Honours Project Report

Graphical Authentication for Secure Social Networks

Lebogang Mametja

Supervised by:

Dr. Anne Kayem

	Category	Min	Max	Chosen
1	Requirement Analysis and Design	0	20	0
2	Theoretical Analysis	0	25	0
3	Experiment Design and Execution	0	20	15
4	System Development and Implementation	0	15	15
5	Results, Findings and Conclusion	10	20	15
6	Aim Formulation and Background Work	10	15	15
7	Quality of Report Writing and Presentation	10		10
8	Adherence to Project Proposal and Quality of Deliverables	10		10
9	Overall General Project Evaluation	0	10	0
Total Marks		**80**		**80**

Department of Computer Science

University of Cape Town

2013

Abstract

Graphical password systems have received significant attention as one potential solution to the need for more usable authentication. However, to this date, they have not been studied in context. This paper explores the viability of graphical passwords as an alternative to text passwords in the context of social networks. To do that it implements and compares two graphical passwords schemes based on the principles of recall and cued-recall. It then compares these two graphical passwords to text passwords. This is done in order to evaluate usability in terms of login and password recovery, and robustness to guessing and capture attacks such as shoulder-surfing, social engineering and dictionary attacks.

This report finds that the graphical password scheme best suited to social networks is one that is based on the principles of cued-recall. This is because it performed in both the usability and security evaluations of our study. Our results also indicate that future work is required to make graphical passwords a viable alternative to text passwords for social networks.

Acknowledgements

I would like to thank my supervisor, Dr. Anne Kayem, for her encouragement, patience and guidance. I would also like to thank my project partner, Dorothy Mhlanga, for her hard work, and for always inspiring me to do my best.

A special thanks goes to Wayne Rotondwa Ratshidaho from BSG for sharing his experience, knowledge and expertise with us, and for his assistance in this project. Another word of thanks goes to our participants for their enthusiasm and willingness to help.

Lastly, I would like to thank Jesus, my Lord and God for the rock He has been to me this year, and the rock that He will always be.

Contents

Chapter 1 Introduction

Authentication systems can be characterised as “something you know” (e.g, passwords and PINs), “something you have” (e.g, token-based authentication), or “something you are” (e.g, biometric authentication). In addition, schemes can be characterised by properties such as the amount of training and setup required, the length of the authentication process, memorability, security, error rate and whether it can used universally in all environments [7].

Biometric authentication schemes may satisfy the limited training, fast setup and fast authentication process properties. However, it lacks the universal use, low error rate and security properties. For instance, voice authentication systems cannot be used in noisy environments, facial recognition systems are sensitive to lighting conditions and fingerprint systems can be bypassed through the use of fake fingerprints.

Physical or token-based authentication schemes satisfy the limited training, low error rate and high memorability properties. However, they are susceptible to theft and therefore do not satisfy the security property. This is because the attacker can easily impersonate a legal user as soon as the physical token is stolen.

Knowledge based authentication systems usually satisfy limited training and setup, fast authentication and universal use. The question is whether they offer a substantial amount of security, and how easy they are to remember. The limitations of human memory are the main challenges to knowledge-based authentication. Traditional passwords and PINs depend on recall-the ability to reproduce something that was created at an earlier time without help.

Graphical authentication schemes have been proposed on the premise that humans are better at retaining visual information. The schemes attempt to address the limits of human memory by relying on different cognitive processes. [7]

1.1 Motivation for Graphical Passwords

Traditional text-based password schemes are ubiquitous due to ease of use, inexpensive implementation, and user familiarity. However, they have the security and usability drawback of being typically difficult to remember, and they suffer from predictability if user-choice is allowed. This is because users tend to select weak passwords.

Graphical passwords have been proposed based on the observation that humans are considerably better at remembering images than they are at remembering text [7]. Psychologists have shown that with both recognition and recall tasks, images are more memorable than words or sentences [5]. As a result, much research has been inspired in both the security and HCI communities in recent years to explore graphical authentication systems as an alternative or an enhancement to text passwords [17].

As the name implies, graphical authentication uses graphics (pictures, icons, faces etc.) instead of the commonly used text strings [17]. There are three different types of graphical passwords: ones based on the principle of recall, recognition and cued-recall.

As previously stated, recall-based schemes require users to reproduce something that was created earlier during registration. Recognition-based schemes rely on the fact that the task of recognizing visual data is easier than recalling something from memory. Schemes depending on cued-recall require users to remember and target specific locations within an image. The image serves as a clue to aid the recollection task.

1.2 Problem Statement

The study of graphical passwords is a relatively young area of research, and the studies conducted have several limitations.

Firstly, there is a lack of comparison between the different types of graphical password schemes. Similarly, there have only been a limited number of studies comparing graphical and text-based schemes. The second problem is that there have been only a few studies conducted in the environment of use, which is necessary to enable realistic evaluations on the use of graphical passwords.

The last limitation is that none of the existing studies have been conducted in the context of social networks. This is important because performance constraints and goals differ depending on the intended environment of use. For example, in a high risk domain like a banking scenario, it would be acceptable for the system to be less usable in order to provide a high level of security. Similarly, social networks need to be very secure because hackers can use social networks as a point of information collection to deduce information that might lead to a strong password.

The focus of this project is to evaluate these authentication methods for secure social networks, with all testing and comparisons done in this context.

1.3 Research Questions

We will be focusing on two research questions, namely:

1. Which category of graphical password schemes is best suited for social networks: schemes based on recall, recognition or cued-recall?

2. Are graphical password schemes a viable alternative to text-based schemes as a means of providing authentication for secure social networks?

1.4 Thesis Contribution

The contribution of this research project is two-tiered. Firstly, we have implemented two graphical password schemes based on the principles of recall and cued-recall. Recall-based

schemes require users to recall something from memory; while cued-recall systems provide users with visual cues that help them remember their password. I implemented the recall-based scheme, Pluto. My project partner, Dorothy Mhlanga implemented the cued-recall scheme, ClickPoints. This is shown in Figure 1 below.

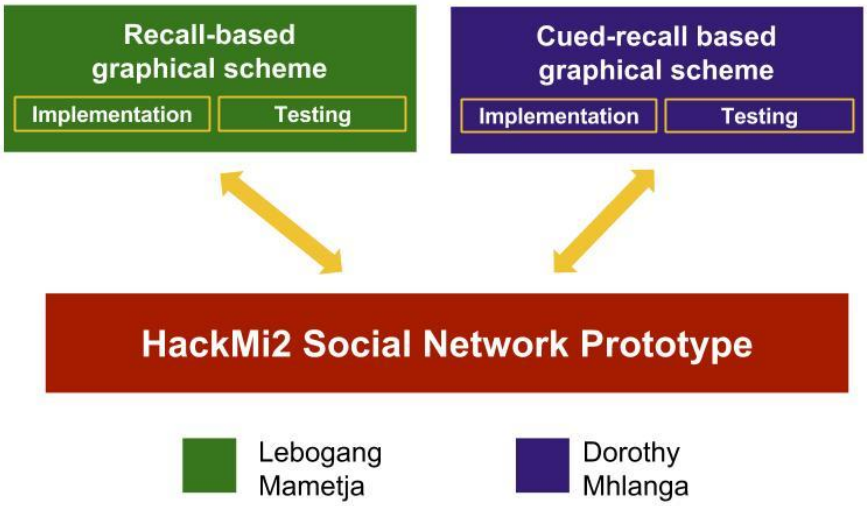


Figure 1: Work allocation

We had initially proposed that we would implement a recognition-based scheme in addition to the graphical password schemes mentioned above. However, this scheme was not implemented because it was found that recognition-based schemes cannot be used on their own because their level of security is only at the level of four or five digit PINS. This is discussed in more detail §2.3.

Pluto can be considered as an improvement on the original design called Draw-a-Secret (DAS). DAS and other related work will be reviewed in §2. The level of security offered in DAS is similar to that in Pluto. However, what makes Pluto stand out is that it inherits the strong points of DAS while achieving better usability. These advantages will be detailed in §3.

The second contribution of the project is to compare Pluto to the cued-recall based scheme developed by my project partner, called ClickPoints. In addition, both of these graphical password schemes will be compared to a text-based password scheme in order to evaluate usability in terms of password initialization and login; and robustness to guessing and capture attacks. All testing and implementation was done on a prototype social networking platform, HACKMI2 (http://www.hackmi2.cs.uct.ac.za).

1.5 Legal Acknowledgements

All software packages and libraries used in this project are open source. Elgg is an open source social network engine and MySQL is an open source database.

The University of Cape Town granted ethical clearance to conduct user testing. The ethical clearance was obtained from the chairman of the Science Faculty Ethics in Research Committee.

1.6 Thesis Outline

This thesis is organized as follows. In Chapter 2, the background of recall and recognition-based schemes will be presented. Next, Chapter 3 discusses the design of Pluto, while Chapter 4 describes the implementation of Pluto and its integration on to HACKMI2. Following this, is a description of the objectives, methodology and results of the user study in Chapter 5.

In addition, Chapter 6 discusses the implications of the results, while Chapter 7 will propose several variations of Pluto. Chapter 8 concludes this thesis with a summary and a discussion of future work.

Chapter 2 Background

2.1 Introduction

This chapter will discuss and review previous research on the three types of graphical passwords and compare them with text-based schemes.

2.2 Recall-based Graphical Passwords

2.2.1 Draw-a-Secret

Draw-a-Secret (DAS) [3] was the first recall-based system proposed. The authentication process consists of an N x N grid on which the user draws their password, as illustrated in Figure 2.1. The user-drawn password is encoded in the system as a sequence of coordinates of the grid cells passed through in the drawing. The password shown in Figure 2.1 would be encoded as:

(2,2) (3,2) (3,3) (2,3) (2,2) (2,1) (0,0)

with (0,0) representing a “pen-up” event.

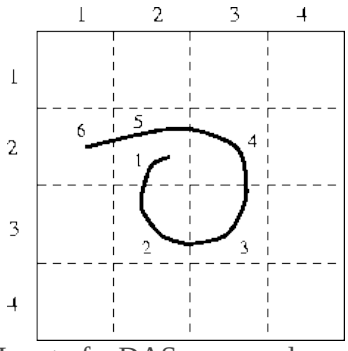


Figure 2.1: Input of a DAS password on a 4x4 grid [3]

This encoding of the DAS password lends itself to various security and usability problems. The first problem is that multiple passwords may have the same internal representation [5]. This is shown in Figure 2.2 - there would be no difference in the internal representation of a complicated symbol, a checkmark and a dot.

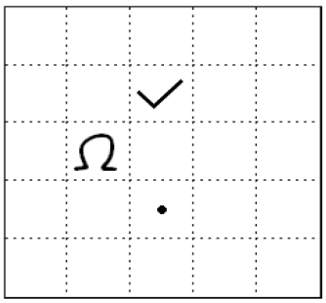
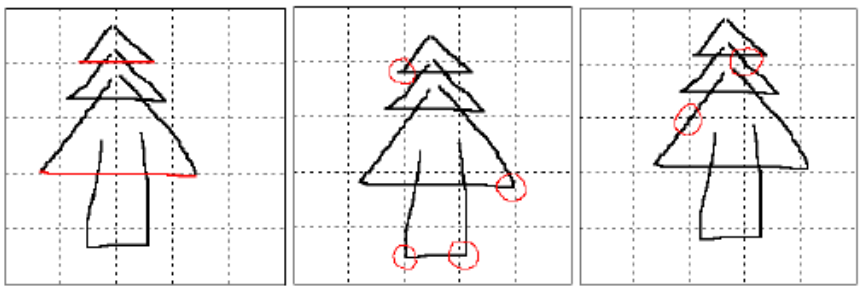


Figure 2.2: Internal representation of a DAS password [5]

Secondly, DAS enforces grid-crossing restrictions which may decrease usability. Illegal crossings are made by lines that cross near grid lines or through cell corners (see Figure 2.3). While the use of stylus pens on mobile devices may aid users in adhering to these restrictions, they may have more difficulty drawing their password on desktop computers. Moreover, a study by Dunphy and Yan [3] found that participants with non-technical background had more difficulty understanding the system due to these grid restrictions. As a result, various implementations and proposals have been made to address these shortcomings.



(a) (b) (c)

Figure 2.3: DAS rule violations: (a) lines crossing near grid lines; (b) end points near grid lines (c) crossings through cell corners [5]

2.2.2 Yet Another Graphical Password

Yet Another Graphical Password (YAGP) [5] is a scheme that aims to inherit the strong points of DAS while relaxing its strict grid restrictions. It aims to achieve those goals in several ways. Firstly, unlike DAS, the password does not consist of grid cell positions but instead reflects drawing trends. It adopts a trend-sensitive judgment mechanism when authenticating re-entered passwords so that users can redraw their password anywhere on the grid.

Secondly, YAGP is resistant to shoulder-surfing as it analyzes individual drawing style, making it harder for an attacker to imitate the legal user. Moreover, this high-level of security is enhanced by the 48 x 64 grid. However, the increased security may prevent legal users from authenticating successfully due to the difficulty of redrawing a password correctly. A screenshot of YAGP is shown in Figure 2.4 below.

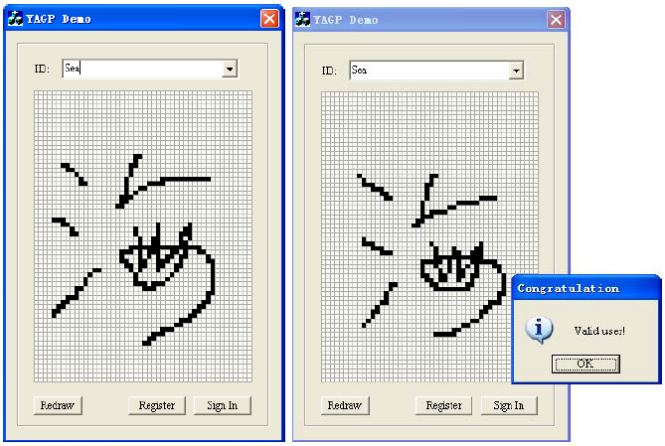


Figure 2.4: YAGP interface (48 x 64 density grid) [5]

2.2.3 Pass-Go

Pass-Go [14] (see Figure 2.5) is another scheme which aims to improve upon the original DAS design. In Pass-Go, users authenticate by drawing their password as a series of grid intersection points (instead of the grid cells in DAS). The Pass-Go password is encoded as a series of these points.

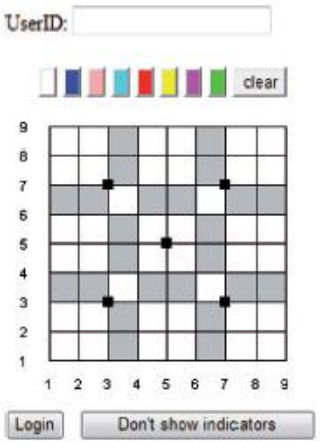


Figure 2.5: Pass-Go interface (9 x 9 density grid) [14]

Pass-Go employs several mechanisms to increase usability, memorability and scalability. Firstly, the intersection points have an error tolerance mechanism called sensitive areas

which aid the user in selecting their points. Secondly, dot and line indicators are displayed to show the grid lines and intersections that correspond most closely with the user's input. Lastly, reference aids in the form of stars and shaded cells appear on the Pass-Go grid. These mechanisms are illustrated in Figure 2.6 below.

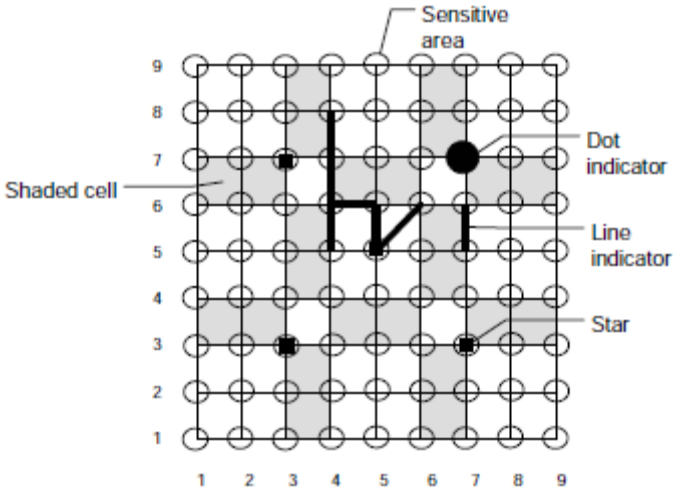


Figure 2.6: Pass-Go mechanisms [14]

The Pass-Go design offers several usability and security advantages. Translating the user's movement into grid lines and intersections eliminates the impact of small variations when redrawing a DAS password. It also eliminates the need to enforce grid restrictions. Security-wise, the theoretical password space is larger than that of DAS due to having a finer grid, allowing diagonal movements and having pen colour as additional parameter.

The 167-user study found that the login success rate was 78%. In addition, users suggested an "undo" option to erase only the most recent stroke instead of requiring the user to start all over [3, 14].

The paper further proposed several variations on Pass-Go that offered either better usability or increased security. Cell Indicators (see Figure 2.7) was one such proposal. The variation displays and encodes the cells that correspond most closely with the user's input (instead of grid lines and intersections). One drawback of this feature is that the cell indicators would clash with the existing indicators, making the login interface look 'messy'. Instead of accommodating the cell indicators by removing the existing ones, this variation can be thought of as a separate password scheme in its own right.

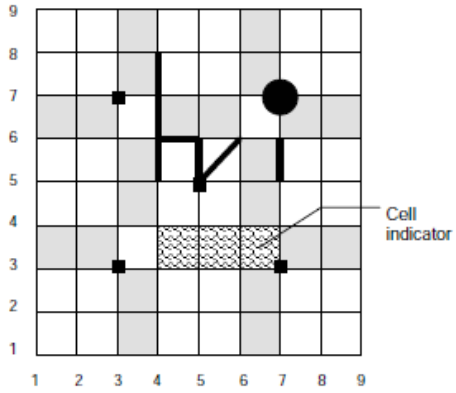


Figure 2.7: Pass-Go with cell indicators [14]

One drawback of Pass-Go is that tracing these grid lines and intersection points would require more effort and concentration when using anything other than a stylus pen. Some of the users in the study complained that it was difficult to draw when using a laptop touch pad or a mouse in their school labs.

Another limitation is that sensitive areas are invisible; therefore a user will not know whether a particular intersection has been successfully selected or not until the dot or line indicator appears. This suggests that users might have to spend a considerable amount of time getting accustomed to the system in order to avoid making unintentional errors (such as touching neighbouring intersections).

Generally, recall-based systems are less vulnerable to brute-force, dictionary, malware and social engineering attacks than text-based passwords [17]. This is because of the complexity of automatically generating mouse motion to imitate human input in brute-force and dictionary attacks. Mouse motion and keystroke loggers would not be effective on their own, but would be effective when coupled with screen scraper malware [1]. In terms of social engineering, screenshots, sketches or notes could be used to aid attackers.

2.2.4 Defence Techniques

Zakaria et al. [16] conducted a study on three shoulder-surfing defence techniques, namely Decoy Strokes, Disappearing Strokes and Line Snaking. The idea behind Decoy Strokes is to distract onlookers from the user's real password. To this end, decoy strokes are drawn at the same rate as the user and in a slightly different colour. The technique is illustrated in Figure 2.8. The study found that 63% of DAS passwords were stolen using this defence.

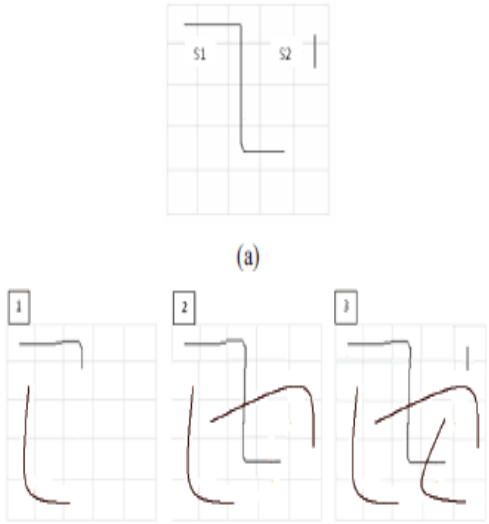


Figure 2.8: The Decoy Strokes defence: (a) when the defence is not activated and (b) when activated [16]

In Disappearing Strokes, the user's stroke is removed from the screen after being drawn. It is therefore optimal for multiple strokes (see Figure 2.9). With this approach, only 14% of passwords were stolen.

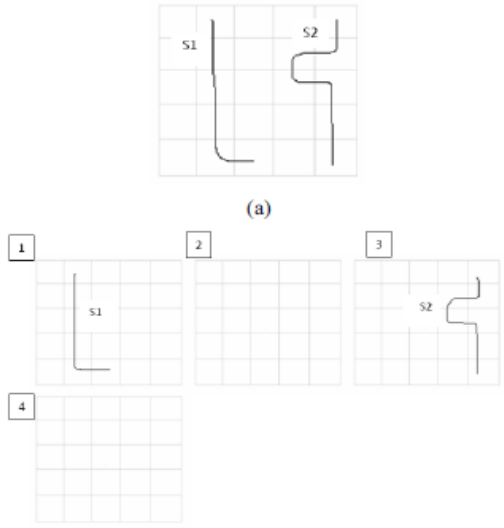


Figure 2.9: The Disappearing Strokes defence: (a) when the defence is not activated and (b) when activated [16]

In the last approach, the start of the user's password disappears as the user is still drawing the password, thereby preventing the attacker from seeing the complete user stroke on screen (see Figure 2.10). It is optimal for long singular strokes.

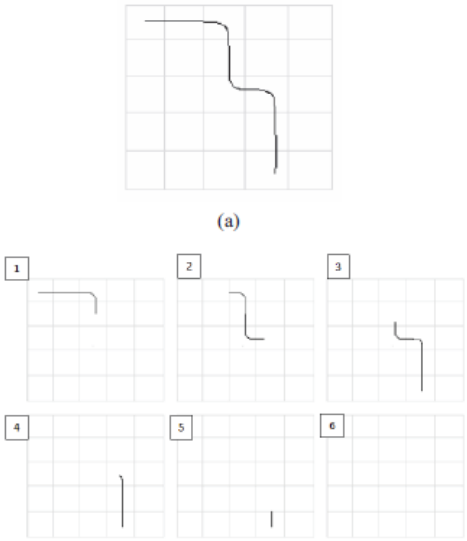


Figure 2.10: The Line Snaking defence: (a) when the defence is not activated and (b) when activated [16]

Line Snaking was found to be the most resistant to shoulder-surfing as none of the DAS passwords were stolen from this defence. However, it had the highest average login time and error rate at 8.3 and 1.3 seconds respectfully. Thus, the security improvement was achieved by sacrificing some degree of usability.

2.3 Recognition-based Graphical Passwords

2.3.1 PassFaces

PassFaces [2] is the recognition-based scheme most extensively studied. The authentication procedure involves users preselecting a set of human faces. During login, a panel of candidate faces will be presented to the user, who must select the face belonging to their set from among decoys images (see Figure 2.11). Several such rounds are repeated with different panels. For a successful login, each round must be executed correctly.

Figure 2.11: PassFaces [2]

Passfaces passwords have been found to be memorable over long-intervals. This can be accredited to the fact that human faces can be recognised more easily than other image types. It is also resistant to social engineering attacks because descriptive elements are stripped off the images leaving only faces which are hard to describe [16]. However, this can be circumvented if attackers successfully prompt users to take screenshots of their password.

Zangooei et al. [17] compared Passfaces with text-based schemes and found that Passfaces had a third of the login failure rate than text-based passwords. Suo et al. [13] found Passfaces passwords to be memorable over long intervals. Everitt et al. [4] found Passfaces to be a suitable alternative to text-based passwords where text-input was difficult or limited (for example, mobile phones). It was further found that it was resistant to social engineering because the things that stand out (and are thus descriptive) are cropped, leaving just a single face which is hard to describe.

The scheme has several usability drawbacks. Firstly, the task of having to scan many images in order to identify a few pre-selected images takes time, potentially making the login process slower than that of text passwords. Secondly, the system may be perceived as being intrusive if some faces are not welcomed by certain users. This may make the login process unpleasant. Zangooei et al. [16] further found that Passfaces required extra storage for storing images corresponding to each user and required extra maintenance of that database. Network transfer delays were of special concern as the system needed to display a large number of images for each round of the login process. Lastly, users who are face-blind and thus cannot tell faces apart will not be able to use the system [14].

2.3.2 Use Your Illusion

Hayashi and Christin [7] discussed another approach called Use Your Illusion (see Figure 2.12). It takes the task of recognizing visual data a step further by relying on the human ability to recognize even distorted versions of previously seen images.

Figure 2.12: Use Your Illusion [7]

The system allows user-chosen images which are then distorted to provide resistance to shoulder-surfing and social engineering attacks. The 99-user study revealed that users were highly skilled in recognizing degraded versions of their images, but that it took them 18 seconds on average to do so.

2.3.3 Multiple graphical passwords

Everitt et al. [4] conducted a 100-user study examining several factors that impact the ease in which users authenticate using multiple graphical passwords. The study examined the frequency of access, training, and memory interference resulting from interleaving access to multiple graphical passwords.

The 5-week online study used email-based prompts to encourage users to authenticate according to four different schedules. This method of testing would allow users to authenticate under more realistic settings.

The study revealed that participants who accessed passwords more frequently were able to authenticate in less time and with less attempts. Moreover, it was found that participants who trained on multiple passwords were more likely to fail than those who trained on a single one. Furthermore, participants who accessed four different password schemes a week where 10 times more likely to fail than ones who accessed only one.

Generally, recognition-based systems are less vulnerable to brute-force, dictionary, malware and social engineering attacks than text-based passwords [13]. This is because brute-force and dictionary attacks are more complex, requiring more effort and more system probes [2] than text-based ones. However, screen scrapers could be used in malware attacks while screenshots, sketches or notes could be used to aid social engineering attacks.

2.4 Cued-Recall based Graphical Passwords

2.4.1 PassPoints

PassPoints [2] is one of the first cued-recall schemes proposed. During login, the user is required to select five points on a system-assigned image. The system is shown in Figure 2.13. The re-entry of the click-points must be in the correct order and should be within the tolerance area around each click point. For example, in Figure 2.13 the five numbered boxes (which would not be visible to the user during login) illustrate the order in which the click-points must be entered. In addition, the size of the boxes indicate the tolerance area within which a point will be accepted as a click-point. The password is then the sequence of the user-selected click-points on a system-assigned image.

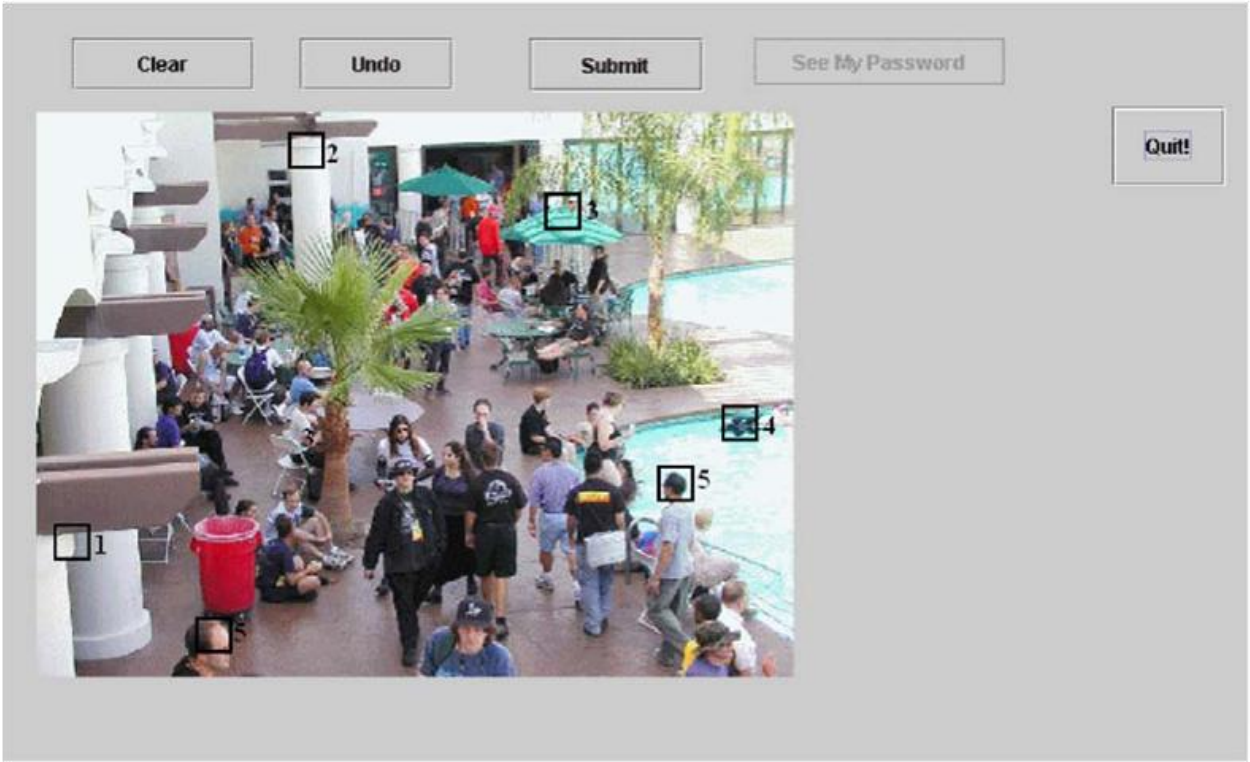


Figure 2.13: PassPoints with numbered user-selected click-points indicating the order. The boxes indicate the tolerance area [2]

The type of discretization used is an important implementation detail as it determines the size of the tolerance area. The three possible types are robust, centered and optimal discretization [2]. The user study found that the password creation time to be 64 seconds with the login success rate varying from 55-90%. A considerable amount of training was required for the user to be comfortable enough to start using the system. The study found that the training took 171 seconds on average. The average login time was only 9-11 seconds indicating that the authentication process did not take long.

2.4.2 Background DAS

Background DAS (BDAS) [3] is a scheme which introduces background images to DAS (see Figure 2.14). The background image acts as a cue to the user, helping them remember where to draw the lines through the grid.

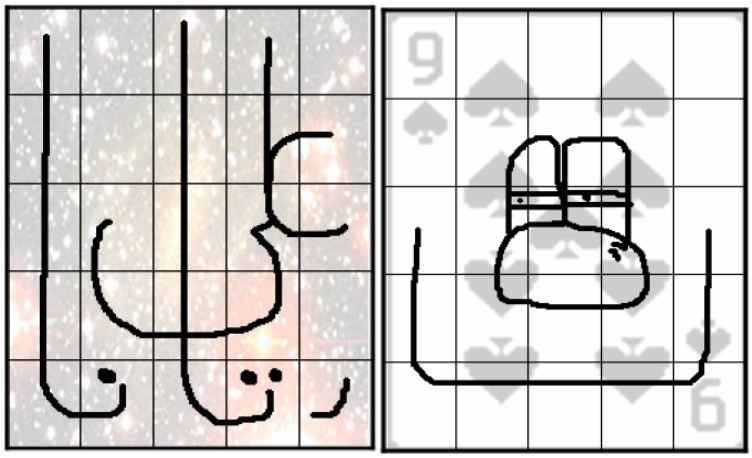


Figure 2.14: BDAS password scheme with background images introduced to encourage more complex passwords [3]

BDAS sought to improve password strength by adding background images with the aim of encouraging users to enter more complex, and less predictable passwords. Results showed that this goal was achieved as the average strength improved by 10 bits. Another result was that of improved memorability: users had the option of using these background images in one of three ways: either to map features to drawing, to use system as cued-recall, or both. Dunphy and Yan [3] further commented that a proactive password checker would be more useful in this context because users where more encouraged.

One limitation to the study is that it was implemented via a paper prototype and therefore issues such as interference and robustness to shoulder-surfing attacks could not be explored [3]. Another limitation is that it is not known whether the background images introduced other types of predictable behaviour such as targeting similar areas of the images or image-specific patterns [2].

2.5 Text-based Schemes

Text-based passwords are ubiquitous because other methods are costly or require special hardware [10].

Coping strategies used in text-based password schemes include writing down passwords and re-using passwords across accounts [11]. The former is not a serious threat anymore because network hackings have become the common form of attacks. The latter requires more attention as it translates into a single point of failure. Tari et al. [15] proposed a more sophisticated strategy called Pass-phrases, which involved using the first letter of a phrase to generate a

password; for example, a phrase like “My uncle and aunt have 12 cousins” would generate the following password: “Mu&ah12c”. The effectiveness of this strategy has not yet been analysed.

The strong text-based password systems were vulnerable because attackers aimed to capture each character one by one, not focusing on the meaning of the password. This contradicts views held by those who thought text-based passwords were defended against this by substituting asterisks for the password characters in the display as the user logs in [12].

Current text-based password reset and change policies are costly: Tari et al. [15] found that 30% of helpdesk calls were for password resets alone. This would cause a loss in worker productivity. Hong [8] suggests the review of budget cost of implementing needlessly high-secure policies.

Graphical passwords cannot be reset as easily as text-based ones because they are hard to describe. One solution is to assign a temporary non-graphical password during password reset, giving system access to create a new password.

In terms of shoulder-surfing attacks, Tari et al. [15] did a study to compare the perceived and real shoulder-surfing risk between text based and graphical passwords, and found text-based passwords to be more prone to this type of attack than graphical ones. The paper conducted a 20-user study to compare 4 configurations: mouse and keyboard entry for Passfaces, and strong and weak text-based passwords.

The results showed that the perceived and real risk were the same for the keyboard entry. This is because the attacker had to look at two places at the same time in order to steal the password: the screen and keyboard. Participants believed that graphical passwords were the most vulnerable due to their mode of entry (mouse-entry). However, the results contradicted this view and showed that strong text passwords were actually more vulnerable to shoulder-surfing attacks.

The strong text password systems were vulnerable because attackers aimed to capture each character one by one, not focusing on the meaning of the password. This contradicts views held by those who thought text-based passwords were protected against this by substituting asterisks for the password characters in the display as the user logs in [12].

Users with weaker passwords tend to enter their password faster than strong text password holders. Therefore, these results serve as an example of how usability could increase security. Speed of entry made it difficult for an attacker to capture weak text passwords. However, while shoulder-surfing resistant systems may be effective, they are slower on an already slow login process [9].

The study removed the myth that strong text passwords are universally better than weak passwords, and showed that it is important to look at the context. In the context of social networks, users login from anywhere including environments conducive to shoulder-surfing attacks such as public areas. Thus, shoulder-surfing resistant approaches are an important area of study for such a context.

There were several limitations to this study: Firstly, less dedicated approaches of real time viewing were used. No cameras or video recordings were used. Secondly, the study did not include a test on long-term memory recall and lastly, entry speed was constant but not controlled.

2.6 Summary and Discussion

Most recall-based schemes are variations of DAS that aim to inherit its strong points while addressing its shortcomings. YAGP adopts trend-sensitive mechanisms and analyzes user-drawing in an effort to give users more freedom when drawing their passwords. In contrast, Pass-Go, aims to provide that freedom by translating the user's movement into grid lines and intersections. This eliminates the need to enforce grid restrictions.

The mechanisms in Pass-Go that increased usability, memorability and scalability also increased security by allowing a finer-grained grid. Although the Cell Indicator variation clashed with the existing design, it could be thought of as a separate password scheme.

Of the several shoulder-surfing techniques studied by Zakaria et al. [16], Line Snaking was found to be the most resistant to shoulder-surfing. Furthermore, the ‘don’t show indicators’ defence technique implemented in Pass-Go did not appear to increase the difficulty of inputting a password.

Within recognition-based schemes, graphical passwords with distorted images were found to have equivalent error rates as their traditional counterparts. Passfaces passwords were found to be memorable over long periods and resistant to shoulder-surfing attacks. However, the login process took long and held the potential of being unpleasant for some.

The major drawback of Passfaces, UYI and most recognition-based systems is that they can rarely be used on their own. This is because their password space is comparable to only four or five-digit PINs. Those that have higher password spaces do so at the expense of usability. An example of this is the Cognitive Authentication scheme which requires extensive training, and has an average login time of 1.5 to 3 minutes [2]. As such, we did not implement recognition-based schemes but chose to rather focus on systems that offer better security and usability; namely, schemes depending on recall and cued-recall.

The novel ideas behind the methods and techniques that have been proposed have various advantages, and will be incorporated into the implemented schemes. The schemes will aim for compatibility on all devices- ones with both large and small screen displays.

Chapter 3 Design

Pluto is a 5x5 grid-based scheme which aims to improve upon the original DAS design. It is an implementation based on a proposed variation of Pass-Go which uses Cell Indicators. These indicators and other aspects of the design of Pluto will be discussed in this chapter.

3.1 Cell Indicators

Pluto differs from DAS in that it requires a user to select grid cells instead of drawing lines through them. As the user inputs his/her password, the chosen cell blocks are highlighted in yellow. This is shown in Figure 5.1 below:

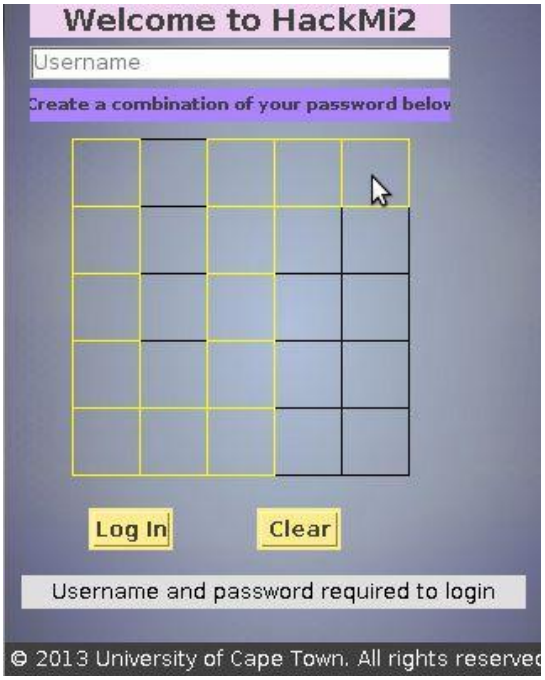
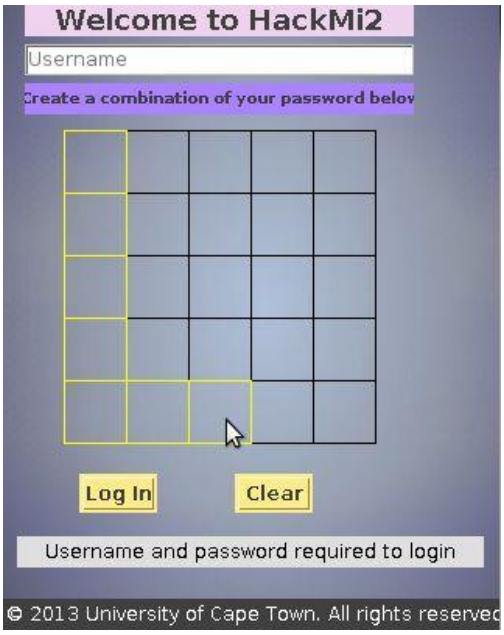


Figure 5.1: Pluto design

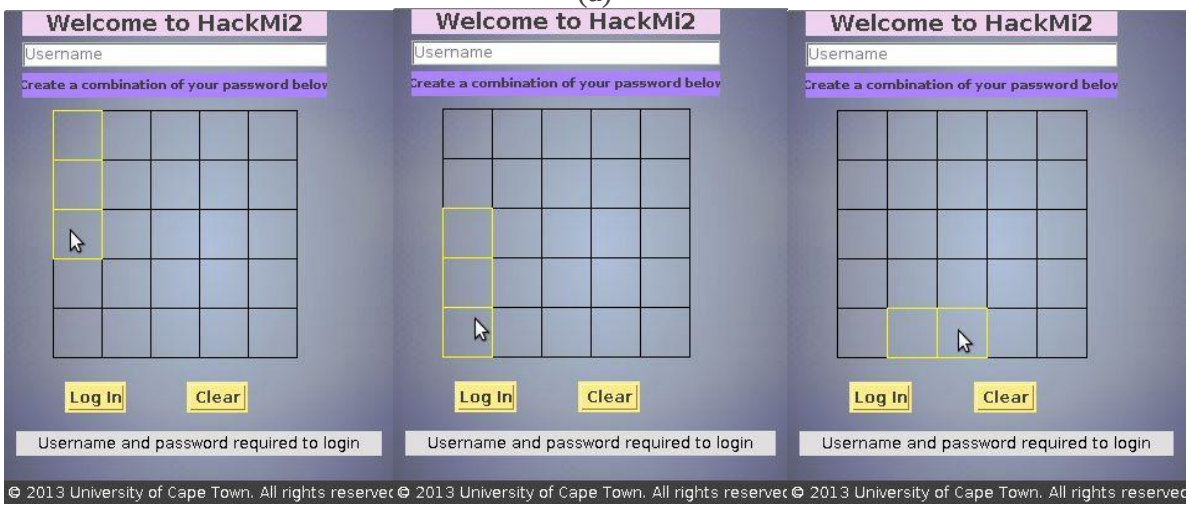
This approach has two advantages. Firstly, the indicators will accelerate the process of memorization as the user will not be inputting a slightly different input trace with each login. Secondly, this approach avoids the grid restrictions imposed on DAS, thereby making Pluto more usable.

3.2 Line Snaking

Pluto incorporates the Line Snaking shoulder-surfing defence technique. In this technique, the start of the user's password disappears as the user is still drawing the password, thereby preventing the attacker from seeing the complete user stroke on screen. This is shown in Figure 5.2 below.



(a)



(b)

Figure 5.2: The Line Snaking defence (a) when defence is not activated and (b) when activated

3.3 Encoding

The Pluto password is encoded the same way as the DAS password. The user-drawn password is encoded in the system as a sequence of coordinates of the grid cells passed through in the drawing. The encoding has the advantage of being efficient and human readable. The password shown in Figure 1.1 would be encoded as:

(1,1), (1,2), (1,3), (1,4), (1,5)

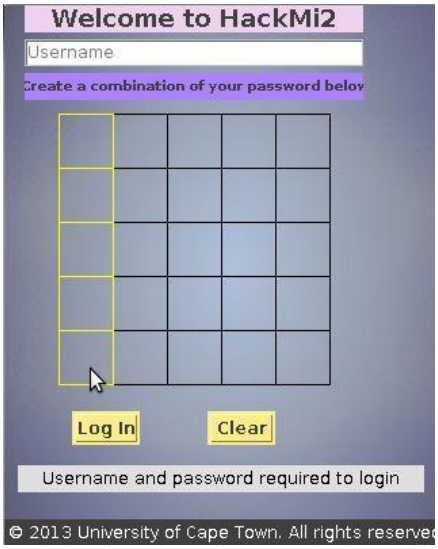


Figure 5.3: Pluto password

Chapter 4: Implementation

The chapter will discuss the tools and technologies that were used to develop Pluto. It will also detail the way in which Pluto was implemented in two iterations.

4.1 Tools and Technologies

4.1.1 Elgg

Elgg is an open source networking engine that provides the basic components needed to create a social network. It runs on the LAMP (Linux, Apache, MySQL, PHP) platform. Elgg can be easily extended through plugins. HackMi2, the prototype social network on which Pluto was is implemented, is powered by Elgg. The features, structures and components of Elgg will be described in §4.2.

4.1.2 Java

Pluto is an applet written in the Java programming language. Java was chosen as the language of use because Elgg allows Java applets to be embedded in its pages through plugins . The applets require Java runtime environment plugins to be installed in the web browser in order to run.

4.1.3 Apache

An Apache web server is used to host the HackMi2 website, and holds the resources and scripts required by the website.

4.1.4 PHP

PHP: Hypertext Preprocessor is a server side language used to create web pages. Since Elgg is written in PHP, certain aspects of the Java code had to be integrated with PHP. An example of this is the file-writing functionality that is needed to record the people who have logged in to the system.

4.1.5 MySQL

MySQL is an open source relational database management system. It is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack.

4.1.6 jarsigner

jarsigner is a JAR signing and verification tool which generates signatures for JAR files, and verifies the signature of signed JAR files. jarsigner uses key and certificate information from a keystore to generate digital signatures for JAR files. A keystore is a database of private keys and their associated X.509 certificate chains authenticating the corresponding public keys. The keytool utility is used to create and administer keystores.

4.2 Elgg Overview

This section will briefly describe the structure and components that make up Elgg, and the information we needed to begin development on Elgg.

4.2.1 Entities

Elgg is built on atomic data units called *entities*. Examples of entities include a user, a group or a blog entry. There is a base class in the Elgg framework called ElggEntity. All other entity classes extend that base class to provide different kinds of functionality. Related to this class are three other classes that make it easy to add new functionality. The ElggRelantionship class establishes connections between entities while the ElggMetadata and the ElggAnnotation classes attach information to entities.

4.2.2 Actions and Events

An *action* in Elgg is the code that runs when a user does something. For example, if the user logs into the site, the login action is called. Elgg events are triggered when something is created, updated or deleted or when the Elgg framework is loading. In addition, arbitrary events can also be triggered by any plugin.

4.2.3 Views

A *view* in the Elgg framework is responsible for creating a section of presentation code from input data. For example, the default *object/blog* view takes in as input an ElggObject entity of subtype blog and creates the HTML to display the blog title, blog

text, tags, and comments. Each component of a page is created by a view. For example, toolbars, headers and two-column layouts are all seperate views. Plugins, which will be discussed in the next section, can override any view.

4.2.4 Plugins

Elgg provides the functionality to add, extend or replace any of the features on the site. Plugins interact with Elgg through various mechanisms such as the ones previously described: actions, events and views. Plugins can override functionalities such as themes, blog engines, photo galleries, or in our case, the registration and login methods. We installed the JPatchwork 1.0 plugin to embed the Java Applets into the HACKMI2 site.

4.3. HACKMI2

HACKMI2 is a social network created from the Elgg v1.8.8 framework. It was created by Rotondwa Ratshidaho, Sanele Macanda and Molulaqhooa Maoyi in 2012 as part of a project that focussed on applying threat models on an open-source social network.

4.4 Iteration 1: Developing Pluto

Pluto was developed in two iterations. The first iteration involved developing a fully-functional Java applet that incorporated the design features described in §3. This was done using the Eclipse 3.7.2 IDE and the latest version of Java (which, at the time of writing was Java 7 update 45). The first iteration consists of the home page, the page to enter a password and the outcome page.

The home page allows users to either register, change their password, or log into the system. Figure 4.1 shows the home page of the first iteration of Pluto.

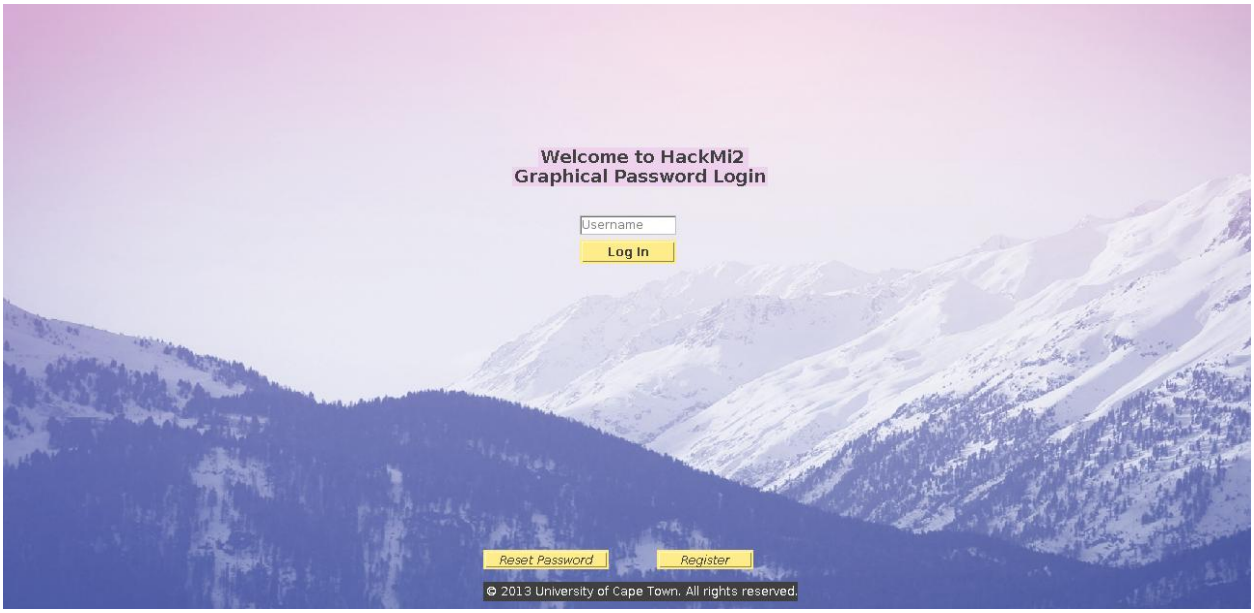


Figure 4.1: Home Screen of the first iteration of Pluto

The password page allows the user to enter their password. If the user is registering or resetting their password, they will be required to enter the password twice to confirm the password. The password is stored internally as Vector. The page for entering the password is shown in Figure 4.2 below.

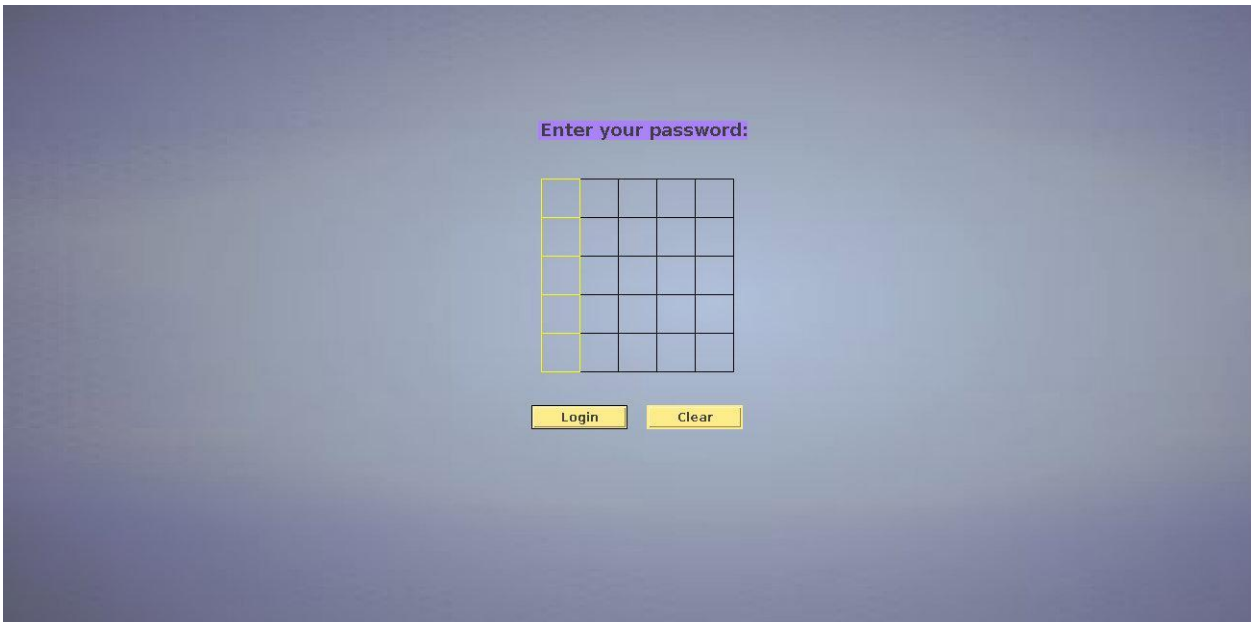


Figure 4.2: The page for entering the password for the first iteration of Pluto

Lastly, the outcome page informs the user if their desired action was successful, i.e if they successfully registered, changed their password or logged into the system. An example of the outcome page for a successful login is shown in Figure 4.3.

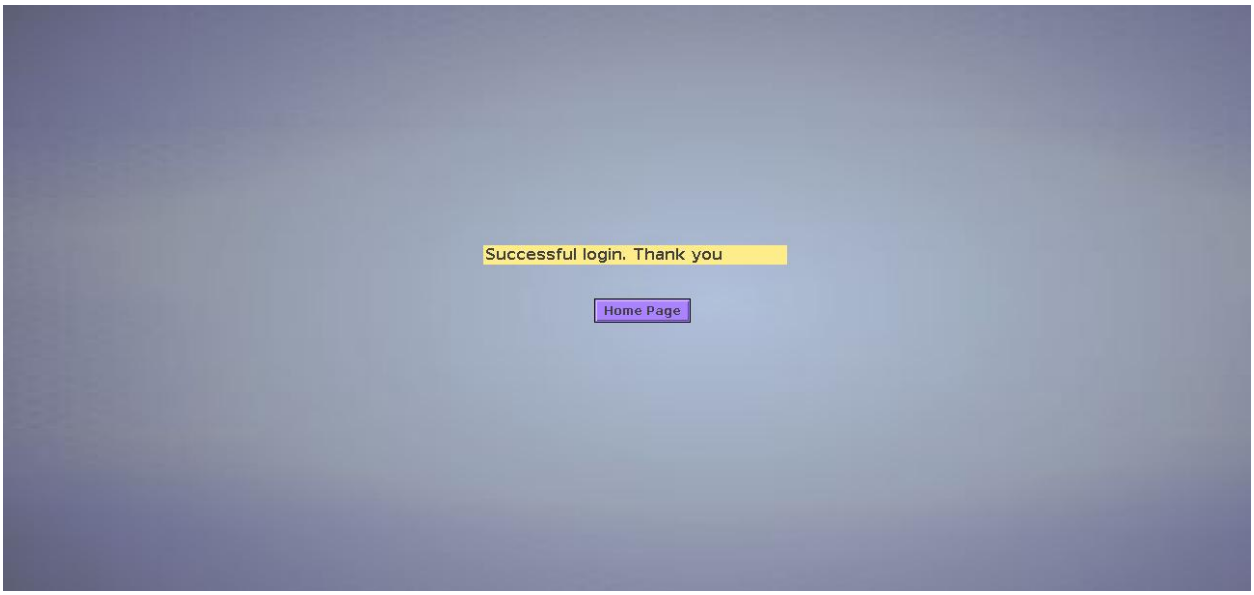


Figure 4.3: The outcome page of the first iteration of Pluto. The above screenshot shows the successful login outcome

4.6 Iteration 2: Integrating Pluto onto HACKMI2

The second iteration involved integrating Pluto on to HackMi2. To achieve this, we build web services using the REST API framework on Elgg to expose the login and set-password functionality on HACKMI2. In addition, various functions such as signing the applet had to be conducted in order for the Java applet to be fully functional on the HackMi2 site. These features and functions will be discussed further in this section.

4.6.1 Setting a Password

We created a method for setting a password on Elgg which takes the username and password, and then exposed it. We then called the API by performing a POST request to the server. The server will either set a new password or update depending on whether the user had previously created one or not.

4.6.2 Logging onto HACKMI2

To expose the login functionality of HACKMI2, we took the password, username and and an Elgg security token and performed a POST request to the server. The server takes the password and username, checks if the user exists and compares the password with the one in the database.

If the user does not exist in the database or if the passwords do not match, an error message is sent to the applet which displays the message.

4.6.3 Signing the Applet

By default, a Java applet is unsigned (not signed by a certificate), untrusted (not initiated by the user), and is a sandbox applet (runs in Java's security sandbox). This means that it cannot access the user's local file system, read certain system properties, nor can it open a socket connection to a server that is different from the web server that hosts the applet.

In order to eliminate these restrictions, the applet needs to be signed with a certificate.
There are two types of signed applets: ones which are signed by a trusted SSL provider and ones which are self-signed by the developer themselves. When the browser starts a signed applet, the applet will automatically request permission to run outside of the sandbox. This will be discussed in the next section (§4.3.3). If the user grants this permission, the applet will be able to run outside of Java's sandbox and will be able to access the necessary resources.

In order to relax the default restrictions on Pluto and ClickPoints, we self-signed the applet. To achieve this, we converted the Java applets into their respective JAR files. We used the keytool utility to create our certificates and then signed the JAR files using the jarsigner tool (refer to §4.1.5).

4.6.4 Java Application Prompts on HACKMI2

The security prompts that appear when a browser attempts to run a Java applet enable users to make informed decisions before allowing Java content to run in the browser. These application prompts ask the user for confirmation before running the Java applets.

The messages presented depends upon different risk factors. For example, running applet code that is not signed from a trusted Certificate Authority.

There are three different kinds of messages for the three previously described types of applets: signed, self-signed and unsigned. The dialogue that appears for Pluto is shown in Figure 4.4 below. This is the kind of message that is presented for all self-signed applets.

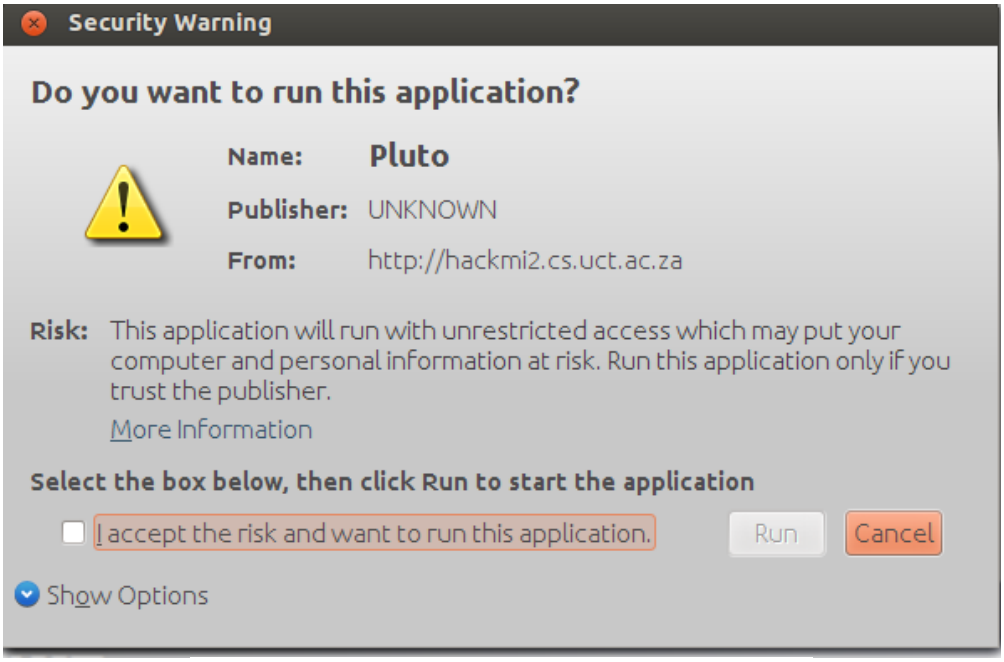


Figure 4.4: Security prompt for Pluto on HACKMI2

4.7 Using Pluto on HACKMI2

This section describes how a user can set or reset a Pluto password. In addition, it shows how the user can log into HACKMI2 with their Pluto password.

4.7.1 Setting the Password

The user is required to register on HACKMI2 using the original text-based registration method shown in Figure 4.5. When registered, the user logs into HACKMI2 with the username and text password they registered on the system.

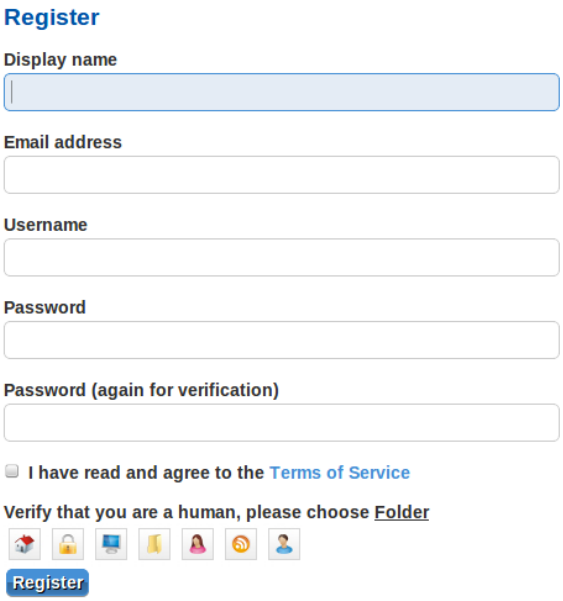


Figure 4.5: Original registration method on HACKMI2

Once the user is logged in, he/she can go to the Settings page. This page allows the user to change their text password. At the bottom of the page is the option of setting their graphical password, as shown in Figure 4.6. The user can also reset their graphical on the same page.

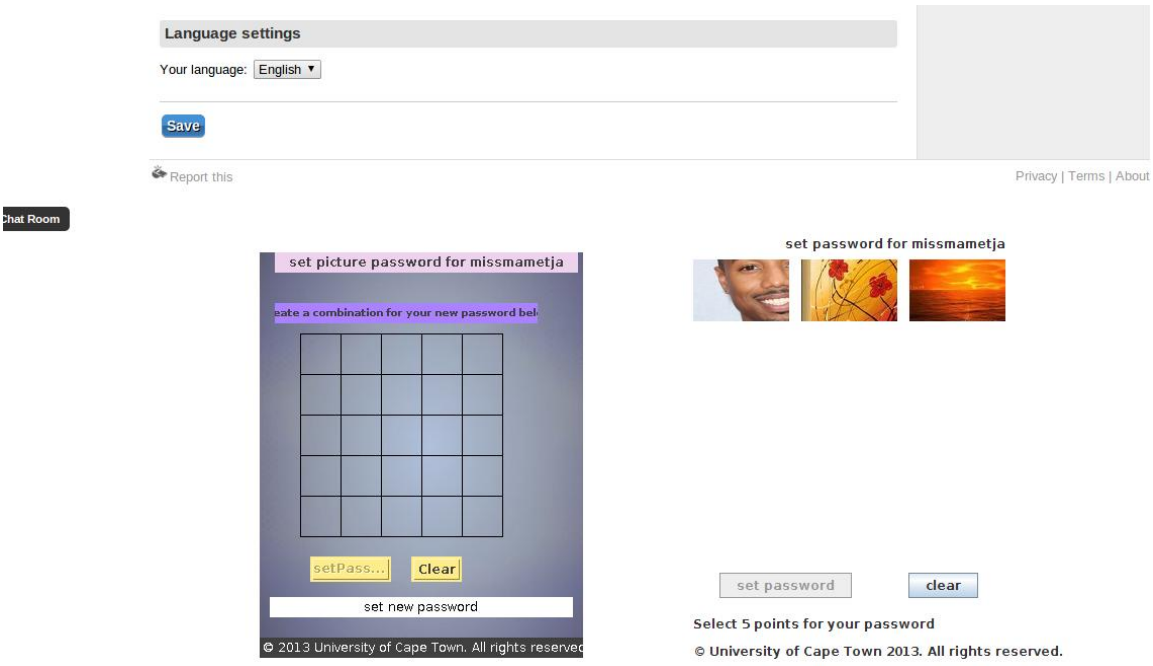


Figure 4.6: The Settings page on which the Pluto and ClickPoints passwords can be set

4.3.6 Logging into Pluto

The original login page before Pluto and ClickPoints were added to HACKMI2 is shown below in Figure 4.7.

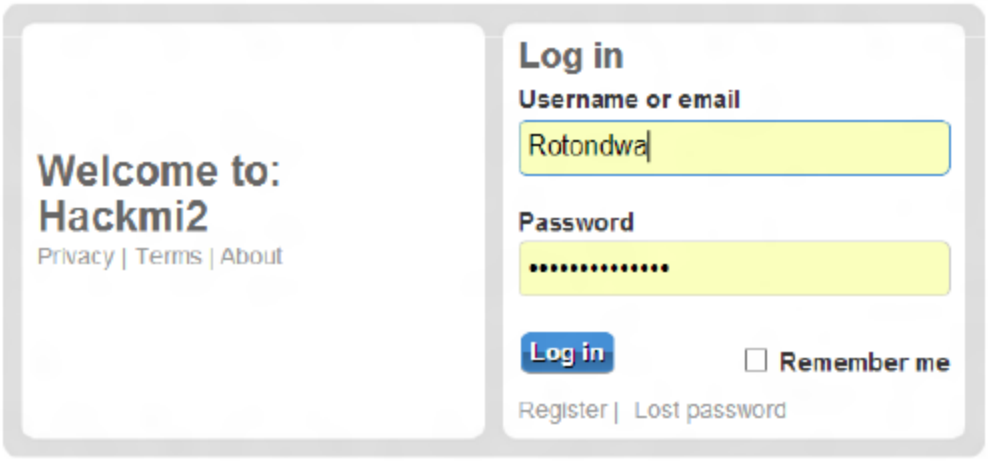


Figure 4.7: Original HACKMI2 login page

The current login page, with Pluto and ClickPoints added to the site is shown in Figure 4.8. ClickPoints is on the far left, with Pluto right next to it. The text-based login scheme is at the top right corner of the site.

Figure 4.8: Current HACKMI2 login page with Pluto and ClickPoints added to the site

Chapter 5 User Study

This chapter describes the user study that was carried out to compare the two graphical passwords schemes; namely, Pluto and ClickPoints. Additionally, these graphical passwords were compared to text password schemes. The user study would help us answer our two research questions and determine which of the three schemes was best suited for social networks.

5.1 Objectives

Our user study had four objectives. The first was to test the usability of Pluto and aimed to answer questions such as, is it simple, easy to understand and convenient to use? How difficult would it be to create a new password on a desktop with a mouse, or on a laptop with a touch pad? In addition, the usability of Pluto would be compared to that of ClickPoints.

The second goal of the user study was to learn the characteristics of user-chosen passwords in a practical, real-life setting (i.e in an environment where the passwords will be used frequently over a period of time). These characteristics included metrics such as the length of the passwords, the number of login attempts, and the number of times the password would be reset. Again, these metrics would be compared to that of ClickPoints and a text-based scheme (in terms of password initialization).

The third objective of the user study was to test the security of Pluto, and aimed to evaluate its robustness to guessing and capture attacks; namely, shoulder-surfing, social engineering and dictionary attacks. These results would be compared to those of ClickPoints.

The last goal of the study was to elicit the participant’s current strategies when using text passwords. This would be achieved by carrying out a pre-study questionnaire that would explore issues such as how often the users forgot their passwords; and whether they made use of coping strategies such as writing down the passwords in order to aid their memory. Additionally, a post-study questionnaire regarding the participants' experience and preferences would be carried out. The questionnaire would examine how the users interacted and managed their graphical passwords.

5.2 Outline of User Study

The four-stage user study was conducted in the second semester of 2013, over a month long period from late September to late October. In total 17 subjects participated, all of whom were students from the University of Cape Town. The participants were from various departments in the University such as Commerce, Engineering, Science and Humanities.

Our user group consisted of students who were all experienced computer users with a relatively high level of education. The study consisted of two questionnaires: the previously described pre-study and post-study questionnaires. These questionnaires can be found in Appendix A and B, respectfully. It also consisted of two sessions: a lab session and a field session, both of which will be outlined in this section.

5.2.1 Lab Session

The lab session was conducted in a controlled environment over a period of 3 weeks. It involved testing the system for robustness against shoulder-surfing, social engineering and dictionary attacks.

Participants were given consent forms before commencing evaluation. The forms explained their rights and outlined our responsibilities in this study. This was done to ensure that all ethics were upheld during the testing. The consent form we gave to our users can be found in Appendix C.

In order to test for robustness against shoulder-surfing and social engineering, the participants were grouped in pairs. One participant took the role of a legal user, and the other of the attacker. In the case of shoulder-surfing, the attacker would be given one chance to see the legal user log into both Pluto and ClickPoints. For social engineering, the legal user would be given one chance to describe his/her password to the attacker. In both cases, the attacker would then be asked to re-create the passwords, if possible.

A dictionary attack is a method used to break security systems, in which the attacker systematically tests all possible passwords beginning with words that have a higher possibility of being used, such as names and places. The attacker successively tries all the words in an exhaustive list called a dictionary. In the case of Pluto, the words in the dictionary consist of the most likely encoded passwords, i.e the sequences of coordinates that were most likely to be chosen by users.

In order to test for robustness against dictionary attacks, the participant's passwords were compared to the words of a dictionary. Figure 5.1 lists the ten passwords I chose as the words in my dictionary.

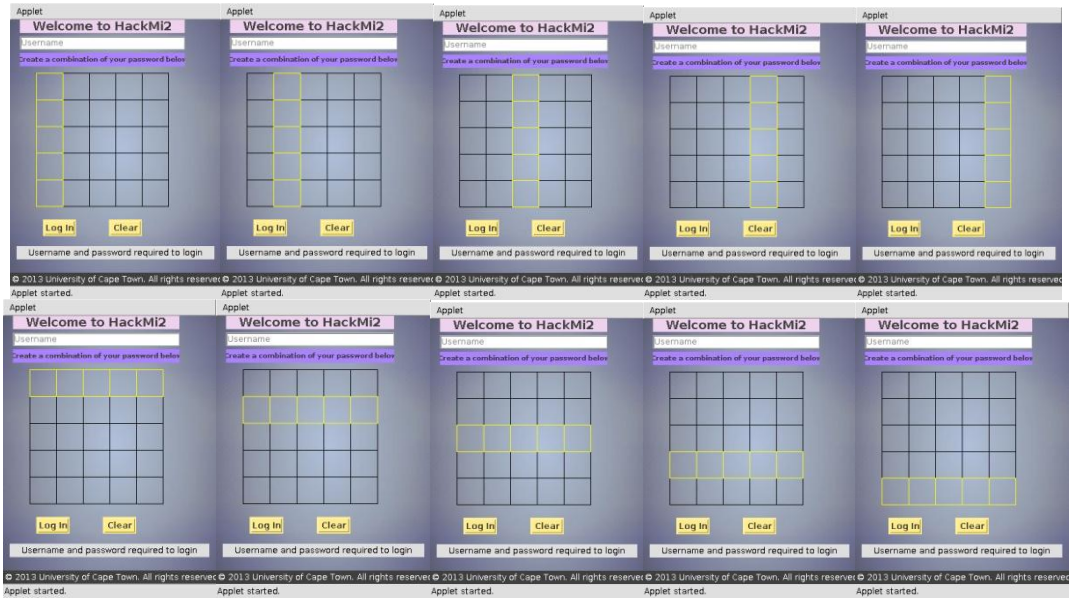


Figure 5.1: Words in the dictionary for Pluto

5.2.2 Field Session

The field session consisted of an online study which was conducted over a period of seven days. We had initially planned for the online study to span four weeks with participants logging onto HACKMI2 three times a week. We were unable to do because integrating our systems onto HACKMI2 took more time than expected. To compensate for the three weeks lost, we encouraged the participants to log onto HACKMI2 once a day for seven days. The advantage of this approach is that it was more similar to real-life settings as social network users check their social network sites very frequently.

The online study allowed users to authenticate under realistic settings and in their environments of use, thereby making the results more reliable. We installed the Java runtime environment plugin on the participant's laptops so that the Java applets would run on their devices. The online study would allow us to evaluate the usability of the system and the memorability of passwords. It required continued participation from the participants. To this end we will offered incentives to encourage high participation.

The next two sections describe the results that were found as we carried out the user study.

5.3 Results

The lab study was conducted with 12 participants, while 15 participants were tested for the field study. Some of the participants that were involved with the lab study were invited to participate in the field study as well. In total, 17 participants were involved in the user study. None of the participants used dedicated password storage software to store their password. In addition, none of them had more than 10 unique passwords that they made use of, partly because they were not a member of more than 10 password-protected websites.

The following describes the results that were found during our user study.

5.3.1 Shoulder-surfing Attack

The shoulder-surfing attack was conducted during the lab session. The study showed that all of Pluto passwords were stolen, while none of ClickPoints passwords were stolen with this attack. This is shown in Figure 5.2 below:

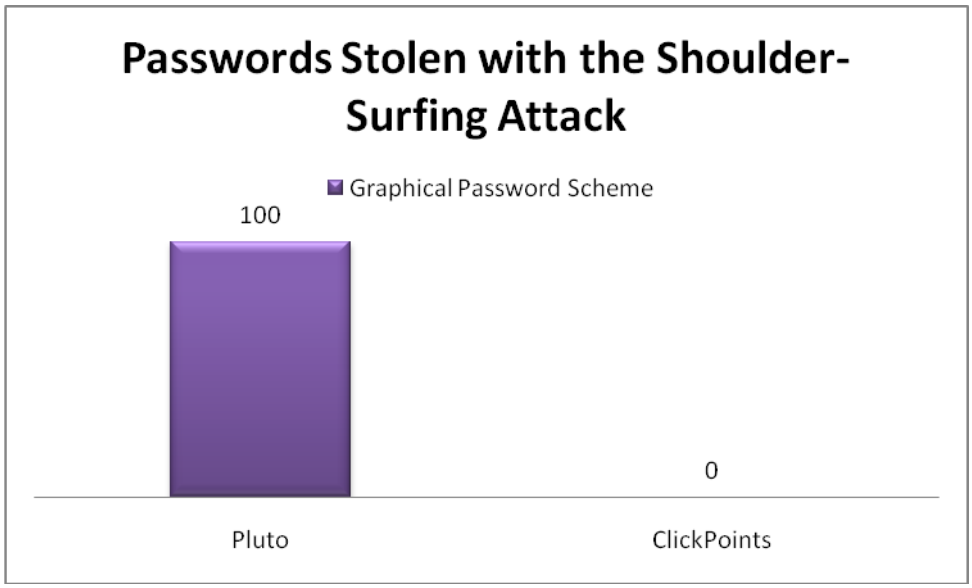


Figure 5.2: A comparison of the percentage of passwords stolen from Pluto and ClickPoints using the shoulder-surfing attack

5.3.2 Social Engineering Attack

The shoulder-surfing attack was conducted during the lab session. The study showed that 57% of Pluto passwords were stolen with this attack, while only 50% of ClickPoints passwords were captured. This is shown in Figure 5.3 below.

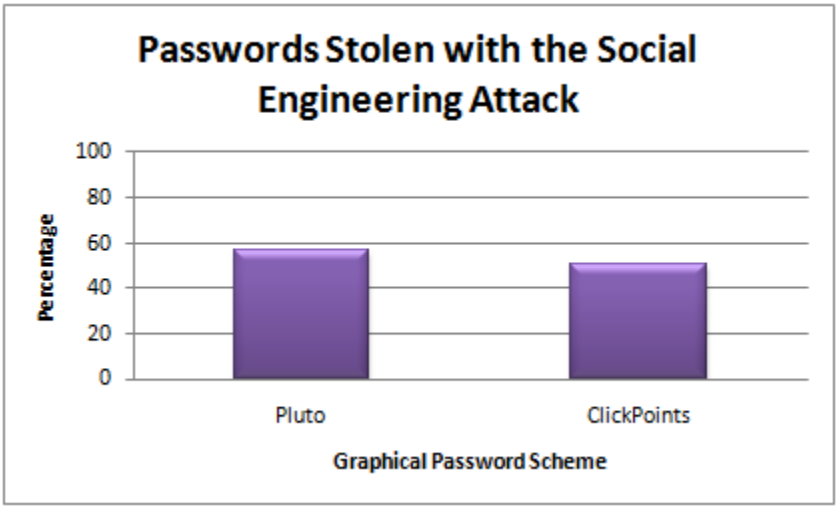


Figure 5.3: A comparison of the percentage of passwords stolen from Pluto and ClickPoints using the social engineering attack

5.3.3 Dictionary Attack

Using the pre-compiled dictionary described in §5.2.1 we tested Pluto for robustness against dictionary attacks. We found that only 12.5% of Pluto passwords were stolen,

while 33% of ClickPoints passwords were captured with this attack. This is shown in Figure 5.4 below.

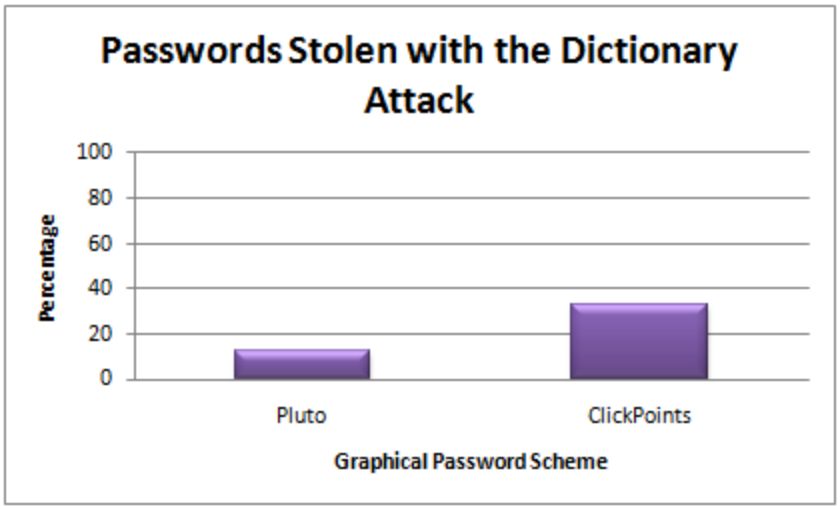


Figure 5.4: A comparison of the percentage of passwords stolen from Pluto and ClickPoints using the dictionary attack

5.3.4 Password Initialization

Regarding the simplicity in creating a password, half the participants preferred creating a Pluto password over a ClickPoints one.

When creating new text passwords, 62.5% of the participants indicated that they used personal information such as birthdays or the names of family members. In addition, 62.5% of the participants said that they used simple variants of their various text passwords across different applications; for example, cat1, cat12, cat123, and so on.

5.3.5 Usability

Regarding the usability of text passwords, 87.5% of the participants indicated that they sometimes forgot their passwords, while 75% indicated that they re-use passwords across different applications. To help with memorability, half of the participants indicated that they use their web browser to remember their passwords, although only 37.5% of the participants write down their passwords.

During the field testing, 76 login attempts were made and 60 of them were successful, giving a login success rate of 78.9%. The login success rate of ClickPoints was higher at 87.7% (Figure 5.6). Pluto had a standard deviation of 3.16, while the standard deviation of ClickPoints was 3.12.

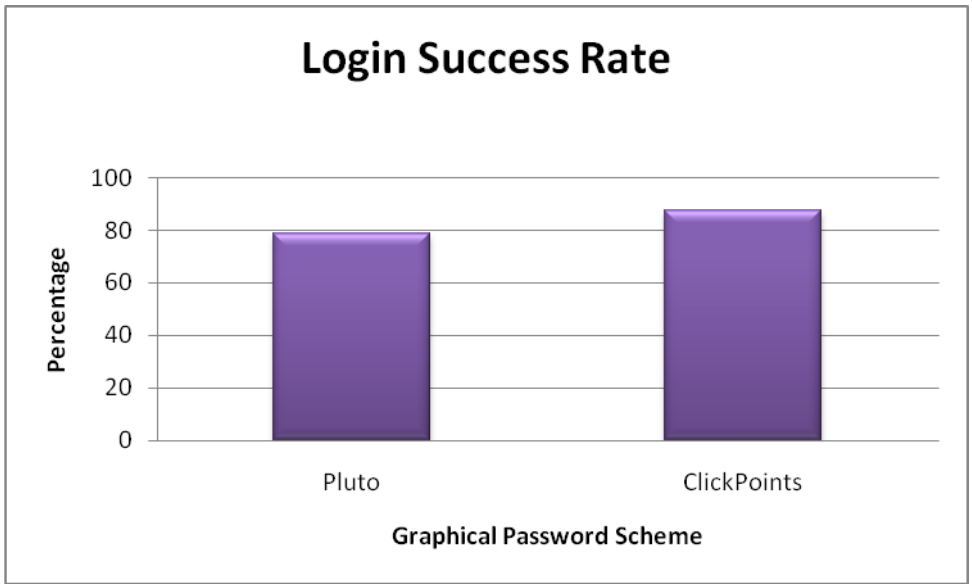


Figure 5.6: The login success rate for Pluto and ClickPoints

Only 2 users reset their passwords for the duration of the field test period, for both of the systems. Moreover, the lab-session participants thought that the passwords for both systems were equally easy to remember. However, after the field test only 22% thought that Pluto had the easiest password to remember.

Figure 5.7 shows the login failure rate for Pluto and ClickPoints at 21.1% and 12.3% respectfully. The failure rate of ClickPoints was less by 8.8%. This may be attributed to the fact that drawing through the cell-blocks would require more effort and concentration when using anything other than a stylus pen. Some of the users in our study complained that it was difficult to draw when using a laptop touch pad or a mouse. This is similar to what was found in the Pass-Go study.

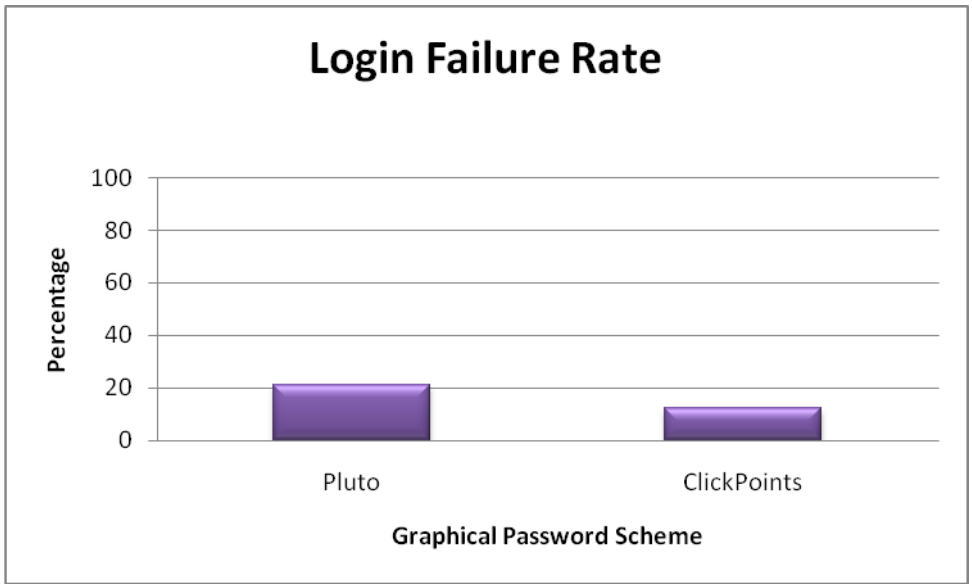


Figure 5.7: The login failure rate for Pluto and ClickPoints

Another similarity of our study results to those of Pass-Go was that users would have to spend a considerable amount of time getting accustomed to the system in order to avoid making unintentional errors (such as touching neighbouring cell-blocks).

5.3.5 Post-Study Questionnaire

Graphical passwords were received relatively well- 55% of the participants indicated that they would use a graphical password to authenticate a social network site. Only, 33% of the participants thought that graphical passwords were easier to manage and remember as compared to text passwords. In addition, 67% thought that graphical passwords took longer to create than text passwords. The same number of participants thought that graphical passwords took longer to login in as well.

Regarding security, 56% of the participants thought that graphical passwords were more secure than text passwords. However, the same number of people thought that the text passwords they create are also very secure.

Chapter 6 Discussion of Results

The evaluation of the performance of Pluto enables us to answer our two research questions. Firstly, 'Which category of graphical password schemes is best suited for social networks: schemes based on recall or cued-recall?' Secondly, 'Are graphical password schemes a viable alternative to text-based schemes as a means of providing authentication for secure social networks?' This chapter answer these questions and discuss the implication of the results we found in the previous chapter.

The results showed that security of ClickPoints was better than that of Pluto. All the Pluto passwords were stolen using the shoulder-surfing attack, despite using the line snaking defence technique. This result contrasts the original findings by Zakaria et al. were none of the DAS passwords were stolen when the line snaking defence was activated. None of the ClickPoints passwords were stolen using this attack, even without having any defence technique implemented.

In addition, ClickPoints passwords were found to be harder to describe and were thus more robust to social engineering attacks than Pluto passwords. Pluto passwords were found to be easier to describe, but only 7% more passwords were stolen with the social engineering attack. Pluto only performed better with respect to dictionary attacks where only one user's password was found in the pre-computed dictionary. Thus, ClickPoints was more robust to two security attacks, while Pluto was only robust to one.

Regarding usability, ClickPoints was found to have performed better than Pluto. It had a higher login success rate and a lower failure rate by a margin of 8.8%. Therefore, based on the results, we have found that the graphical password scheme best suited for social networks is the cued-recall password scheme, ClickPoints.

With respect to the use of graphical password schemes as a whole, our user study found that the graphical password schemes were generally well-received. However, some of the participants expressed concern over their usability and security.

More research is needed to find the balance between usability and security. Therefore, graphical passwords are not a viable alternative to text passwords, yet. However, as more work is done on this area of research, it may be a good alternative in the future.

6.1 Limitations

The project had several limitations. Firstly, because the Java applets required a Java environment plugin in order to run, we had to make sure that the plugin was installed on each of the participant's machine. This meant that we could not test the system on as many people as we had initially planned to because it was not purely online.

Secondly, our user study was conducted on 17 participants in total. Therefore the results presented may not be statistically significant. In addition, our user group consisted of students who were all experienced computer users with a relatively high level of education. Therefore, they may have performed slightly better than the general population in understanding and using our schemes.

Chapter 7 Conclusion and Future Work

This paper explores the viability of graphical passwords as an alternative to text passwords in the context of social networks. To do that it implements and compares two graphical passwords schemes based on the principles of recall and cued-recall. It then compares these two graphical passwords to text passwords. This is done in order to evaluate usability in terms of login and password recovery, and robustness to guessing and capture attacks such as shoulder-surfing, social engineering and dictionary attacks.

This report finds that the graphical password scheme best suited to social networks is one that is based on the principles of cued-recall. This is because it performed in both the usability and security evaluations of our study. Our results also indicate that future work is required to make graphical passwords a viable alternative to text passwords for social networks.

The contributions of this project include a new graphical scheme that aims to inherit the strong-points of DAS. Future work should be directed toward investigating long-term memory recall of graphical passwords, and evaluating variations of Pluto that include adding colour as a parameter and finding a better shoulder-surfing defence technique.

References

[1] Amzer®. Amzer® Privacy Protector Shield For Blackberry curve 8530,BlackBerry Curve 3G 9300. Available at: http://www.amzer.com/Amzer-Privacy-Protector-Shield-P84116.html (Accessed 22 April 2013)

[2] Biddle, R., Chiasson, S., and van Oorschot, P. C. 2012. Graphical Passwords: Learning from the First Twelve Years. ACM Computing Surveys (CSUR), vol. 44, 4, Article 19 (August 2012), 41 pages

[3] Dunphy, P. and Yan, J. 2007. Do Background Images Improve "Draw a Secret" Graphical Passwords? In Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS), ACM Press, New York, pp. 36-47

[4] Everitt, K., Bragin, T., Fogarty, J., and Kohno, T. 2009. A Comprehensive Study of Frequency, Interference, and Training of Multiple Graphical Passwords. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press, New York, pp. 89-898

[5] Gao, H., Guo, X., Chen, X., Wang, L., and Liu, X. 2008. Yagp: Yet Another Graphical Password Strategy. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), California, pp. 121-129

[6] Grossman, J., Livshits, B., Bace, R., and Neville-Neil, G. 2013. Browser Security: Appearances Can Be Deceiving. Communications of the ACM, vol. 56, 1, (January 2013), ACM Press, New York, pp. 60-67.

[7] Hayashi, E., and Christin, N. 2008. Use Your Illusion: Secure Authentication Usable Anywhere. In Proceedings of the 4th Symposium on Usable Privacy and Security, (Pittsburgh, PA, USA, July 23-25), ACM Press, New York, pp. 35-45

[8] Hong, J. 2013. Passwords Getting Painful, Computing Still Blissful. Communications of the ACM, vol. 53, 3, (March 2013), ACM Press, New York, pp. 10-11

[9] Jermyn, I., Mayer, A., Monrose, F., Reiter, M., and Rubin, A. 1999. The Design and Analysis of Graphical Passwords. In Proceedings of the 8th USENIX Security Symposium, vol. 8, ACM Press, New York, pp.1-1

[10] Malek, B., Orozco, M., and El Saddik, A. 2006. Novel Shoulder Surfing Resistant Haptic-based Graphical Password. In Proceedings of the EuroHaptics Conference (Paris, France, July 3-6)

[11] Moglen, E. 2013. Privacy and security the tangled web we have woven. Communications of the ACM, vol. 56, 2 (February 2013), ACM Press, New York, pp. 20-22.

[12] Sobrado, L. and Birget, J. C. 2002. Graphical Passwords.The Rutgers Scholar, vol. 4. Available at: http://rutgersscholar.rutgers.edu/volume04/sobrbirg/sobrbirg.htm (Accessed 22 April 2013)

[13] Suo, X., Zhu, Y., and Owen, G. 2005. Graphical Passwords: A Survey. In Proceedings of the Annual Computer Security Applications Conference (Tucson, Arizona, December 5-9), IEEE

[14] Tao, H. Pass-Go, A New Graphical Password Scheme. M.S. 2006. Thesis, School of Information Technology and Engineering, University of Ottawa.

[15] Tari, F., Ozok, A., and Holden, S. 2006. A Comparison of Perceived and Real Shoulder-Surfing Risks Between Alphanumeric and Graphical Passwords. In *Proceedings of the 2nd ACM Symposium on Usable Privacy and Security (SOUPS)*

[16] Zakaria, N. H., Griffiths, D., Brostoff, S. and Jeff, Y. 2011. Shoulder Surfing Defence for Recall-based Graphical Passwords. In Symposium On Usable Privacy and Security (SOUPS), Article No. 6, ACM Press, New York

[17] Zangooei, T., Mansoori, M., and Welch, I. 2012. A Hybrid Recognition and Recall Based Approach in Graphical Passwords. In Proceedings of the 24th Australian Computer-Human Interaction Conference, ACM Press, New York, pp. 665-673

Appendix

A. Lab Session Questionnaire

1. Have you ever forgotten a password?
   _____
2. Do you reuse your password for different applications e.g. UCT access, email, Facebook, etc.?
   _____
3. Do you use simple variants of your password for different applications e.g. sam1, sam12, sam123?
   _____
4. Do you use your web browser to remember your passwords for certain sites?
   _____
5. Do you ever write down your passwords, either on paper or electronically to help you remember them?
   _____
6. When creating a new password, do you make use of personal information? e.g. birthdays or names of family members
   _____
7. Do you make use of dedicated secure password storage software?
   _____
8. Do you have more than 10 unique passwords that you make use of?
   _____
9. Are you a member of 10 or more sites that require you to login in to them?
   _____
10. Which password scheme did you find the easiest to create?
    _____
11. In your opinion which password scheme had the easiest password to remember?
    _____

B. Post-Study Questionnaire

1. Would you use a graphical password to authenticate a social network site, for example your Facebook account?

   _____

2. In your opinion which password scheme had the easiest password to remember?

   _____

3. Do you think graphical passwords are easier to manage as compared to text based passwords?

   _____

4. What methods did you use to help you remember your graphical password?

   _____

5. If you had to choose a password scheme for your Facebook account, from the three that you have been using, which one would you pick?

   _____

6. Do you think graphical passwords take longer to create than a text password?

   _____

7. How long do you think it took to create your graphical password? 1-2min 3-4min 5-6min

   _____

8. How long do you think it takes to create a text password? 1-2min, 3-4min, 5-6min

   _____

9. Do you think it takes longer to login with a graphical password scheme as opposed to a text based password scheme?

   _____

10. How long do you think it took you to login with a graphical password scheme? 1-2min, 3-4min, 5-6min

    _____

11. How long do you think it took you to login with a text password scheme? 1-2min, 3-4min, 5-6min

    _____

Text-based passwords

1. How many times did you have to reset your text passwords?
   _____
2. Did you make any mistakes entering your text passwords and if so how many?
   _____
3. How secure do you think your text password is?

C. Informed Consent Form

Graphical Authentication for Secure Social Networks

Investigators:	Dorothy Mhlanga	mhldor003@myuct.ac.za	0790511129
	Lebogang Mametja	mmtleb002@myuct.ac.za	0769989404
Supervisor:	Dr. Anne Kayem	akayem@cs.uct.ac.za	0216502664

You are being invited to take part in a research study. Before you decide to participate in this study, it is important that you understand why the research is being done and what it will involve. Please take the time to read the following information carefully. Please ask the researchers if there is anything that is not clear of if you need more information.

Purpose of the Research
This study is designed to compare two graphical password schemes to each other and to text-based password schemes to evaluate usability, and robustness to guessing and capture attacks. All testing will be done on a prototype social networking platform called HackMi2 (http://hackmi2.cs.uct.ac.za/).

Description of Subject Involvement
The study will consist of a two-hour lab session and a four-week online study. If you agree to be part of the study, we will ask you to take part in tasks that include:

1. Answering questions about your current strategies when using text passwords, and about your experience and preferences on your use of graphical passwords (lab)
2. Participating in activities that will test the system for robustness against attacks (lab)
3. Logging onto the social networking site three-five times a week for four weeks (online)

Risks and Discomforts
Potential risks or discomforts include feeling frustrated from forgetting your password.

Confidentiality and Anonymity
We will make every effort to protect your privacy. We will not use your name or any other identifying details in any of the research reports. We plan to publish the results of this study, but will not include any information that would identify you. The results will be published in the form of a research paper and may be published in a professional journal or presented at professional meetings.

Compensation
You will be compensated with a small gift for your participation.

Withdrawal without Prejudice
Participation in this study is voluntary; refusal to participate will involve no penalty. You are free to withdraw consent and discontinue participation in this project at any time without prejudice or penalty. You are also free to refuse to answer any question we might ask you.

Ethics Approval

This study has been approved by the Faculty of Science Research Ethics Committee of the University of Cape Town. If you have questions about your rights as a research participant; or wish to obtain information, ask questions or discuss any concerns about this study with someone other than the researchers, please contact the Chair of the Faculty of Science Research Ethics Committee at 021 650 2786 or via email at richard.hill@uct.ac.za.

Consent

By signing this consent form, I confirm that I have read and understood the information and have had the opportunity to ask questions. I understand that my participation is voluntary and that I am free to withdraw at any time, without giving a reason and without cost. I voluntarily agree to take part in this study.

Name (please print)	**Signature**	**Date**