CHAPTER 1

INTRODUCTION

Password authentication is a common approach to the system security and it is also a very important procedure to gain access to user resources. In the conventional password authentication methods a server has to authenticate the legitimate user.

1.1 Introduction

For the vast majority of computer systems, passwords are the method of choice for authenticating users. The most widely and commonly used authentication is traditional "Username" and "password". For such authentication generally text (alphanumeric) is used. It is well-known, however, that passwords are susceptible to attack as users tend to choose passwords that are easy to remember, and often this means that they are also easy for an attacker to obtain by searching for candidate passwords. On the other hand, if a password is hard, then it is often hard to remember. Thus an innovative and more secure way of selecting passwords: Graphical Passwords.

Alphanumeric Password is the leading mechanism for verifying the identity of computer users, even though it is well known that people normally choose passwords that are vulnerable to different attacks. However, secure passwords should be random and should be hard to guess, they should be changed frequently.

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 called graphical user authentication.

Various graphical password schemes have been proposed as alternatives to text-based passwords. Psychology studies have revealed that the human brain is better at recognizing and recalling images than text. Graphical passwords are intended to capitalize on this human characteristic in hopes that by reducing the memory burden on users, coupled with a larger full password space offered by images, more secure passwords can be produced and users will not resort to unsafe practices in order to cope.

Graphical passwords offer better security than text-based passwords because many people, in an attempt to memorize text-based passwords, use plain words. A dictionary search can often hit on a password and allow a hacker to gain entry into a system in seconds. But if a series of selectable images is used on successive screen pages, and if there are many images on each page, a hacker must try every possible combination at random.

1.2 Literature Survey

The "Pass-Go, a New Graphical Password Scheme [1] ", paper reviews the different graphical password schemes like Passfaces, PassPoints, V-Go, Background DAS, their limitations and inspired by an old Chinese game, new graphical password scheme, Pass-Go, was designed in which a user selects intersections on a grid as a way to input a password and by offering an extremely large password space.

The "Pass-Go: A Proposal to Improve the Usability of Graphical Passwords [2]"presented a new graphical password scheme and shown that it keeps most of the advantages of the previous scheme and offers stronger security and better usability. This also includes an efficient and human readable encoding scheme; identification of the need and a solution for keyboard input support; several solutions for the shoulder surfing problem.

The "Background Pass-Go (BPG), a New Approach for GPS [3]" paper have proposed an enhancement method of Pass-Go which named BPG. This discusses about how BPG works and its ability to improve limitation over other schemes and also address some solutions to overcome certain threats to networked computer systems by using BPG.

The "Graphical password system using scalable multi-grid method [4]" aid the Scaling of Multi-Grid methodology in memorability and usability of graphical password systems have been shown and MGBPG is presented as a new approach in graphical password schemes. Usability and memorability advantages of new scheme over other Graphical Password Systems especially Background Pass-Go have been shown.

The "Multi-Grid Background Pass-Go [5]" paper presents MGBPG scheme, which manages to minimize the memorability issue by enabling each user to personalize their background image and the grid line scaling. Instead of having to remember the coordinate pairs of their passwords as in previous schemes, the users can now remember their passwords better by recalling back which part of the background image and the specific grid coordinate scale that they had clicked or drawn.

The "Authentication Using Graphical Passwords [6]" describes the two key human factors criteria which is memorability and efficiency and also provides the need of Graphical Passwords instead of textual passwords.

1.3 History

A **password** is a secret word or string of characters that is used for authentication, to prove identity or gain access to a resource (example: an access code is a type of password) .
The use of passwords is known to be ancient. Sentries would challenge those wishing to enter an area or approaching it to supply a password or *watchword.* Sentries would only allow a person or group to pass if they knew the password. In modern times, user names and passwords are commonly used by people during a login process that controls access to protected computer operating systems, mobile phones, cable TV decoders, automated teller machines (ATMs), etc. A typical computer user may require passwords for many purposes: logging in to computer accounts, retrieving e-mail from servers, accessing programs, databases, networks, web sites, and even reading the morning newspaper online.

Computer security depends largely on passwords in order to authenticate human users. The main drawback of passwords is what we call the password **problem**, namely the fact that passwords are expected to comply with two conflicting requirements:
(1) Passwords should be easy to remember, and the user authentication protocol should be executable.
(2) Passwords should be secure, i.e., they should look random and should be hard to guess; they should be changed frequently, and should be different on different accounts of the same user. They should not be written down or stored in plain text.

Classical studies have shown that, human users tend to choose and handle alphanumeric passwords very insecurely. **Graphical passwords** are a solution to the password problem. The idea of graphical passwords, first described by Greg Blonder [G. Blonder, Graphical Passwords, United States Patent 5559961 (1996)], is to let the user click (with a mouse or a stylus) on a few chosen regions in an image that appears on the screen. To log in, the user has to click in the same regions again. In Blonder-style graphical passwords, only pre-processed images can be used, the click regions can only be chosen from certain pre-designed regions in the image. This implies that the users cannot provide images of their own for making passwords, and users cannot choose click places that are not among the preselected ones. Users choose any places that attract them as click regions, such places are easier to remember. However, allowing arbitrary click locations lead to a stability problem, which has to be overcome. The problem is that users cannot be expected to click always on exactly the same location (when they intend to). So images are discretized , by using a square grid. But that leads to border problems: If the chosen click location is near the edge of a grid-square, the user will sometimes click in one square, sometimes in a neighboring square. A multi-grid method is devised, called robust **discretization**, and which leads to a stable output for the user's clicking actions. An approximation parameter r is used; as long as the user clicks within distance r of the originally chosen click location, the output of the clicking will be the same (e.g., $r = 2$ mm).

It is important to have stable output, because the output of the discretized clicking will undergo a secure hash ("password encryption") for security reasons, the actual graphical password in the computer are not stored, just the hash value. So, the system does not know the graphical password explicitly and hence cannot check whether a user's clicks are approximately correct. The hashing of passwords leads to the requirement that the user's clicks at login must always be in the same multi-grid squares. Hence, a robust discretization is needed.

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. Suppose an 8-character password is necessary to gain entry into a particular computer network. Instead of w8KiJ72c, for example, a user might select images of the earth (from among a screen full of real

and fictitious planets), the country of France (from a map of the world), the city of Nice (from a map of France), a white stucco house with arched doorways and red tiles on the roof, a green plastic cooler with a white lid, a package of Gouda cheese, a bottle of grape juice, and a pink paper cup with little green stars around its upper edge and three red bands around the middle.

Graphical passwords may offer better security than text-based passwords because many people, in an attempt to memorize text-based passwords, use plain words (rather than the recommended jumble of characters). A dictionary search can often hit on a password and allow a hacker to gain entry into a system in seconds. But if a series of selectable images is used on successive screen pages, and if there are many images on each page, a hacker must try every possible combination at random. If there are 100 images on each of the 8 pages in an 8-image password, there are 100^8, or 10 quadrillion (10,000,000,000,000,000), possible combinations that could form the graphical password. If the system has a built-in delay of only 0.1 second following the selection of each image until the presentation of the next page, it would take (on average) millions of years to break into the system by hitting it with random image sequences.

1.4 Motivation

The most common computer authentication method is to use alphanumerical usernames and passwords. This method has been shown to have significant drawbacks. For example, users tend to pick passwords that can be easily guessed. On the other hand, if a password is hard to guess, then it is often hard to remember. To address this problem, authentication method has been developed, that use pictures as passwords.

Use graphics (images) instead of alphanumerical passwords

- A picture is worth a thousand words
- Humans remember pictures better than words

It is more difficult to break graphical passwords using the traditional attack methods such as brute force search, dictionary attack, or spyware.

Graphical password schemes have been proposed as a possible alternative to text-based schemes, motivated partially by the fact that humans can remember pictures better than text; psychological studies supports such assumption. Pictures are generally easier to be remembered or recognized than text. In addition, if the number of possible pictures is

sufficiently large, the possible password space of a graphical password scheme may exceed that of text based schemes and thus presumably offer better resistance to dictionary attacks. Because of these (presumed) advantages, there is a growing interest in graphical password. In addition to workstation and web log-in applications, graphical passwords have also
been applied to ATM machines and mobile devices.

1.5 Problem Statement

In this project, we are implementing the 'PASS-GO' technique which is a grid based scheme in order to overcome the drawbacks of traditional password schemes. Pass-Go requires a user to select (or touch) intersections, as a way to input a password, which is stored in a database during the initial registration process. User can give their password as a set of points as well as multiple lines satisfying the constraints (i.e. intersections).Later during the login process once the user logins with appropriate username and password, the process is success otherwise 2-3 attempts is given to correct his/her passwords for login.

1.6 Objective

1. The main aspect of this project is to provide the stronger security in the system as Pass-Go implement a grid of larger size.
2. Pass-Go scheme improves the memorability issue in such a way that users are able to remember and memorize their passwords better.
3. One more aspect of this implementation is that, we provide a user friendly platform by adding attractive features.

1.7 Organization of the report

The Chapter 2 explains about the Strength of Graphical Passwords and the security factors. In Chapter 3 a brief introduction about the authentication methods is given. The Chapter 4 discusses the requirement analysis which includes different tools used to implement the project. In Chapter 5 design and implementation describes about design constraints and implementation

activities respectively. The Chapter 6 gives the project results and snapshots. The Chapter 7 includes the conclusion and future scope of the undertaken project.

CHAPTER 2

STRENGTH OF GRAPHICAL PASSWORD

Using strong passwords lowers overall risk of a security breach, but strong passwords do not replace the need for other effective security controls. The effectiveness of a password of a given strength is strongly determined by the design and implementation of the authentication system software, particularly how frequently password guesses can be tested by an attacker and how securely information on user passwords is stored.

2.1 Textual Password

Alphanumeric password is derived from a Character Set. There are so many types of Character sets depending upon the application where we need authentication. One of the well known Character Set is the American Standard Code for Information Interchange (ASCII). It is a character encoding scheme based on the ordering of the English alphabet. ASCII includes definitions for 128 characters: 33 are nonprinting control characters (now mostly obsolete) that affect how text and space is processed; 94 are printable characters, and the space is considered as an invisible graphic.

Some of the usual *problems* are the following:

1. Users choose passwords that are very short in length.
2. Users choose passwords that are easy to remember.
3. Users write passwords down or share them with others, in order to remember them easier.
4. Users use the same passwords for different applications.

2.2 Why Graphical Password

Graphical passwords were originally described by Blonder (1996). In this an image would appear on the screen, and the user would click on a few chosen regions of it. If the correct regions were clicked in, the user would be authenticated.

Memory of passwords and efficiency of their input are two key human factors criteria.

Memorability has two aspects: (1) how the user chooses and encodes the password and (2) what task the user does when later retrieving the password. In a graphical password system, a

user needs to choose memorable locations in an image. Choosing memorable locations depends on the nature of the image itself and the specific sequence of click locations. To support memorability, images should have semantically meaningful content because meaning for arbitrary things is poor. This suggests that jumbled or abstract images will be less memorable than concrete, real-world scenes. To retrieve the locations a user will be dependent on the encoding used while learning. A poor encoding will hurt retrieval by failing to distinguish similar objects.

Efficiency is important in password systems because users want to have quick access to systems. The time to input a graphical password by a highly skilled, automated user can be predicted by Fitts' Law (1954). The law states that the time to point to a target depends on the distance and size of the target – greater distance and smaller targets lead result in slower performance.

Password scheme has three components: password selection (during which a user chooses click points in the provided image);password confirmation (during which users re-enter
their click points. If they make an error, they can clear their clicks and choose others); and login (during which a user clicks close to the previously chosen click points in the provided image). This is figuratively represented in Figure 2.1:

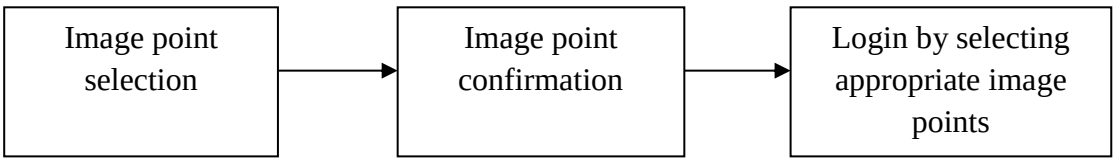

Figure 2.1: steps to create a graphical password

SECURITY

An authentication system must provide adequate security for its intended environment, otherwise it fails to meet its primary goal. A proposed system should at minimum be evaluated against common attacks to determine if it satisfies security requirements. The types of attacks on knowledge-based authentication into two general categories: guessing and capture attacks. In successful guessing attacks, attackers are able to either exhaustively search through the entire theoretical password space, or predict higher probability passwords (i.e., create a dictionary of likely passwords) so as to obtain an acceptable success rate within a manageable number of guesses. Guessing attacks may be conducted through the intended login interface, or Online if some variable text [50] (e.g., hashes) can be used to assess the correctness of guesses. Authentication systems with small theoretical password spaces or with identifiable Patterns in user choice of passwords are especially vulnerable to guessing attacks. Password capture attacks involve directly obtaining the password, or part thereof, by capturing login credentials when entered by the user, or by tricking the user into divulging their password. Shoulder-surfing, phishing, and some kinds of malware are common forms of capture attacks. In shoulder surfing, credentials are captured by direct observation of the login process or through some external recording device such as a video camera. Phishing is a type of social engineering where users are tricked into entering their credentials at a fraudulent website recording user input.

Malware uses unauthorized software on client computers or servers to capture keyboard, mouse, or screen output, which is then parsed to and login credentials. As will be seen, early graphical password systems tended to focus on one particular strength, for example being resistant to shoulder-surfing, but testing and analysis showed that they were vulnerable to one or more other types of attacks. Except in very specific environments, these would not provide adequate security. Often playing an important role related to security is the particular process of encoding or discretization used transforming the user input into discrete units that can be identified by the system and used for comparison duringpassword re-entry. As will be seen, some schemes require that the system retains knowledge of the exact secret (or portion thereof), either to display the correct set of images to the user or to verify password entries. In other cases, encoded or discretized passwords may be hashed, using a one-way cryptographic hash, to provide additional security in case the password _le is compromised.

2.3 Security Factors

There are many aspects of security issues, brutal force attack

1) Brute force search.
2) Dictionary attacks.
3) Guessing.
4) Spyware.
5) Shoulder surfing.
6) Social Engineering.

2.3.1 Brute force search

A Brute Force attack is a type of password guessing attack and it consists of trying every possible code, combination, or password until you find the correct one. This type of attack may take long time to complete. A complex password can make the time for identifying the password by brute force long. The main defense against brute force search is to have a sufficiently large password space. Text-based passwords have a password space of 94^N, where N is the length of the password, 94 is the number of printable characters excluding SPACE. Some graphical password techniques have been shown to provide a password space similar to or larger than that of text-based passwords. Recognition based graphical passwords tend to have smaller password spaces than the recall based methods. It is more difficult to carry out a brute force attack against graphical passwords than text-based passwords.

The attack programs need to automatically generate accurate mouse motion to imitate human input, which is particularly difficult for recall based graphical passwords. Hence, a graphical password is less vulnerable to brute force attacks than a text-based password.

The advantage to exhaustive offline attacks is that with enough time and computing power, all passwords will be found. However, full search of large password spaces is limited in practice by the time or processing power available; searching only subsets is faster, but doesn't guarantee success. To minimize the threat of exhaustive attacks, the theoretical password space should be too large to search. Note that this is not the case for many recognition-based systems e.g., the most common con_guration of Passfaces has 9-image panels and 4 rounds, yielding only 94 = 6561 passwords. Often such systems require complementary mechanisms such as limiting the number of online guesses per account, or securely combining multiple mechanisms (e.g., TwoStep Authentication). Helping the defender, attacks may require obtaining the image set used, which

involves additional effort; the added barrier depends on the size of the image set and the methods required to access it.

2.3.2 Dictionary attacks

A dictionary attack is a type of password guessing attack which uses a dictionary of common words to identify the user's password. Since recognition based graphical passwords involve mouse input instead of keyboard input, it will be impractical to carry out dictionary attacks against graphical passwords. For some recall based graphical passwords, it is possible to use a dictionary attack but an automated dictionary attack will be much more complex than a text based dictionary attack. Therefore, graphical passwords are less vulnerable to dictionary attacks than text-based passwords.

One of the problems with text based passwords is dictionary attacks. Since recognition based graphical passwords involve the user to input using mouse instead of keyboard, it is impractical to carry out dictionary attacks against this type of graphical passwords. For some recall based graphical passwords, it is possible to have a dictionary attack, but it will be more complex. So, we can say that graphical passwords are less vulnerable to dictionary attacks than text-based passwords.

Dictionary attacks on graphical passwords follow a long line of attacks on text passwords. The original idea involved guessing passwords from a relatively short pre-compiled list (dictionary) of high probability candidate password, based on empirical data or assumptions about user behaviour. Massive dictionaries and powerful data structures have created a continuum from small dictionaries to prioritized brute-force attacks, with smart dictionary attacks combining time-memory trade-offs of exhaustive attacks with higher success probabilities of prioritized dictionaries, in some cases algorithmically generated .

In systems allowing user-choice, dictionary attacks exploit skewed password distributions resulting from certain subsets of passwords being more attractive to non-negligible sets of users. Attacks succeed as users select passwords from predictable, relatively small subsets of the theoretical password space known as weak password subspaces, which can be enumerated, are small enough to search, and contain a significant fraction of passwords chosen in practice. These are collectively modeled as an effective password space including passwords with predicted probabilities higher than some threshold. A theoretical space too large to be exhaustively attacked does not guarantee security; the effective password space must also be too large to search. The

challenge here is to understand what composes the effective password space, which remains an open problem even for text passwords. Many graphical password proposals fall to dictionary attacks due to predictable patterns in user choice, as we discuss next.

2.3.3 Guessing

Here a legitimate users access rights to a computer and network resources are compromised by identifying the user id/password combination of the legitimate user. One such method is to take a common word and perform certain actions on it. Using the word Dinosaur as an example, users often create passwords such as DiNoSaUr (by alternating upper and lower case), rUaSoNiD (by reversing the string), oSNaiUDr (by shuffling the string), D9n6s7u3 (combining numbers and letters).

An online guessing attack requires interaction with the live system; password guesses are entered in turn to see if login succeeds. For graphical as well as text passwords, defenses
may be aided by clever use of CAPTCHAs; increasingly delaying (e.g., doubling) system response time on successive incorrect guesses; or limiting, per user account, the number of incorrect login attempts allowed before disabling further attempts. The latter risks locking out legitimate users who forget their password, enables denial-of-service attacks which intentionally provide incorrect passwords, and is less effective against multi-account attacks.

In an offline guessing attack, attackers gain access to variable text and need not interact with the live system to verify guesses. Schemes vulnerable to offine attack are at higher risk than those requiring online verification, for equivalent password spaces: offine work is not visible, processing trial guesses can be quicker, and pre-computed data structures or special hardware may be used.

2.3.4 Spyware

Except for a few exceptions, key logging or key listening spyware cannot be used to break graphical passwords. It is not clear whether “mouse tracking” spyware will be an effective tool against graphical passwords. However, mouse motion alone is not enough to break graphical passwords. Such information has to be correlated with application information, such as window position and size, as well as timing information.

2.3.5 Shoulder surfing

Shoulder surfing refers to using direct observation techniques, such as looking over someone's shoulder, to get information. It is commonly used to obtain passwords, PINs, security

codes, and similar data. Like text based passwords, most of the graphical passwords are vulnerable to shoulder surfing. At this point, only a few recognition-based techniques are designed to resist shoulder-surfing.

Shoulder-surfing is a targeted attack exacerbated by the visual aspect of graphical passwords. As users enter login information, an attacker may gain knowledge about their credentials by direct observation or external recording devices such as video cameras. High-resolution cameras with telephoto lenses and surveillance equipment make shoulder-surfing a real concern if attackers target specific users and have access to their geographic location. Several existing graphical schemes believed to be resistant or immune to shoulder-surfing have significant usability drawbacks, usually in the time and effort required to log in, making them less suitable for everyday authentication. Multi-touch tabletop interfaces support novel approaches offering shoulder-surfing resistant properties. For some graphical passwords, multiple successful logins must be observed to deduce the full password (e.g., when only a subset of user portfolio images are displayed at each login, or if the shared secret is not explicitly revealed at login). Passwords in other schemes can be recovered from one successful login.

2.3.6 Social engineering

Comparing to text based password, it is less convenient for a user to give away graphical passwords to another person. For example, it is very difficult to give away 26 graphical passwords over the phone. Setting up a phishing web site to obtain graphical passwords would be more time consuming. Overall, we believe it is more difficult to break graphical passwords using the traditional attack methods like brute force search, dictionary attack, and spyware. There is a need for more in-depth research that investigates possible attack methods against graphical passwords.

Phishing is a form of social engineering attack ; users may be tricked to reveal credentials by any means, e.g., phone calls from a fake help desk or credit company. While such methods may require targeted background work (or knowledge of personal details in personalized attacks), this is often easier than otherwise breaking into a system.

Text passwords and alphanumeric information are relatively easy to share with colleagues or attackers. For graphical passwords, a frame of reference must first be coordinated to convey the password in sufficient detail for use. This security advantage (complicating social engineering attacks) has usability drawbacks, e.g., complicating password reset by phone, and safe backup storage of passwords. Despite the added difficulty, Dunphy et al. give preliminary evidence that

users can describe PassPoints passwords sufficiently to enable use by others. Other means of sharing a graphical password include taking photos, screen shots, and drawing.

2.3.7 Malware

Malicious software includes any unauthorized software installed or downloaded without a user's informed consent, e.g., computer viruses and worms, Trojan horse software including login spoofing, code silently installed upon visiting web sites, and mobile code (e.g., JavaScript, Flash components). Keystroke-loggers record keyboard input; mouse-loggers and screen scrapers capture mouse actions and record screen memory, to be sent remotely or made available for retrieval. Many graphical passwords require one or both a mouse-logger and screen scraper for capture, and often a keystroke-logger as well to collect usernames. Keystroke-loggers alone may suffice for schemes like Inkblot

Authentication, which use keyboard input only. If graphical passwords gain popularity, such malware will likely follow.

2.3.8 Phishing and pharming

Phishing attacks trick users into entering their credentials at a fraudulent website, e.g., by having the user follows a link, in an email or engineered to return as a search engine result. As noted earlier, phishing attacks on recall-based graphical passwords resemble those on text passwords. For phishing attacks on recognition-based or cued-recall systems, specific images must be presented to the user. To do so, a phishing site may conduct earlier server probes to collect the images, or may retrieve and relay information from the legitimate site, in a man-in-the-middle (MITM) attack. Pharming , an advanced form of phishing, subverts the

DNS system (by forged DNS responses or DNS cache poisoning) such that domain names are fraudulently resolved to the IP address of an attacker's site. Depending on the password scheme, recording one or more login attempts at a phishing site may provide suffcient information for an attacker to subsequently log in. With a MITM attack, attackers may also log in to the legitimate site at least once by hijacking a single correct authentication response during the attack.

2.8.9 Reconstruction

Some attacks involve password reconstruction instead of direct capture. For example one graphical password scheme designed specfically to resist shoulder-surfing, was shown to fall to a SAT (boolean satisfiability problem) solver, which reconstructs user secrets in a few secondson observing a small number of logins. In general, these and intersection attacks involve pooling leaked information gathered from observing or recording several logins for schemes in which the authentication response varies across login instances. Acoustic-based reconstruction attacks on text passwords, such as the password cracker of Berger et al., seem less suited to graphical passwords, though ideas from the reconstruction techniques may apply to graphical schemes involving text input.

2.8.10 Manipulation Attack (MA)

This attack exploits the file location algorithm of a file by creating another file with the same name as the protected and privileged file. Then system can then be manipulated once it accepts the fake file as a trusted application component and loads it instead of the original file. Applications tend to load external components or files such as system libraries and configuration files and should be protected again malicious manipulation attempts. Unfortunately, an attacker can create a file with the same name and place it in the directory that will be search before the legitimate directory is selected, especially if the application only locates using the filename.

MAJOR DESIGN AND IMPLEMENTATION ISSUES:

1. SECURITY: We have briefly examined the security issues in the above section.

2. USABILITY: One of the main points which favors graphical password is that it is easier to remember pictures than text strings. A major complaint of the users using graphical passwords is that the password registrations and log-in process take too long.

3. RELIABILITY: It is the major design issue related to graphical password technique, especially for recall-based methods. In this type of method, the error tolerances have to be set carefully.

4. STORAGE AND COMMUNICATION: It also requires much more storage space than text-based passwords. Thousands and thousands of pictures have to be stored in the centralized database. Network transfer delay is yet another problem arousing out of it.

TABLE-1

Password Scheme	Password Input	Recapitulation Power	Processing Speed	Authentication
Text based	Fast	Depend on length and type of characters combination	Fast; Complexity, N	Low
Birget	Fast Input	Low; when large number of objects involved	Slow; Complexity depends on size and type of pictures. Can be given as N!/K!(N-K)! (N is the total number of picture objects; K is the number of pre-registered objects)	High
PassFace	Take Longer than Text based	Easier to remember, but, prediction	N^K (K is the number of rounds of authentication, N is the total number of pictures at each round)	High, but, chance of dictionary attack
Goldberg	Draw with stylus on touch sensitive screen; time taking	Depends on drawing complicacy	High Password Space	Guess dictionary attack
DAS	Depends on type of input; Draw with stylus on touch sensitive screen	Depends on drawing complicacy	Space consuming	Dictionary attack
User Authentication by Secured Graphical Password Implementation	Depends on size of password	Easy to remember	Minimum consumption due to digitization	Totally secured; Handwritten Characters are varied from person to person. Forgery Detection can be incorporated

Comparison between different methods.

CHAPTER 3

GRAPHICAL PASSWORD AUTHENTICATION

Human factors are often considered the weakest link in a computer security system. Patrick, et al. point out that there are three major areas where human computer interaction is important: authentication, security operations, and developing secure systems. Authentication is to confirm the claimed identity of a human user.

3.1 Overview of Authentication Methods

The process of verification that someone is actually who they claim they are, i.e., the act of establishing or confirming something (or someone) as authentic which is also a process of establishing a level of confidence regarding a claim. Current authentication methods can be divided into three main areas as shown in Figure 3.1:

1) Token based authentication
2) Biometric based authentication
3) Knowledge based authentication
    a) Text based authentication
    b) Picture based authentication

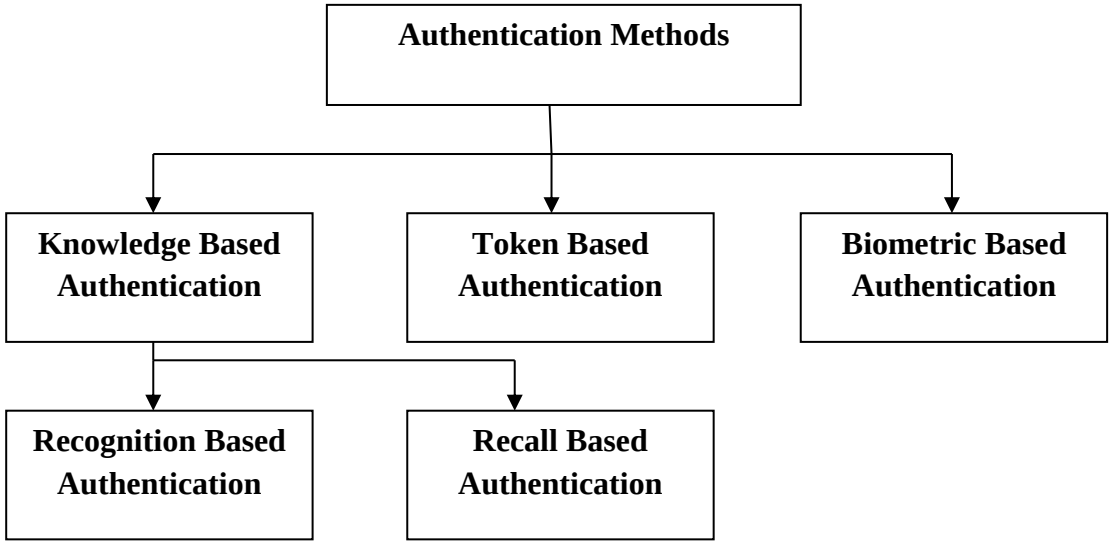

Figure 3.1 Authentication Methods

Token based techniques, such as key cards, bank cards and smart cards are widely used. Many token-based authentication systems also use knowledge based techniques to enhance security. For example, ATM cards are generally used together with a PIN number.

Biometric based authentication techniques, such as fingerprints, iris scan, or facial recognition, are not yet widely adopted. The major drawback of this approach is that such systems can be expensive, and the identification process can be slow and often unreliable. However, this type of technique provides the highest level of security.

Knowledge based techniques are the most widely used authentication techniques and include both text-based and picture-based passwords.

The picture-based techniques can be further divided into two categories:

1) Recognition-based graphical techniques
2) Recall-based graphical techniques.

In **recognition-based techniques**, a user is presented with a set of images and the user passes the authentication by recognizing and identifying the images he or she selected during the registration stage.

Example: **"Passface"** is a technique developed by Real User Corporation. The basic idea is that user will be asked to choose four images of human faces from a face database as their future password as shown in Figure 3.2. In the authentication stage, the user sees a grid of nine faces, consisting of one face previously chosen by the user and eight decoy faces. The user recognizes and clicks anywhere on the known face. This procedure is repeated for several rounds. The user is authenticated if he/she correctly identifies the four faces. The technique is based on the assumption that people can recall human faces easier than other pictures.

Disadvantages:

1) Most users tend to choose faces of people from the same race.
2) Female faces were preferred by both male and female users.
3) Better looking faces were more likely to be chosen.

All of these make the Passface password quite predictable.

Figure 3.2 An example of Passfaces

In **recall-based techniques**, a user is asked to reproduce something that he or she created or selected earlier during the registration stage.

Example:**"Reproduce a drawing",** Jermyn, et al. proposed a technique, called "**Draw - a - secret (DAS)"**, which allows user to draw their unique password as shown in Figure 3.3. A user is asked to draw a simple picture on a 2D grid. The coordinates of the grids occupied by the picture are stored in the order of the drawing. During authentication, the user is asked to re-draw the picture. If the drawing touches the same grids in the same sequence, then the user is authenticated. Jermyn, et al. suggested that given reasonable-length passwords in a 5 X 5 grid, the full password space of DAS is larger than that of the full text password space.

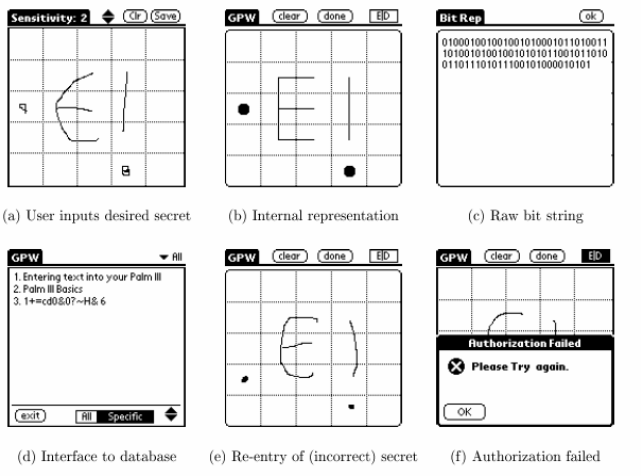

Figure 3.3 Draw-a-Secret (DAS) technique proposed by Jermyn, et al.

3.2 Graphical Password

Graphical password approach or sometimes also known as graphical user authentication is an alternative means of authentication that utilize images instead of text to login.

3.2.1 Classifications of Graphical Password System

Based on HaiTao, graphical password schemes depend on background being used, are divided into to two major groups:

1) Image based schemes which use images in the background further divided into subclasses based on the number of images displayed:

a) Single-Image based schemes: such as

V-GO is also a commercial product provided by Pass-logix Inc. V-GO allows a user to create a graphical password by navigating through an image. To enter a password, a user can click and or drag on a series of items within a single image as shown in Figure 3.4.

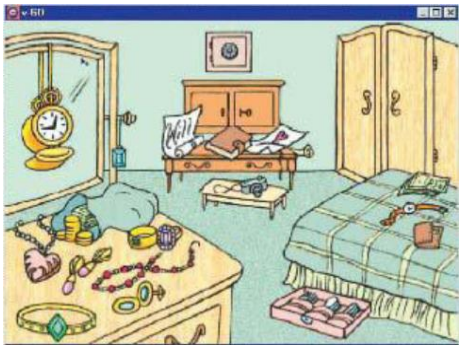

Figure 3.4 V-GO

For example, a user can take clothes from wardrobe and put it on the bed, closing the door, or set a timer on the clock. Other similar settings include dialing a phone number, hiding things under the bed, setting the date and time on a clock, making a stock trade, or choosing a hand of cards. The main major drawback of V-GO is the password space is limited. There are only a few places that user can create their password. Thus, chosen password might be easily guessed. Besides guessing attacking, V-GO also could possible to encounter brute force and shoulder surfing attacks.

PassPoints scheme proposed by (Wiedenbeck et al., 2006). According to the authors, they extended Blonder scheme by eliminating the predefined boundaries and allowing arbitrary images to be used.

Figure 3.5 PassPoints

As a result, a user can click 5 times on any place on an image to generate a password. In order to get authorized, the user has to click on the similar places in sequence. Due to a rich picture may contain hundreds to thousands of memorable points; the password space produced by PassPoints is large because any pixel can be selected as a click-point in the image as shown in Figure 3.5. In terms of limitation, PassPoints also possible to encounter attack such as guessing, brute force, hotspot and shoulder surfing attacks.

b) Multiple-Image based schemes: such as

Passfaces Corporation produced a commercial product named Passfaces that uses the idea of faces images to login.

Figure 3.6 Passfaces

To create a password in Passfaces, a user is required to select 4 images of a face database as a password as shown in Figure 3.6. In order to gain access, the user has to view a grid of nine faces and identify one correct face which has been registered previously in each attempt for four continuous challenges

To prevent key logging, Passfaces has implemented a random mechanism to arrange the images in each challenge set. However, due to the limited images used and low number of attempt, Passfaces is possible to encounter attack such as guessing, brute force and shoulder surfing attacks. In addition, according to users in Passfaces are highly affected by race, gender and the attractiveness of the faces when they are selecting their password.

2) Grid-based schemes which uses a grid on the background in DAS scheme, led researchers to use grid in the background. Other methods in this group are:

a)Background DAS

Paul Dunphy and Jeff Yan from Newcastle University superimposed a background over the blank DAS grid as to create a brand new system called Background Draw a Secret (BDAS).

Figure 3.7 Background DAS

The mechanism of BDAS is exactly the same as DAS except with an aid of background image as shown in Figure 3.7. According to the authors, the purpose of superimposing a background over the blank DAS grid is helping users to remember where they began the drawing that they are using as a password. By using this method, they believe that users are able to create passwords that are less predictable and more complex. Besides, based on the user study that has been carried out by the authors, they are able to show that the background image feature can reduce the symmetry effect and lead users from creating longer passwords.

b)Multi Grid DAS

Multi-Grid DAS proposed by Konstantinos Chalkias et al. (2006). Multi-Grid DAS was constructed with different cell size. In Multi-Grid DAS as shown in Figure 3.8, users have the opportunity to choose a grid from a list of predefined multi-grid templates.

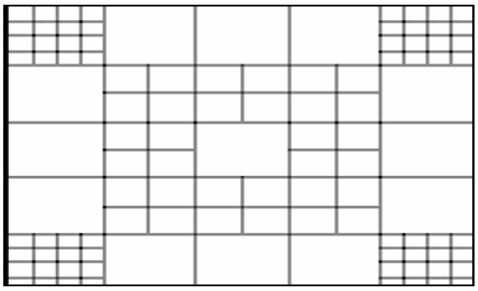

Figure 3.8 Multi Grid DAS

Users are allowed to draw their password in more than one area where the password can be centred to. However, the overall password produced using this method will have less symmetrical and centred effect. Apart from addressing the aforementioned issue, the Multi-Grid DAS also produces more password space compared to DAS.

c) Pass-Go

Hai Tao proposed a new grid-based scheme and named as Pass-Go. Pass-Go requires users to select intersections, instead of cells as shown in Figure 3.9, as a way to input a password. Consequently, the coordinate system refers to a matrix of intersections, rather than cells as in DAS. The password space of Pass-Go increases and users are able to draw a shape more freely and with less error tolerance compared with DAS.

This approach has some restrictions, i.e., Users have to choose their password within the sensitive areas and the number and location of the sensitive areas are fixed.

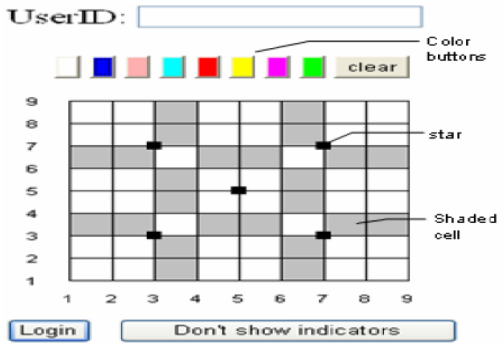

Figure 3.9 Pass-Go

Sensitive areas are invisible. Users will not know if they have successfully selected an intersection until the dot or line indicator appears.

d) Background Pass-Go

Por et al. (2008) proposed a new scheme for graphical password named Background Pass-Go (BPG). The password creation and authentication mechanisms of BPG are exactly the same as Pass-Go.

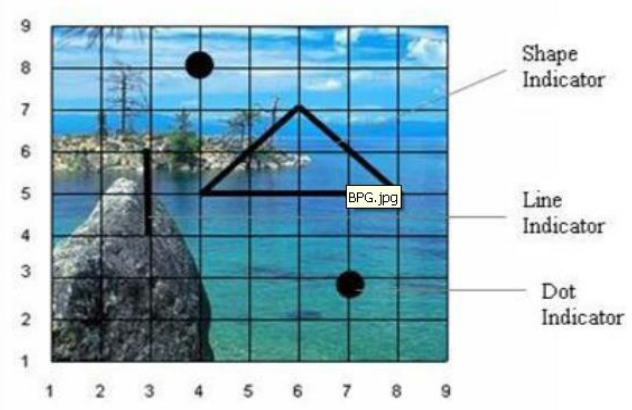

Figure 3.10 Background Pass-Go

BPG keeps most of the advantages of Pass-Go except shaded cells which are replaced by an image behind the grid lines as shown in Figure 3.10. Based on the heuristic testing carried out by the authors, BPG uses smaller radius size (area which surrounded intersections) compared to Pass-Go.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 JAVA Technology

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at Sun Micro system, Inc. in 1991. This language was initially called as "Oak" but was renamed as Java in 1995. The original impetus for Java was not the Internet. Instead the primary motivation was the need for a platform –independent (i.e. architecture neutral). To overcome the problem of time-consuming and expensive Compilers, the more cost efficient solution was Java. With the emergence of the World Wide Web, Java was propelled to the forefront of computer language design, because the Web also demanded portable programs.

While developing a project, it is significant that the technologies used for development must be reliable, flexible and robust. In case of a customer support tool application like this, the matter is more important because it details with a large number of clients and handles confidential data. After comprehensive analysis, it was found that Java and related technologies are more suitable for customer support tool applications since java has many features set that allow it to be an effective platform for customer support tool.

In addition, sun has a strong understanding of the critical business issues necessary to consider for customer tool. Another reason that Java is used in variety of application Nodes is that Java language is better in the customer arena because of some key features. Java makes a fully-fledged Node-side development platform. Java has a solid infrastructure that provides a well-tested implementation of many common applications needs such as security and messaging. The specifications outline the various components needed within Java and the technologies of Java for accessing and providing services and even the roles played during the development, deployment and runtime lifecycle.

When Java-Compatible Web browser is used, it can safely download Java applets without fear of viral infection or malicious intent.

4.1.1 Features

The Java programming language is a high level language that can be characterized by all of the following features. In short Java is

- **Simple:** Java was designed to be easy for the professional programmer to learn and use effectively. If you already understand the basic concepts of OOPS, learning Java will be even easier.

- **Secure:** The key that allows Java to solve security problems just described is that the output of a Java compiler is not executable code. Rather, it is byte code.

- **Portable:** The key that allows Java to solve and the portability problems just described is that the output of a Java compiler is not executable code. Rather, it is byte code. Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. This reason is straightforward: only the JVM needs to be implemented for each platform.

- **Object-Oriented:** Java is Object Oriented programming language. The object model in Java is simple and easy to extend.

- **Robust:** The ability to create robust programs was given high priority in the design of Java. In a well written Java program, all run-time errors can and should be managed by your program.

- **Multithreaded:** Java was designed to meet the real-world requirement of creating interactive, networked programs. Java supports multithreaded programming, which allows you to write programs that do many things simultaneously.

- **Architecture-neutral:** Write once; run anywhere, anytime, forever.

- **Interpreted:** Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java byte code. This code can be interpreted on any system that provides a Java Virtual Machine.

- **Distributed:** Java is designed for the distributed environment of the Internet, because it handles TCP/IP protocols.

- **Dynamic:** Java programs carry with them substantial amounts of run time type information that is used to verify and resolve accesses to objects at run time.

- **High Performance:** While the performance of interpreted byte code is usually more than adequate, there are situations where higher performance is required. The byte codes can be translated on the fly (at runtime) into machine code for the particular CPU the application is running on. For those accustomed to the normal design of a compiler and dynamic loader, this is somewhat like putting the final machine code generator in the dynamic loader. The byte code format was designed with generating machine codes in mind, so the actual process of generating machine code is generally simple. Efficient code is produced.

4.1.2 Why Java?

The most common types of programs written in the Java programming language are applets, servlets, java beans, java Node pages, window applications and console applications. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser. However, the Java programming language is not just for writing cute, entertaining applets for the web. The general purpose high level programming language is also a powerful software platform. Using the generous API, we can write many types of programs. An application is a standalone program that runs directly on the java platform. A special kind of application known as Node serves and supports clients on a network .Examples of Nodes are Web Nodes, proxy Nodes, Mail Nodes and print Nodes.

4.1.3 Java Programming Language

The Java Programming Language is a general-purpose, concurrent, strongly typed, class-based object-oriented language. It is normally compiled to the byte code instruction set and binary format defined in the Java Virtual Machine Specification.

4.1.4 J2SE Runtime Environment (JRE)

The JRE provides the libraries, Java virtual machine, and other components necessary for you to run applets and applications written in the Java programming language. This runtime environment can be redistributed with applications to make them free-standing.

4.1.5 J2SE Development Kit (JDK)

The JDK includes the JRE plus command-line development tools such as compilers and debuggers that are necessary or useful for developing applets and applications.

4.1.6 Java Virtual Machine

The Java virtual machine is an abstract computing machine that has an instruction set and manipulates memory at run time. The Java virtual machine is ported to different platforms to provide hardware- and operating system-independence.

4.2 Software Engineering Research

4.2.1 Methodology

The methodology used in this application is as follows:

- Swings
- AWT

Swings: Swing is a set of classes that provides more powerful and flexible components than are possible with the AWT. In addition to the familiar components, such as buttons, check boxes, and labels, Swing supplies several exciting additions, including tabbed panes, scroll panes, trees and tables. Even familiar components such as buttons have more capabilities in Swing. For example, a button has both an image and a text string associated with it. Also, the image can be changed as the stage of the button changes. Unlike AWT components, Swing components are not implemented by platform specific code. Instead, they are written entirely in Java and, therefore, are platform independent. The term light weight is used to describe such elements.

A Java toolkit for developing graphical user interface (GUIs). It includes elements such as menus, toolbars and dialog boxes. Swing is written in Java and is thus platform independent, unlike the Java Abstract Window Toolkit (AWT), which provides platform-specific code. Swing also has

more sophisticated interface capabilities than AWT and offers such features in the Java Foundation Classes (JFC) which are provided in the Java Developers Toolkit (JDK).

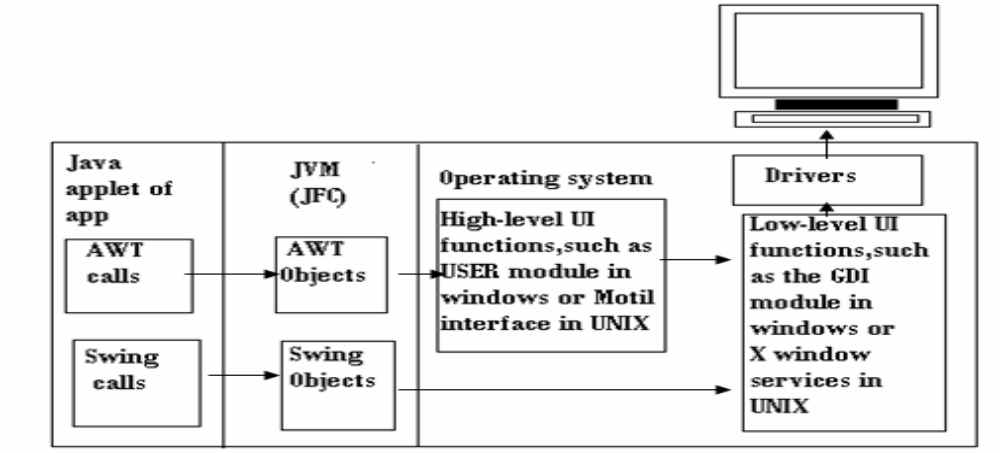

Figure 4.1 Java applet application model

AWT: (**A**bstract **W**indowing **T**oolkit) A class library from Sun that provides an application framework and graphical user interface (GUI) routines for Java programmers .AWT was the first user interface development system included in the Java Foundation Classes (JFC). In 1997, Swing was introduced, which provides more capability and is written entirely in Java.

4.3 JDBC

Java Database Connectivity or in short JDBC is a technology that enables the java program to manipulate data stored into the database. JDBC is Java application programming interface that allows the Java programmers to access database management system from Java code. It was developed by JavaSoft, a subsidiary of Sun Microsystems. JDBC is consists of four Components: The JDBC API, JDBC Driver Manager, The JDBC Test Suite and JDBC-ODBC Bridge. JDBC is an API specification developed by Sun Microsystems that defines a uniform interface for accessing various relational databases. JDBC is a core part of the Java platform and is included in the standard JDK distribution. The JDBC DriverManager class defines objects which can connect Java applications to a JDBC driver. DriverManager has traditionally been the backbone of the JDBC architecture. It is quite small and simple.

This means that the JDBC makes everyday database tasks easy. The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a Relational Database.

JDBC helps you to write Java applications that manage these three programming activities:

1. Connect to a data source, like a database
2. Send queries and update statements to the database
3. Retrieve and process the results received from the database in answer to your query.

4.3.1 Microsoft Access Database

Microsoft Office Access, previously known as Microsoft Access, is a database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. MS Access is a commercial Relational Database Management System (RDBMS) from Microsoft. MS Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases. Software developers and data architects can use Microsoft Access to develop application software, and "power users" can use it to build software applications. Like other Office applications, Access is supported by Visual Basic for Applications, an object-oriented programming language that can reference a variety of objects including DAO (Data Access Objects), ActiveX Data Objects, and many other ActiveX components.

In addition to using its own database storage file, Microsoft Access may also be used as the 'front-end' with other products as the 'back-end' tables, such as Microsoft SQL Server and non-Microsoft products such as Oracle and Sybase. Multiple backend sources can be used by a Microsoft Access Jet Database (accdb and mdb formats). Similarly, some applications will only use the Microsoft Access tables and use another product as a front-end, such as Visual Basic or ASP.NET. Microsoft Access may be only part of the solution in more complex applications, where it may be integrated with other technologies such as Microsoft Excel, Microsoft Outlook or ActiveX Data Objects.

4.4 Hardware Requirements

Processor : PENTIUM IV 2.8 GHz

RAM : 256 MB RAM

Monitor : 15” color

Hard disk : 1 GB

Keyboard : Standard 102 keys

Mouse : 3 BUTTONS

4.5 Software Requirements

Operating system : Microsoft Windows XP

Language : JAVA

Compiler : JAVA 6

Editor : JCreator

Libraries : AWT, SWING and other java libraries

Database : Microsoft Access

CHAPTER 5

DESIGN AND IMPLEMENTATION

This chapter gives a detailed description about the system design followed by the implementation. System design is a solution, a "how to" approach to the creation of a new system. This important phase is composed of several steps which provide the understanding of procedure details necessary for the implementation.

5.1 Design of Pass-Go

Pass-Go is a grid-based scheme. As an intersection is actually a point, which doesn't have an area, theoretically it would be impossible for a user to touch it without an error tolerance mechanism. Therefore we introduce sensitive areas to address this problem. The shape and size of the sensitive area can be predefined. In our implementation, sensitive areas are round circles with a radius of d*0.4 Sensitive areas are sensitive to the touch of an input device, and touching any point inside a sensitive area will be treated in the same way as touching the exact corresponding intersection point. Sensitive areas are invisible to users. The most obvious advantage of changing from cell to intersection is that drawing diagonal lines becomes feasible. Pass-Go contains G + 1 horizontal and vertical line.

5.1.1 Select Intersection Points

Pass-Go requires a user to select (or touch) intersection points, as a way to input a password. The coordinate system refers to a matrix of intersections. As an intersection is actually a point, which doesn't have an area, theoretically it would be impossible for a user to touch it without an error tolerance mechanism. Therefore sensitive areas are introduced to address this problem. A sensitive area is an area surrounding each intersection, as shown in Figure 5.1.

The shape and size of the sensitive area can be predefined. The larger the size of the sensitive areas is, the more easily an intersection can be selected, but the more difficult it is to avoid touching other neighbor intersections. Therefore, the optimal size of the sensitive area should be particularly studied and quantified. In this implementation, sensitive areas are round circles with a radius of $0.4 \times d$ (where d is the side length of a grid cell). Sensitive areas are sensitive to the touch of an input device, and touching any point inside a sensitive area will be treated in the

same way as touching the exact corresponding intersection point. Sensitive areas are invisible to users. The advantage of selecting intersection is that drawing diagonal lines becomes feasible, as shown in the letter "h" in Figure 5.1. A user can draw
a shape more freely with G × *G* grid in Pass-Go. It can implement a grid of larger size, thus offering stronger security.

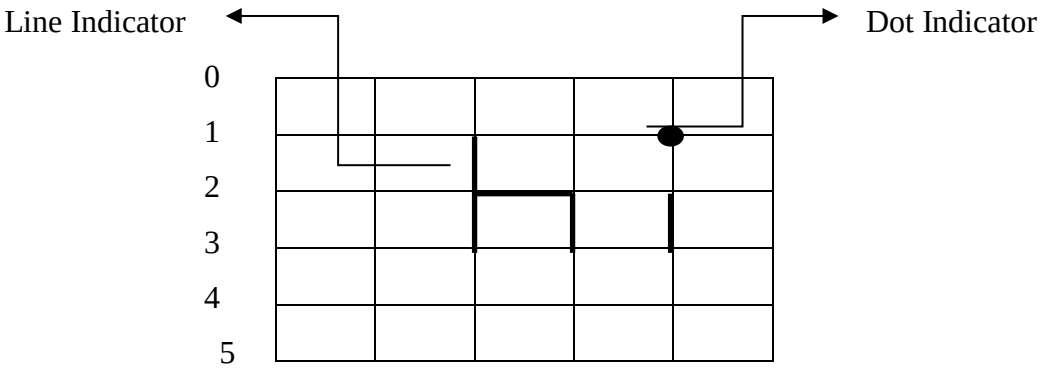

Figure 5.1 Pass-Go Design

5.1.2 Indicators

In Pass-Go, dot and line indicators are displayed to show the intersections and grid lines that correspond most closely with the input trace. A dot indicator appears when one intersection is selected (or clicked), and a line indicator appears when two or more intersections are touched continuously (by dragging the input device). The thickness and pat-
tern of indicators can be optimized to give the best visual perception effect for users.

5.1.3 Encoding

The password is encoded as a sequence of intersections, represented by two-dimensional coordinate pairs, with penup events, represented by (-1, -1)here, inserted into the place where breaks occur. For example, the password in the Figure 5.1 can be encoded as:

(1,2), (2,2), (3,2),(-1,-1), (2,2), (2,3), (-1,-1), (2,3), (3,3), (-1,-1), (2,4), (3,4), (-1,-1), (1,4), (-1,-1)

The definitions are as follows:

• The length of a password is the total number of co-ordinate pairs, excluding penups , in the encoding of a password;

• The stroke-count of a password is the total number of penups in the encoding of a password;

• The dot-count of a password is the total number of strokes of length 1;

5.1.4 Background Pass-Go

Background Pass-Go (BPG) is a grid-based scheme with an image background. BPG background is superimposed by an image selected by the users. This is purposely to help users to be able to remember their passwords better in such a way that they are able to remember where they started the password or which parts of image background consist of their passwords. Instead of having to remember the coordinate pairs of their passwords, the users are able to be more remindful towards their passwords by just recalling back which part of the image background they have clicked before as their passwords while at the same time the grids in this scheme provide better accuracy for their click points.

5.1.5 Grid Line Scaling

To a great degree, security of graphical password is affected by password space size. In most grid-based techniques, the password space size is determined mostly by the grid density, the number of strokes and the length of each stroke.

A scroll bar is used to control the grid line scaling of the MGBPG scheme. Due to the significant indicator size of 0.4×d radius, an end user can only manipulate the grid lines up to the scale of 20 lines each. When the grid line has been scaled more than 20 lines, the indicators which are selected by the user will potentially overlap. When an overlapping of the indicators has arisen, the user might have difficulty in identifying the correct password due to the poor vision and the messy GUI display of the system.

The slider will get inactive after the first click of user to prevent scaling for the moment of inputting password. For reactivating the sliders, user must reset screen and then readjust the sliders in a correct position.

5.1.6 Features Provided

- **Clear:** This clears all the point and line indicators selected as password by the user during login and registration phase.
- **Undo:** This removes point or line indicator which was previously drawn by the user.

- **Conceal:** It provides the user with an option to hide the password entered during login phase.
- **Reveal:** It helps the user in identifying the point and line indicators by making it visible.
- **ImagePanel:** This provides the user with the set of background images for selecting the password.

5.2 The Logic Flow

The flow involved with authenticating a user with the graphical password system is as follows:

Registration process:

Step 1: User is allowed to enter the Username.

Step 2: Checking for the Availability of the Username,

(a) If the entered Username already exists, goto Step 1.

(b) If the entered Username is Unique, goto Step 3.

Step 3: Enter the Password as follows

(a) User is provided with a scroll bar to control the grid line scaling.

(b) User has to select the intersection points using dot and line indicators.

(c) User has to select the background image for better security.

Step 4: On submitting the Password, user is registered successfully.

Login process:

Step 1: Enter the Username and Graphical Password.

Step 2: Check for Username and Password compatibility,

(a) If the entered username and password is valid, goto Step 3.

(b) If not, goto Step 1.

Step 3: Successful login.

5.3 Implementation

Pass-Go is implemented using Java Swings. Pass-Go requires a user to select (or touch) intersections, As an intersection is actually a point, doesn't have an area. Pass-Go is implemented based on a grid which is comprised of *G* horizontal and vertical lines. The gridframe class which extends JPanel contains grids of 300 (width)×300 (height)pixels.

A sensitive area is an area surrounding each intersection. Sensitive area takes a shape of the circle with $0.4 \times d$ (where d is the side length of a grid cell) as the radius. Sensitive areas are sensitive to the touch of an input device, and touching any point inside a sensitive area will be treated in the same way as touching the exact corresponding intersection point. Sensitive areas are invisible to users.

A dot and line indicators are displayed to show the intersections and grid lines that correspond most closely with the input trace. A dot indicator appears when one intersection is selected (or clicked), and a line indicator appears when two or more intersections are touched continuously (by dragging the input device).

An extra grid line scaling function has been added to ease users in remembering their passwords. There is a scroll bar which used to control the grid line scaling, with minimum and maximum slider value of 5 and 25 respectively. With the slider value of 5, user will get 5X5 intersection points, as the gridline scaling increases, number of intersection points will also increase and hence security is enhanced.

To improve usability, "clear" button is added which can erase all the previously inputted strokes. In addition, conceal and reveal button are deployed to switch from "not show indicators" to "show indicators" mode and vice-versa.

Background can be superimposed with an image which can be selected by the users. ImagePanel provides the user with the set of images for selecting the background image for the password.

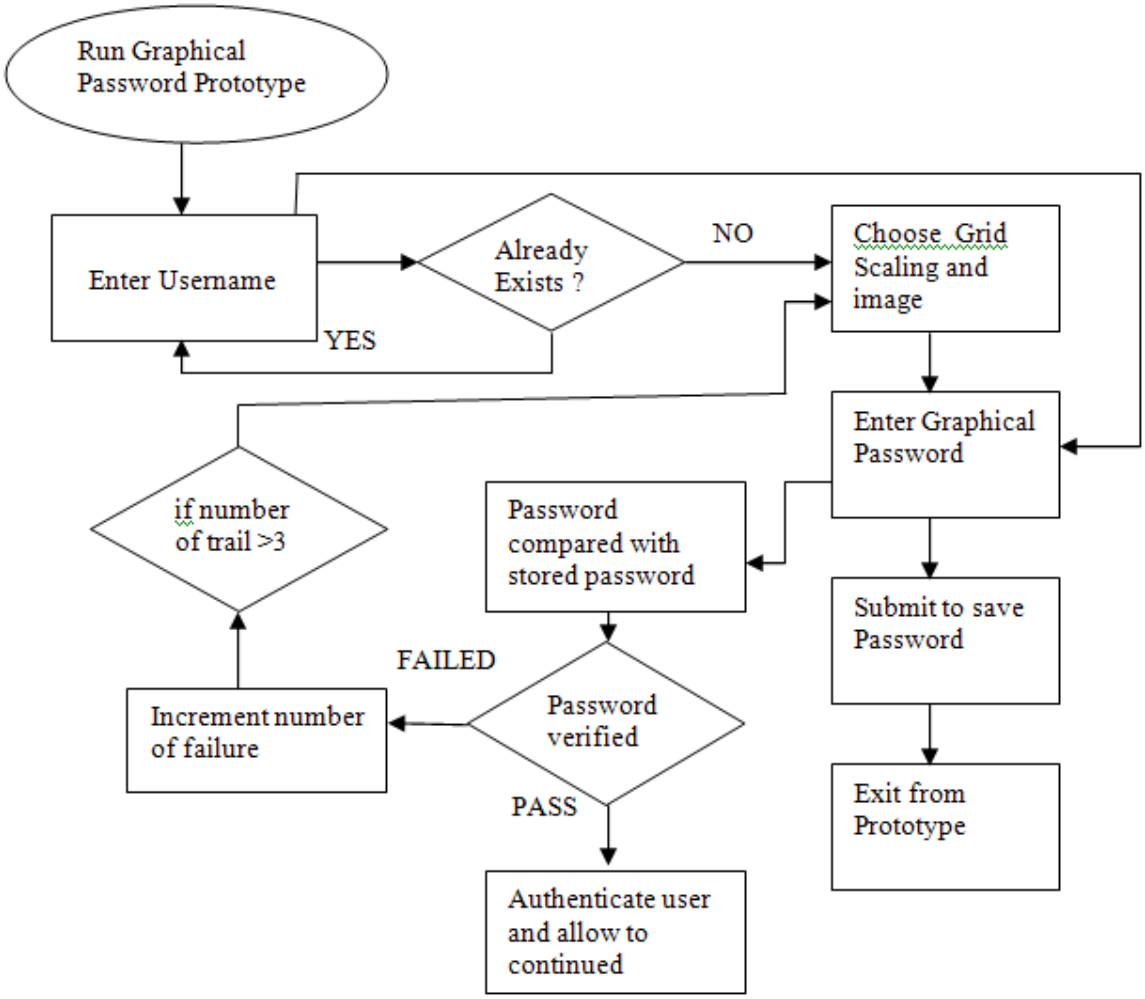

Figure 5.1 Pictorial Representation of Authenticating user

5.4 Library Functions

Functions

- **PaintComponent():**This enables **PaintPanel** to write directly to the surface of the component when Painting takes place.
- **Repaint():** Whenever there is a need to update the information displayed in its window, Repaint () is called.

Event Handlers

PaintPanel class implements the following interfaces

- **ChangeListener():**A change listener is registered on an object typically a component and the listener is notified when the object has changed. A change listener is most useful when it is necessary to know when an object has changed in any way. **stateChanged(ChangeEvent)** API is called when the listened-to component changes state.
- **MouseMotionListener():**This interface defines two methods. The **mouseDragged()** method is called multiple times as the mouse is dragged to draw lines . The **mouseMoved()** method is called multiple time as the mouse is moved.
- **MouseListener():**This interface defines five methods. If the mouse is pressed and released at the same point, **mouseClicked()** is invoked. When the mouse enters a component, the **mouseEntered()** method is called. When it leaves, **mouseExited()** is called. The **mousePressed()** and **mouseReleased()** methods are invoked when the mouse is pressed and released, respectively.

Classes

- **PaintPanel :** This class extends **JPanel.** It Overrides the **PaintComponent()** method so that grid lines are plotted in the Panel.
- **Grid frame :**It creates PaintPanel and then adds the Panel to the ContentPane. When the application is first displayed, the overridden **PaintComponent()** method is called and lines and dots are drawn.
- **Front:** This class extends JFrame. On clicking register button, registration window will be displayed and on login button, login window will be displayed.
- **Login:** It creates PaintPanel and then adds the Panel to the ContentPane. When the application is first displayed, the overridden **PaintComponent()** method is called and lines and dots are drawn.
- **MyData:** This module is used for interacting with the database which includes inserting, retrieving the user information which helps in saving and validating the authorized users.

User defined functions

- **search(int,int):** This function is used to get the position of each grid intersection in co-ordinate form and convert these to Matrix form.
- **addPoint(int,int):** This function adds points into a Hashset of type string to avoid duplicate values.

- **addtograph(HashSet):** This function generates intermediate points of each line, and adds to the arraylist of type string.
- **isSensitive(int,int):** This function checks whether the points and lines drawn by the user is within the sensitive area or not.
- **checkuname(String):** This function checks availability of the user in the database.
- **insert(String,String,String,int):**This function inserts the username, password points, image name, grid scaler in database.
- **validate(String,String,String,int):** This function validates the username and passwords during login process.

CHAPTER 6

RESULTS AND ANALYSIS

In this chapter, the Graphical Password scheme is analyzed by registering the new users and their login process and the corresponding results is shown with the following snapshots.

Figure 6.1 shows the main window of register and login

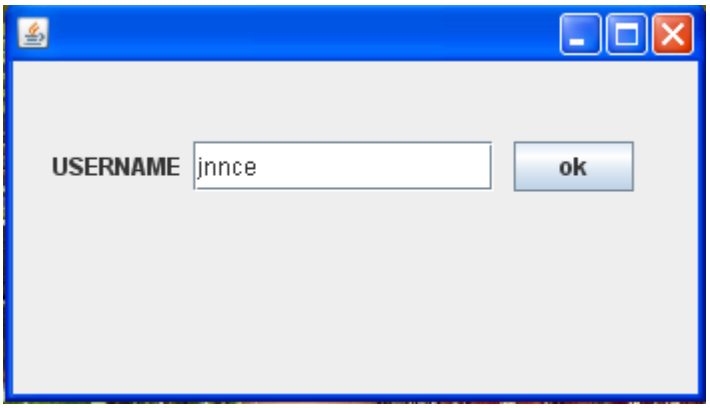

Figure 6.2 registering the username

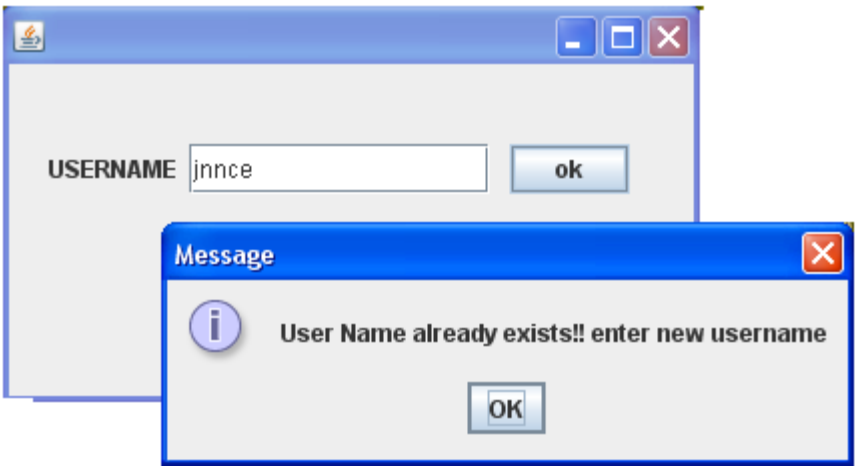

Figure 6.3 Alert message showing the existing username

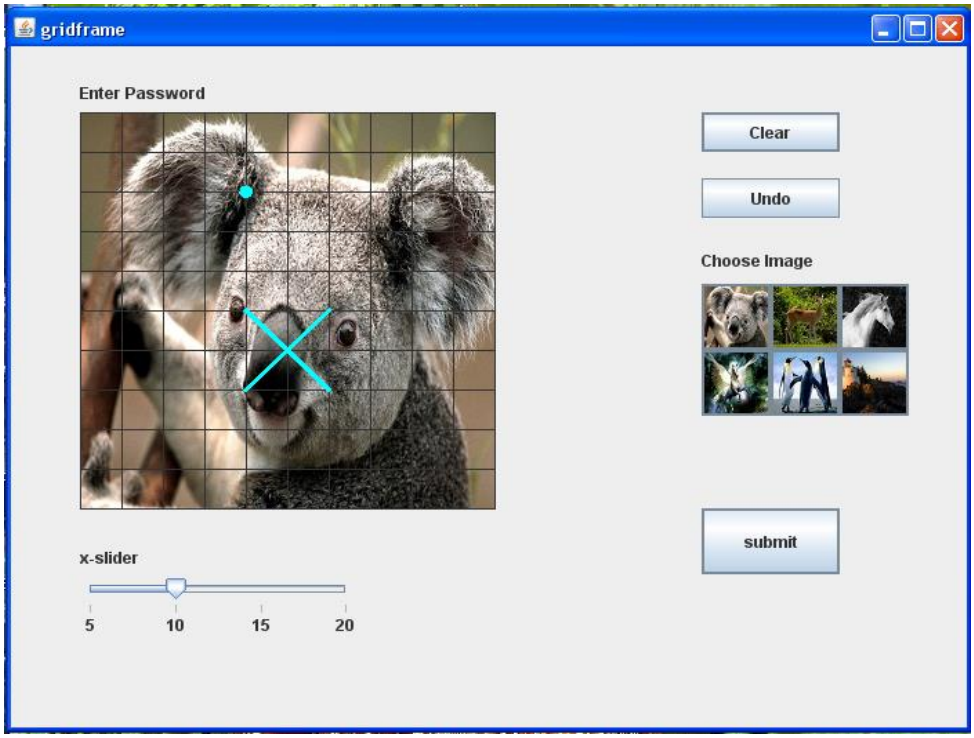

Figure 6.4 New user entering the Graphical Password

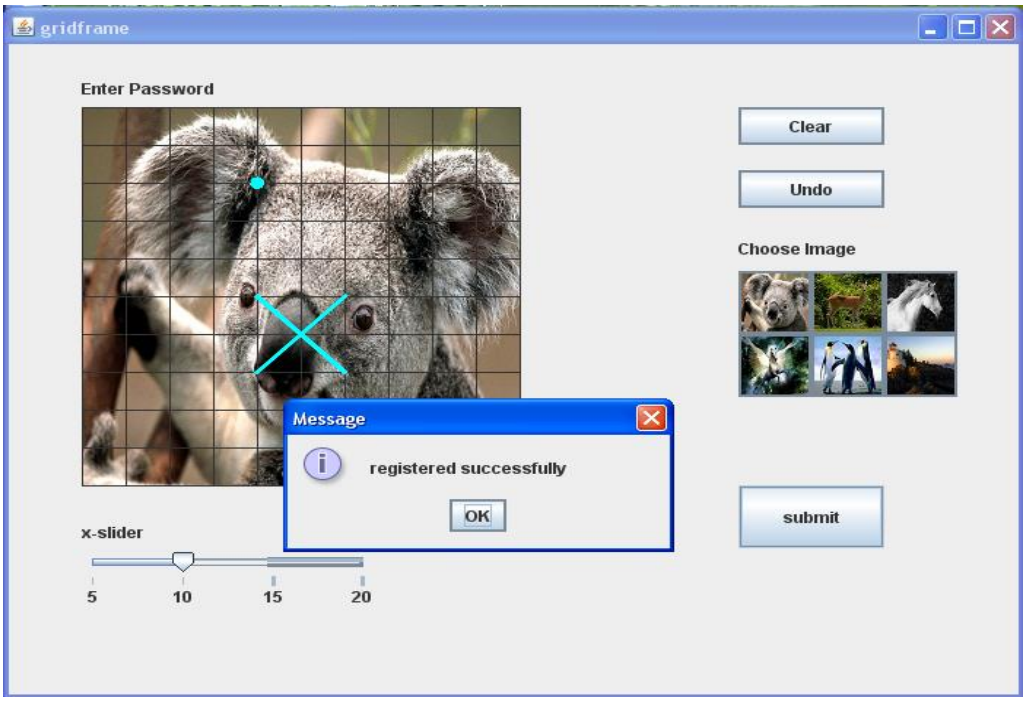

Figure 6.5 Message showing successful registration

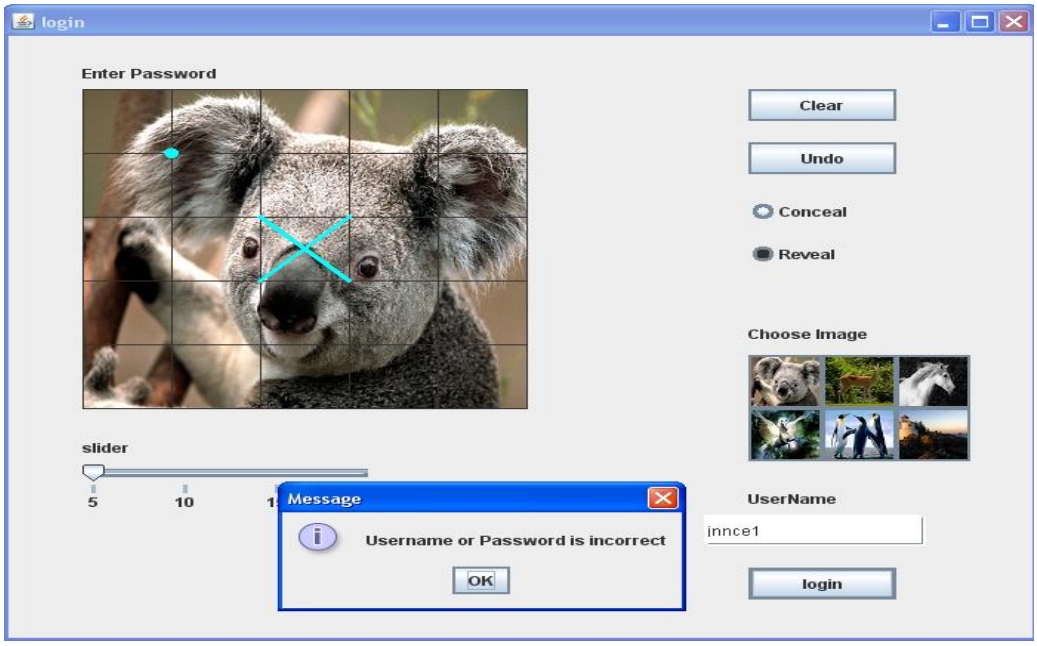

Figure 6.6 Message showing the incorrect password entered during Login

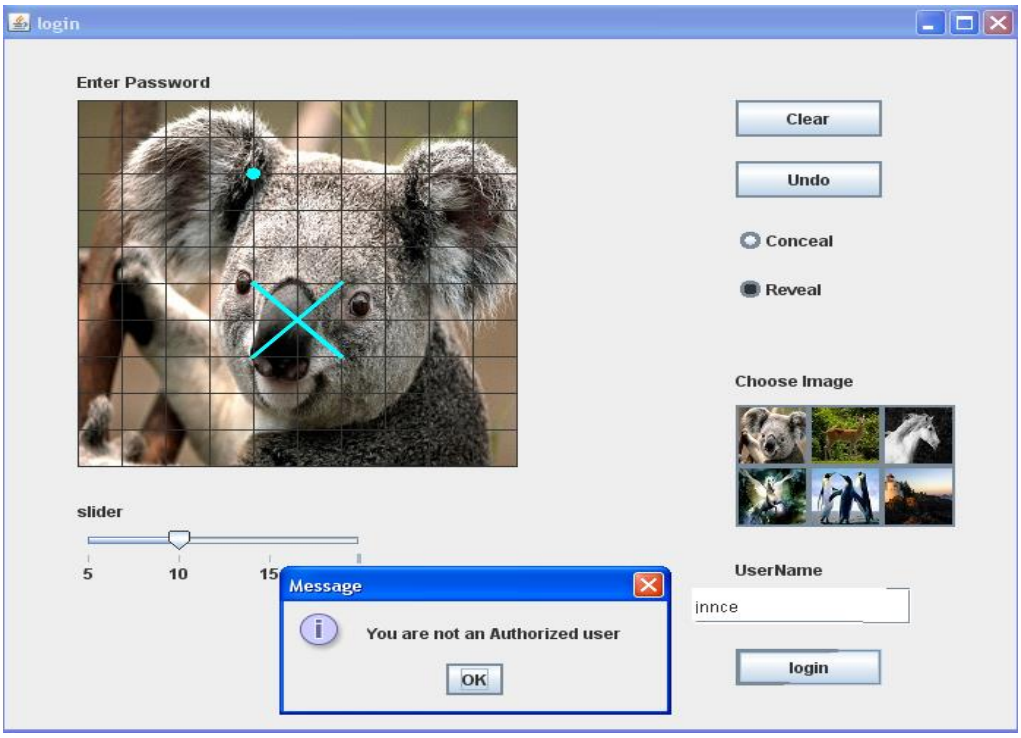

Figure 6.7 Alerting the user that he is not authorized

points

username	pts	iname	slider
sanju	0,0,-1,-1	koala.jpg	5
kaveri	2,7,2,6,2,9,2,8,2,3,2,2,2,5,2,4,2,1,2,10,-1,-1,5,2,5,1,5,10,5,8,5,6	pen.jpg	10
shruthi pm	1,1,-1,-1,6,4,-1,-1	c.jpg	10
shruthi	1,1,2,2,-1,-1,3,1,-1,-1,2,3,1,3,-1,-1	b.jpg	5
jnnce	6,7,4,5,5,6,-1,-1,6,5,4,7,5,6,-1,-1	koala.jpg	10
jnnce1	6,7,4,5,5,6,-1,-1,6,5,4,7,5,6,-1,-1,4,2,-1,-1	koala.jpg	10

Figure 6.8 shows how the Password is stored in database

The main window includes Registration and Login options for both new and existing users as shown in Figure 6.1. New users register their username as shown in Figure 6.2, if the username already exists then an alert message is displayed as shown in Figure 6.3.

The next step in registration process is to enter the graphical password as in Figure 6.4 and a confirm message is displayed as shown in Figure 6.5. Later if the existing users try to login with

incorrect passwords or username a warning is shown as in Figure 6.6, if the number of attempts exceeds three, message showing unauthorized user is shown as in Figure 6.7. All the information corresponding to the user like username, their graphical points, slider value is stored in database and is validated further in login process as shown in Figure 6.8.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Anew graphical password scheme is implemented in this project which offers better usability and memorability. This scheme manages to perform real time preprocessing by using Java Applet platform as ingenerating the grid line scaling and the indicators identification before the actual authentication process was taken place. It also manages to minimize the memorability issue by enabling each user to personalize their background image and the gridline scaling. Users are able to input one or more disguising dot or line indicators at random positions without having them being shown. With this feature, an attacker is unable to recognize the user's passwords although shoulder surfing attack.

Future Scope

In term of authentication, a user can still choose color code together with the correct sequence order of the indicators which have been set by the user before he or she can gain access into the system. A user can also be provided with a supportive sound signature to increase the remembrance of the password.

REFERENCES

[1] HaiTao, ”Pass-Go,a New Graphical Password Scheme “, Ottawa, Canada, June, 2006.

[2] Hai Tao and Carlisle Adams, ”Pass-Go: A Proposal to Improve the Usability of Graphical Passwords”, International Journal of Network Security,Vol.7, No.2, PP.273–292, Sept. 2008.

[3] L. Y. Por1, X. T. Lim2, F. Kianoush3, ”Background Pass-Go (BPG), a New Approach for GPS”, 12th WSEAS International Conference on COMPUTERS, Heraklion, Greece, July 23-25, 2008.

[4] KianoushFarhat, ”Graphical password system using scalable multi-grid method “ November 2010.

[5] L. Y. POR1, X. T. LIM2,”Multi-Grid Background Pass-Go”,Issue 7, wseas transactions on Information science & applications, Volume 5, July 2008.

[6] Susan Wiedenbeck Jim Waters , Alex BrodskiyNasirMemon, **“**Authentication Using Graphical Passwords**”,** Vol.6,December 2003.