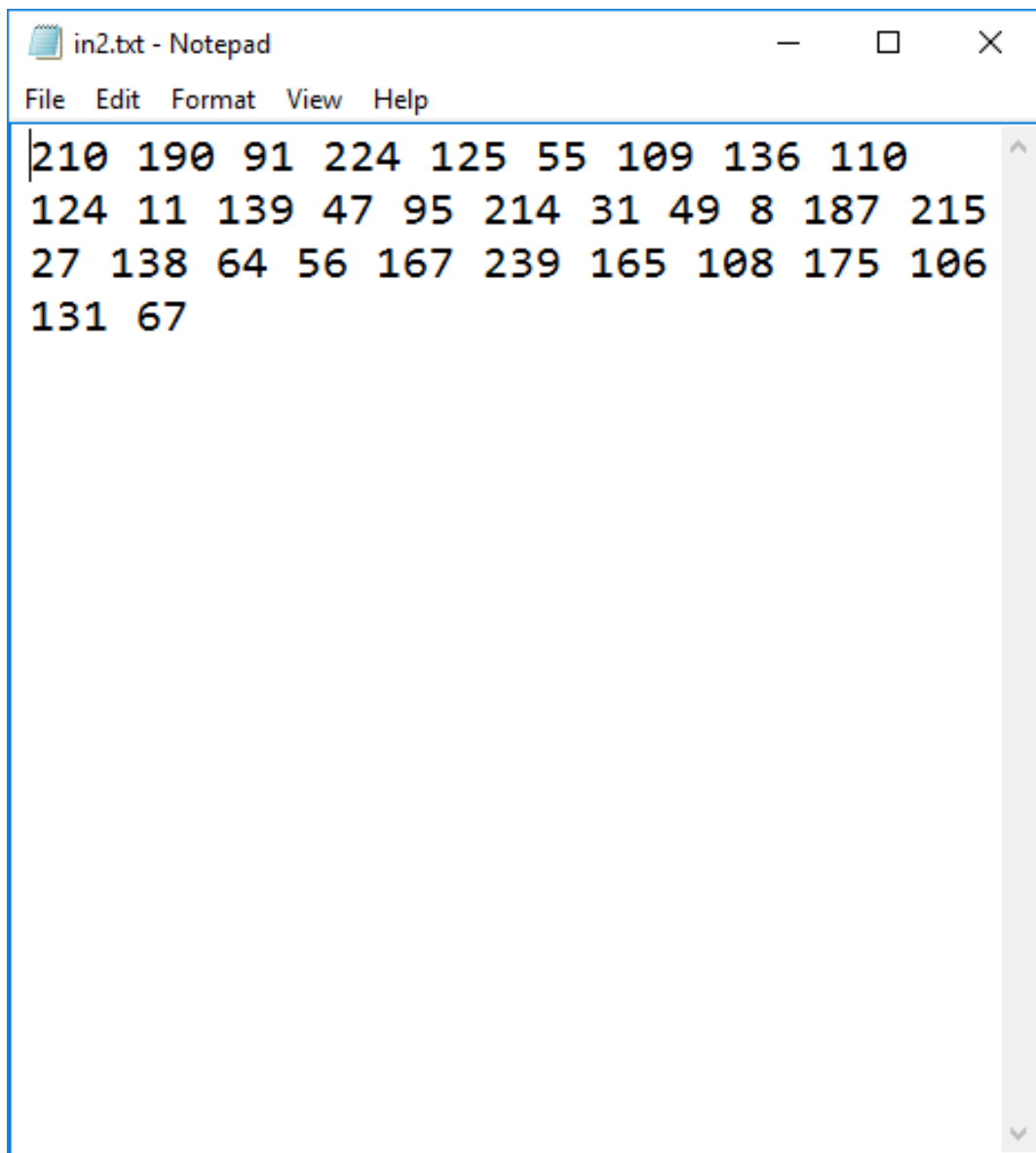


Figure 10.3: AES encrypted file

## 10.2 Login phase

When a user need to login, at first he/she have to provide the user name.

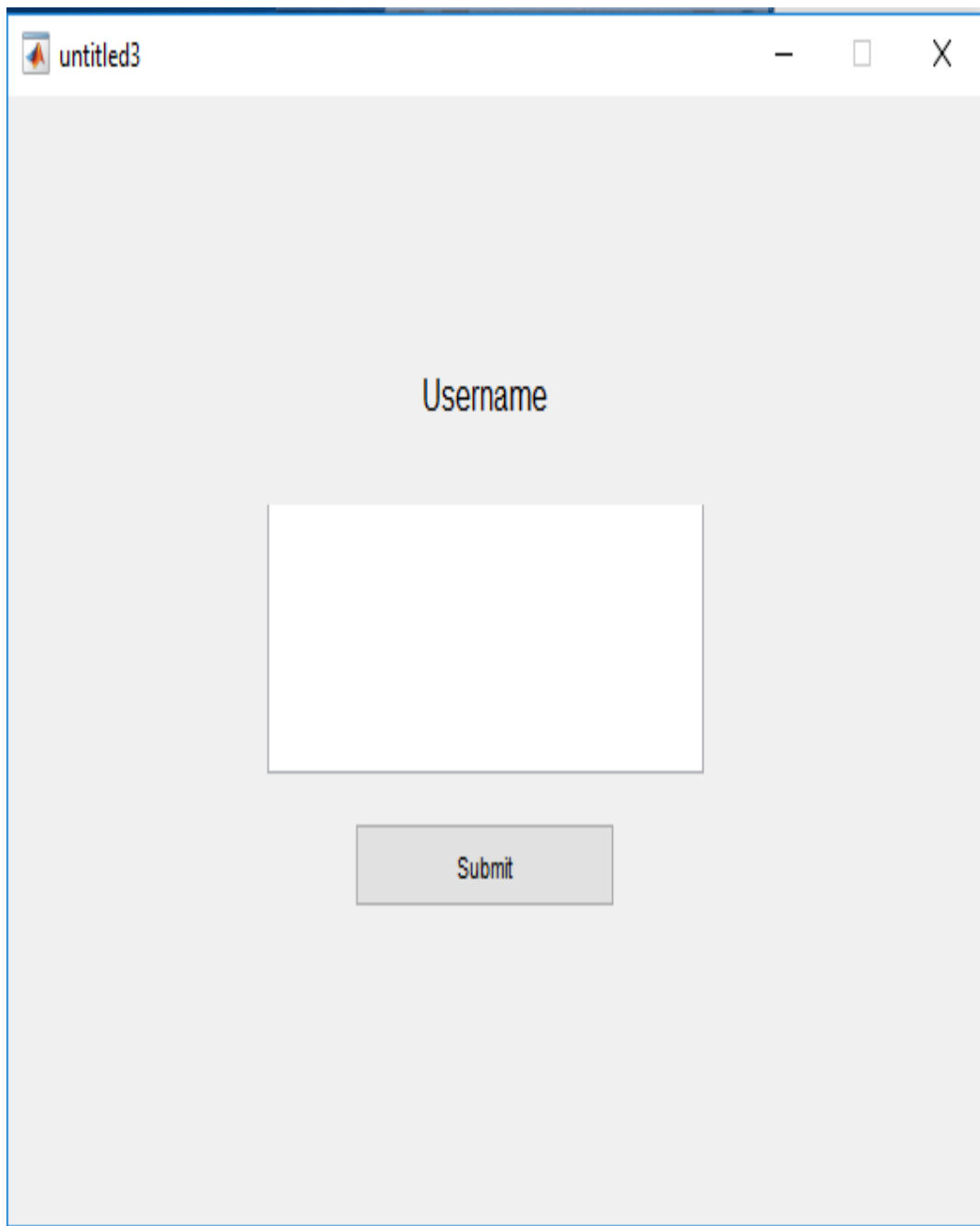


Figure 10.4: Login Module ( system ask for username )

Then after completing this step, a 10\*10 image grid will appear in the screen. User will click the images to input the key number. by each clicking the images will shuffle randomly and the 1st row is non clickable .

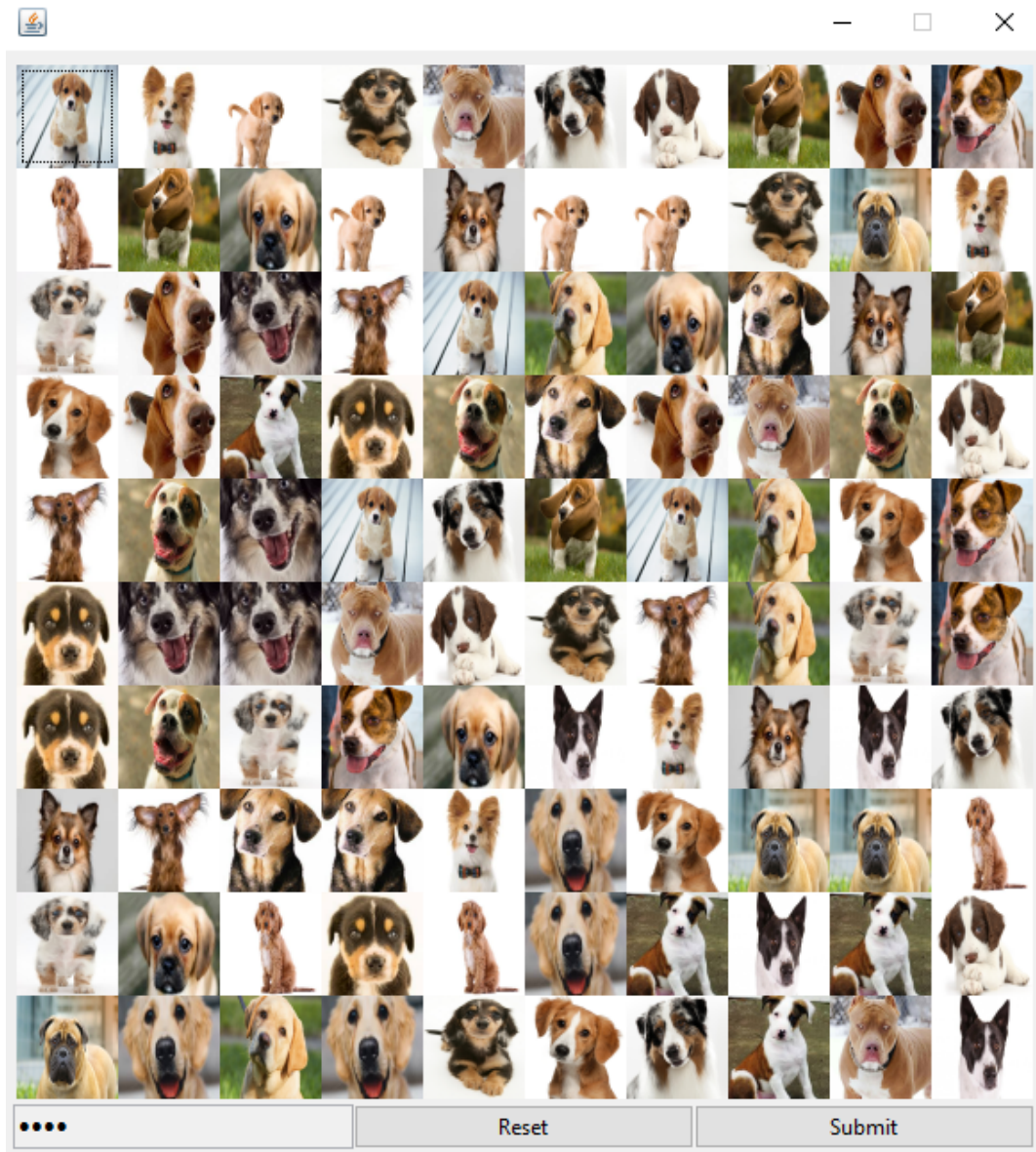


Figure 10.5: Clickable Image grid( system ask to enter key number by clicking this grid)

When the key will be entered correctly then the login will be successful and it will shown to user. If the key is not correct then login will unsuccessful.

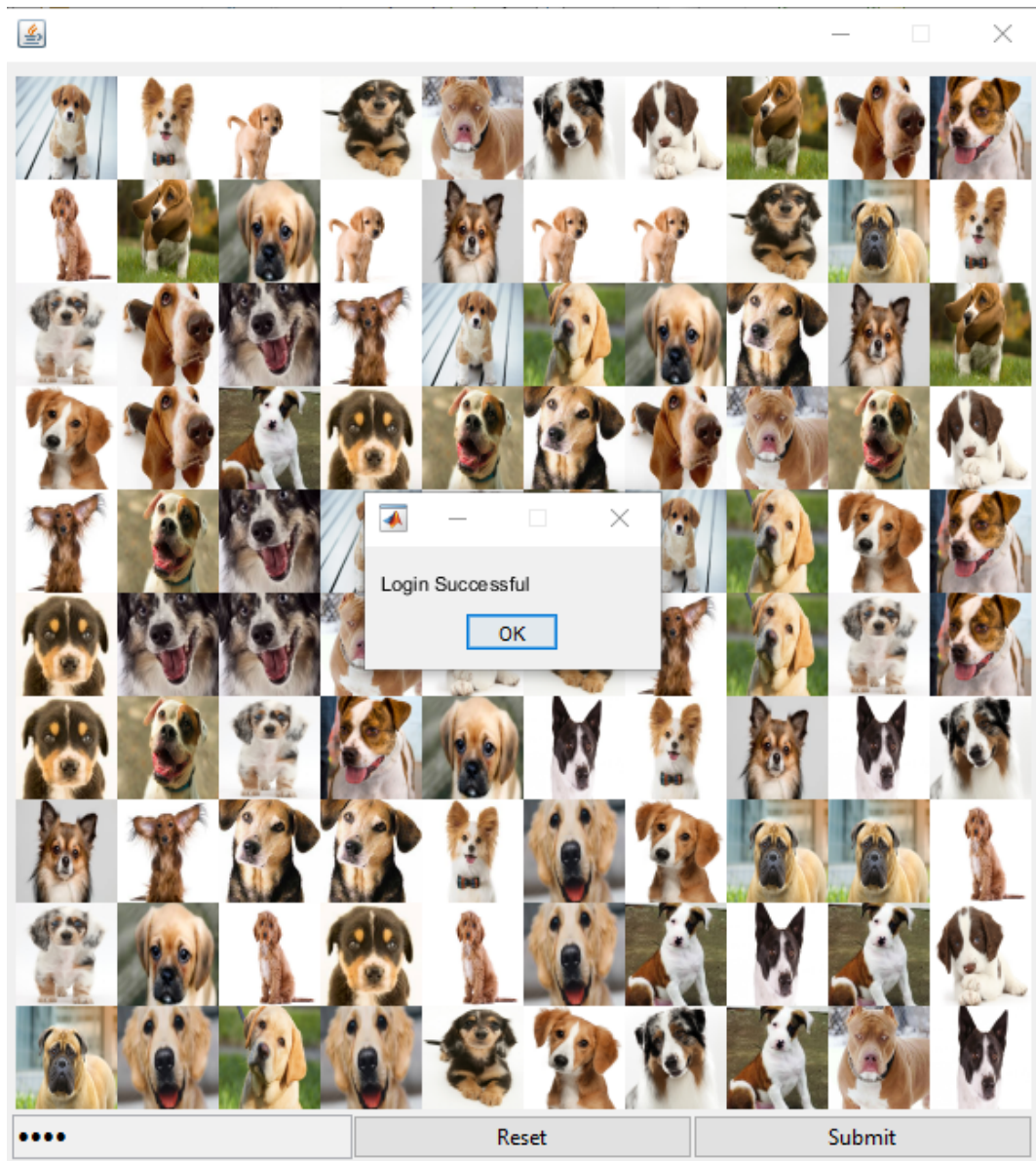


Figure 10.6: Login Module ( system shows that login is successful)

### **10.3 Analysis**

By analyzing the system performance we have reached to the conclusion that our system is quite high in performance. All the module works properly and smoothly. The image grid randomization speed is expectedly good and every time shuffle properly. In short the system performance is according to our expectation.

# **CHAPTER 11**

## **FUTURE WORK**

### **11.1 Stego image**

In future the system will ask user to enter any image he/she desire. Then the given username by the user will be encrypted using cryptography technique( here AES-128 encryption algorithm is used) before hiding it in the image using LSB. After that applying modified LSB Algorithm on the pixel byte array the final stage image will be returned to user so that user can store that in any pen drive. Then the binary value of this encrypted username is put in the image pixel by maintaining the key number which is provided by our system. If the key number is 1457 then system will change the least significant bit of color value of 1st pixel of image then 5th pixel then 10th pixel and then 17th pixel according to the binary value of encrypted.

#### **11.1.1 Replace username for login**

If a user wish to login, then the system will initially ask for that steno object from user in which the user name is hidden. Applying counter part of the modified LSB algorithm the hidden bit stream will be recovered in usable format. Then the user name can be finally decrypted using AES decryption algorithm by the system. By this user no need to enter the user name.

#### **11.1.2 Use pen drive to take username from stago image**

By inserting the pen drive, a window will appear in front the screen that from which device user will enter the steno image. Then when user will choose the USB device, the system will automatically choose the steno image from the predrive and decrypt the user name. If the user name matched, then the image grid will appear automatically.

## **11.2 AES key randomization**

Initially we are encrypting the key number and user name using a static AES key. Further we will use a randomize key which will enrich the security of the system.

## **11.3 Prevent brute force attack**

In future we will add an extra feature in our system which will prevent brute attack. By entering wrong key number 5 times the login system will be block for almost half an hour. Only administrator will be able to unlock the system. For this brute attack will be prevented.

## **11.4 Transform the whole system into prototype**

In future we will transform our system ( proof of concept) into prototype using any scripting language like java script so that any one can easily use our module in their respective application where authentication is needed.

## **CHAPTER 12**

### **ADVANTAGE AND LIMITATION**

#### **12.1 Advantage**

Our system prevent all most up to 95 chance of shoulder surfing which is very high for any system. It also overcome all most all the problem of textual password system. The randomize speed of image grid is good enough which makes our system high performing. Percentage of database hack is almost low. In case of hacking occur, the information will be safe as information are encrypted and then saved in database. User memorization skill is not a big concern here. User just need to memorize the key number and nothing else. So its a huge advantage which makes our system efficient.

#### **12.2 Limitation**

User name is textual which is a limitation here. We are only using date to randomize key. No recovery system is included in our system yet which is a considerable limitation. Use of Static key for AES makes our system a little insecure.



# **CHAPTER 13**

## **CONCLUSION**

### **13.1 Contribution**

Our system will contribute a lot in the platform of graphical authentication system. It will be a efficient tool for protecting highly confidential data and information which will add a new dimension to the security purpose. Almost No textual password breaking and graphical password failure will occur in our system, so it will serve security platform at a great deal.

### **13.2 Future work**

In this paper we have proposed graphical authentication system which is free from textual password. Our model is workable for application level, further we are desired to enhance this model to OS level. Now this model is proof of our proposed system, further switching this to any other scripting language like java script we can use this model as library function. So that it can be integrated with any system for authentication to enhance and provide tight security to the system. Basically we r highly craving to transform our proposed system to framework so that peoples or organizations who actually need secured authentication system they can use our system easily.

## REFERENCES

- [1] S. Man, D. Hong, and M. Mathews, "A shoulder-surfing resistant graphical password scheme," in Proceedings of International conference on security and management. Las Vegas, NV, 2003
- [2] Yamamoto, Takumi Kojima, Yuko Nishigaki, Masakatsu. (2009). A Shoulder-Surfing-Resistant Image-Based Authentication System with Temporal Indirect Image Selection.. 188-194.
- [3] P Waghmare, R Longadge, D Kapgate. "A Review on Two Level Authentication Using Image Selection and Voice Recognition",Feb 2017.
- [4] Pintu R Shah. "Image based Authentication System".
- [5] Ms. Sreya Prakash,Mrs. Sreelakshmy M K. "A Secure Graphical Password Authentication System.
- [6] <https://en.wikipedia.org/wiki/Password>
- [7] <https://en.wikipedia.org/wiki/Passphrase>
- [8] <http://searchsecurity.techtarget.com/definition/graphical-password>
- [9] <http://searchsecurity.techtarget.com/definition/shoulder-surfing>
- [10] <http://www.itprotoday.com/security/types-password-attacks>

## APPENDIX A

### Registration.m File

```
function varargout = untitled10(varargin)

gui_singleton = 1;
gui_state = struct('gui_Name', mfilename, ...
'gui_singleton', gui_singleton, ...
'gui_OpeningFcn', @untitled10_OpeningFcn, ...
'gui_OutputFcn', @untitled10_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin ischar(varargin{1})
    gui_state.gui_Callback = str2func(varargin{1});
end

if nargout
    varargout{1:nargout}
    = gui_mainfcn(gui_state, varargin:);
else
    gui_mainfcn(gui_state, varargin:);
end

function untitled10_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = untitled10_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;
```

```
function edit1_Callback(hObject,eventdata,handles)
```

```
function edit1_CreateFcn(hObject,eventdata,handles)
```

```
if ispc isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
set(hObject,'BackgroundColor','white');  
end
```

```
function edit2_Callback(hObject,eventdata,handles)
```

```
function edit2_CreateFcn(hObject,eventdata,handles)
```

```
if ispc isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
set(hObject,'BackgroundColor','white');  
end
```

```
function pushbutton1_Callback(hObject,eventdata,handles)
```

```
a=get(handles.edit1,'string');  
if (a =='')  
fid=fopen('in.txt','wt');  
fprintf(fid,'end  
aa=randi([1000, 9999])  
set(handles.edit2, 'String', num2str(aa));  
fid=fopen('in1.txt','wt');  
fprintf(fid,'msgbox('Registration Successful');  
key=num2str(aa);  
lenkey = length(key);  
keyextra = mod(lenkey,16);  
keyadd = 16 - keyextra;  
add = char.empty(0,15);  
for i = 1 : 1 : keyadd  
add(i) =' *';  
end
```

```
key = [keyadd]
```

```
sms = a;  
lenms = length(sms);  
smsextra = mod(lenms, 16);  
smsadd = 16 - smsextra;  
add = char.empty(0, 15);  
for i = 1 : 1 : smsadd  
    add(i) = ' *';  
end  
sms = [smsadd]
```

```
ke='4444*****';  
sms = double(sms);  
ke = double(ke);  
s = aesinit(ke);  
chipertext = aes(s,'encrypt','ecb', sms)  
disp(char(chipertext));
```

```
fid2=fopen('in2.txt','wt');  
fprintf(fid2,"
```

```
ke='8888*****';  
sms = double(key);  
ke = double(ke);  
s = aesinit(ke);  
chipertext = aes(s,'encrypt','ecb', sms)  
disp(char(chipertext));
```

```
fprintf(fid2,"
```

## APPENDIX B

### Login.m File

```
function varargout = untitled3(varargin)
```

```
    gui_singleton = 1;
```

```
    gui_state = struct('gui_Name', mfilename, ...'gui_singleton', gui_singleton, ...'gui_OpeningFcn', @untitled3_OpeningFcn, ...  
    'ifnarginischar'(varargin1)//gui_state.gui_Callback = str2func(varargin1);  
    end
```

```
    if nargin
```

```
        varargout1:nargout
```

```
        = gui_mainfcn(gui_state, varargin:);
```

```
    else
```

```
        gui_mainfcn(gui_state, varargin:);
```

```
    end
```

```
    handles.output = hObject;
```

```
    guidata(hObject, handles);
```

```
function varargout = untitled3_OutputFcn(hObject, eventdata, handles)
```

```
varargout1 = handles.output;
```

```
function edit3_Callback(hObject, eventdata, handles)
```

```
function edit3_CreateFcn(hObject, eventdata, handles)
```

```
    if ispc & isequal(get(hObject,'BackgroundColor'),
```

```
        get(0,'defaultUicontrolBackgroundColor'))
```

```
        set(hObject,'BackgroundColor','white');
```

```
    end
```

```
function pushbutton1_Callback(hObject,eventdata,handles)
```

```

a=get(handles.edit3,'string');
sms = a;
lenms = length(sms);
smsextra = mod(lenms,16);
smsadd = 16 - smsextra;
add = char.empty(0,15);
for i = 1 : 1 : smsadd
add(i) = ' *';
end
sms = [smsadd]

```

```

key='4444';
lenkey = length(key);
keyextra = mod(lenkey,16);
keyadd = 16 - keyextra;
add = char.empty(0,15);
for i = 1 : 1 : keyadd
add(i) = ' *';
end
key = [keyadd]

```

```

fid=fopen('in2.txt','r');
chipertext=fscanf(fid,'

key =double(key)
s = aesinit(key)
smsstar = aes(s,'decrypt','ecb',chipertext)
smsstar = char(smsstar)

```

```
l= strcmp(sms,smsstar);
```

```
if (l==1)
```

```
setappdata(0,'lvalue',l);
```

Untitled7

else

setappdata(0,'lvalue',1);

Untitled7

end



## APPENDIX C

### imageCall.m File

```
q='0000'  
fid=fopen('example.txt','wt');  
fprintf(fid,'o=Window  
javaMethod('main',o,'');
```

```
fid1=fopen('example.txt','r');  
qqq=fgets(fid1)  
while qqq=='0000'  
fclose(fid1)  
fid1=fopen('example.txt','r');  
qqq=fgets(fid1)  
end  
fclose(fid1)
```

## APPENDIX D

### cal.m File

```
ll=datetime('now')
fid=fopen('date.txt','wt');
fprintf(fid,'fclose(fid)
fid=fopen('date.txt','r');
aaaa=fgets(fid)
aaaaa=regexp(aaaa,'-', 'split')
d=str2num(aaaaa1)
```

```
key='8888';
lenkey = length(key);
keyextra = mod(lenkey,16);
keyadd = 16 - keyextra;
add = char.empty(0,15);
for i = 1 : 1 : keyadd
add(i) = '*';
end
key = [keyadd]
```

```
fid=fopen('in2.txt','r');
chipertext1=fscanf(fid,'chipertext=fscanf(fid,'
key =double(key)
s = aesinit(key)
smsstar = aes(s,'decrypt','ecb',chipertext)
smsstar = char(smsstar)
```

```
a = char.empty(0,15);
for i=1 : 1:4
a(i)= smsstar(i);
end
```

```
a=str2num(a)
```

```
e=rem(d,10)
f=floor(d/10)
g=e+f
if(gi=10)
```

```
e=rem(g,10);
f=floor(g/10);
g=e+f;
end
```

```
aa=rem(a,10);
aa=aa+g;
if(aai=10)
```

```
aa=rem(aa,10);
```

```
end
bb=floor(a/10);
```

```
cc=rem(bb,10);
cc=cc+g;
if(cci=10)
```

```
cc=rem(cc,10);
```

```
end
dd=floor(bb/10);
```

```
ee=rem(dd,10);
ee=ee+g;
if(eei=10)
```

```
ee=rem(ee,10);
```

```

end
ff=floor(dd/10);

ff=ff+g;
if(ff_i=10)

ff=rem(ff,10);

end

ff=ff*1000;
ff=ff+(ee*100);
ff=ff+(cc*10);
ff=ff+(aa*1);
fprintf('

fid1=fopen('example.txt','r');
qqq=fgets(fid1)
a=num2str(a)
ff=num2str(ff)
LLL=strlength(ff)
if(LLL==3)
ff=strcat('0',ff)
end

llll = getappdata(0,'lvalue')

if(qqq==a)
if(llll==1)
msgbox('Login Successful');
else
msgbox('Key or UserName Does not Match');
end
elseif(qqq==ff)
if(llll==1)
msgbox('Login Successful');
else

```

```
msgbox('Key or UserName Does not Match');  
end  
else  
msgbox('Key or UserName Does not Match');  
end
```