UNIVERSITY OF BEDFORDSHIRE

SECURITY AND USABILITY IN CLICK-BASED AUTHENTICATION SYSTEMS

By

Haider al-Khateeb

A thesis submitted to the University of Bedfordshire, in partial fulfilment of the requirements for the degree of Ph.D.

April 2011
SECURITY AND USABILITY IN CLICK-BASED AUTHENTICATION SYSTEMS

H. AL-KHATEEB

Abstract

Web applications widely use text passwords to confirm people's identity. However, investigations reveal text passwords have many problems and that there is a need for alternative solutions. For instance, users often forget their passwords, choose passwords which are easy-to-guess or vulnerable to cracking tools. Further, people write passwords down and/or share them with others. In addition, phishing attacks (using fraudulent websites to steal users’ credentials) continue to cost millions of dollars every year. During the second half of 2009, the Anti-Phishing Working Group (APWG) reported 126,697 unique phishing attacks worldwide. As such, one of this research’s objectives is to investigate public awareness of, and attitude towards, text password security and usability supported by surveying both up-to-date literature and users.

The aim of this research is to develop an alternative solution using visual passwords (VPs) to authenticate users on web applications and investigate its security and usability. A VP can be many things: a set of images used as a login portfolio, click-points inside images or a doodle (signature) drawn by a user. Since text passwords are favoured for their usability over tokens and biometrics, the research scope has been set to investigate alternative ideas which do not require resources additional to standard computer devices used to sustain human-computer interactions, such as mouse and keyboard. VPs have the potential to develop an alternative solution within this scope.

A comprehensive survey of the VP schemes found in the literature is conducted followed by a security and usability evaluation in which click-based systems are selected as the most suitable approach to achieve the aims and objectives of this research. Click-
based systems are VP authentication schemes in which the VP is a sequence of click-points performed on one or more images. Further, user perceptions were investigated to study their acceptance of various authentication mechanisms and techniques.

A novel click-based scheme is presented and developed throughout the research to introduce and investigate novel ideas to maintain security and usability simultaneously. It can resist multiple phishing and shoulder-surfing attacks without revealing the full user credentials. Further, the layout is designed to prevent MiTM attacks, also known as the second generation of phishing attacks. The VP is hashed to resist database attacks and the password space is extremely large compared to text passwords to resist brute force attacks. It has dual cues to maintain memorability and password recall is easy even when it is system-generated. Usability is considered through observation and laboratory studies to meet the requirements of HCI-Sec (Secure Human-Computer Interactions) aiming to present a secure scheme people can actually use.
Dedication

I dedicate this thesis to my parent Manahl Hasan and Malik al-Khateeb who have always supported me to continue my education and be the person I am today. They taught me that learning is a lifelong journey and that the largest assignments can be accomplished one step at a time.
Table of Contents

Abstract ....................................................................................................................... ii

Dedication .................................................................................................................... iv

Table of Contents ....................................................................................................... v

List of Tables ............................................................................................................... x

List of Figures ........................................................................................................... xiii

Acknowledgments ...................................................................................................... xix

Declaration ................................................................................................................... xx

Chapter 1 Introduction ............................................................................................... 1

1.1. Context .................................................................................................................. 1
1.2. Problem statement ............................................................................................... 2
1.3. Motivation ............................................................................................................ 2
1.4. Thesis statement (aims and objectives) ............................................................. 3
1.5. Scope .................................................................................................................... 4
1.6. Constraints ......................................................................................................... 4
1.7. Research methodology ....................................................................................... 4
1.8. Thesis overview ................................................................................................ 6
1.9. Research contributions ...................................................................................... 7
1.10. Related publications ......................................................................................... 9

Chapter 2 Background and Related Work ............................................................... 10

2.1. Authentication ...................................................................................................... 10
2.2. Text passwords ................................................................................................. 14
2.2.1. The text password problem .......................................................................... 14
2.2.2. Countermeasures and guidelines ................................................................. 16
2.3. Password strength ................................................................. 19
  2.3.1. Password space ............................................................ 19
  2.3.2. Entropy (bit strength) ................................................... 20

2.4. Type of password attack ..................................................... 20
  2.4.1. Shoulder-surfing .......................................................... 21
  2.4.2. Guessing ................................................................. 21
  2.4.3. Dictionary attacks ....................................................... 22
  2.4.4. Brute force .............................................................. 22
  2.4.5. Social engineering ..................................................... 23
  2.4.6. Phishing ................................................................. 23
  2.4.7. Malware ................................................................. 24

2.5. Introducing visual passwords (VPs) ..................................... 25
  2.5.1. Categorisation and listing of VP schemes ....................... 26
  2.5.2. Pure-recall .............................................................. 26
  2.5.3. Cued-recall ............................................................. 30
  2.5.4. Recognition .............................................................. 37
  2.5.5. Category-based evaluation ......................................... 48
  2.5.6. More insight into click-based systems ......................... 52

2.6. Other proposals ............................................................... 53
  2.6.1. Audio passwords ....................................................... 53
  2.6.2. Behavioural biometrics ............................................... 53

2.7. HCI-Sec ........................................................................... 54

2.8. Terminology ...................................................................... 56

Chapter 3 Survey of User Perceptions ....................................... 58

3.1. The survey ...................................................................... 59

3.2. Analysis of results ............................................................ 60
  3.2.1. General ................................................................. 60
  3.2.2. Level of computer usage ............................................ 61
  3.2.3. Text password related questions .................................. 62
  3.2.4. Alternative techniques for initial login ....................... 66
  3.2.5. Continuous monitoring .............................................. 67

3.3. Discussion ........................................................................ 70

3.4. Conclusion ....................................................................... 73
Chapter 4 HybridPass – A New VP Scheme ............................................. 75
4.1. Description of the system ........................................................................ 77
4.2. Usability discussion ................................................................................. 80
4.3. Security discussion .................................................................................. 81
4.4. Laboratory study ...................................................................................... 84
4.5. Experiment methodology ......................................................................... 84
4.6. Analysis of results .................................................................................... 87
  4.6.1. Training (prototype 1) ........................................................................... 88
  4.6.2. Registration phase ................................................................................ 89
  4.6.3. Retention phase ................................................................................... 90
  4.6.4. Accuracy ............................................................................................ 90
  4.6.5. Hotspots ........................................................................................... 91
4.7. Discussion .................................................................................................. 93
4.8. Conclusion ................................................................................................ 96

Chapter 5 True Discretization ........................................................................ 97
5.1. Introduction .............................................................................................. 97
5.2. Robust Discretization .............................................................................. 98
5.3. Centred Discretization ............................................................................ 99
5.4. Occurrence of false accepts .................................................................... 100
5.5. True Discretization .................................................................................. 102
5.6. Security analysis ...................................................................................... 103
5.7. Conclusion ................................................................................................ 104

Chapter 6 CCBS – A Method to Maintain Memorability, Accuracy of Password Submission and the Effective Password Space ........ 105
6.1. Introduction .............................................................................................. 106
6.2. The effective password space in click-based VPs .................................. 108
6.3. Accuracy of password submission ............................................................ 111
6.4. VP retention in click-based systems ......................................................... 112
6.5. Cued Click-Based System (CCBS) .......................................................... 113
6.6. Hypotheses ............................................................................................ 114
6.7. Experiment methodology ........................................................................ 115
  6.7.1. Experimental design .......................................................................... 115
  6.7.2. Materials .......................................................................................... 116
  6.7.3. Procedure ......................................................................................... 117
Chapter 7 Protection against Phishing Attacks .......................... 131

7.1. Insights into phishing ................................................................ 132
  7.1.1. The problem ......................................................................... 132
  7.1.2. Sample attacks ..................................................................... 133
  7.1.3. Key facts of recent phishing attacks ..................................... 135

7.2. Analysis of solutions .................................................................. 136
  7.2.1. SSL/TLS and third-party certification .................................. 136
  7.2.2. Anti-phishing tools ................................................................. 139
  7.2.3. Extending user authentication mechanisms ......................... 140

7.3. Why phishing still works ............................................................. 142

7.4. Towards a better anti-phishing approach ................................. 143

7.5. Developing the PassMark scheme .............................................. 145
  7.5.1. PassMark recognition ............................................................ 146
  7.5.2. PassMark protection ............................................................... 148

7.6. Developing the HybridPass/CCBS scheme ................................. 152

7.7. Evaluation and analysis ............................................................... 154

7.8. Man-in-The-Middle (MiTM) phishing attack ............................. 162
  7.8.1. Sample attacks ..................................................................... 162
  7.8.2. Analysis of solutions ............................................................. 164
  7.8.3. Proposed scheme to resist MiTM-phishing ............................ 165

7.9. Conclusion .................................................................................. 167

Chapter 8 Conclusion and Future Work ................................. 169
8.1. Conclusion ................................................................................................................. 169
8.2. Future work ............................................................................................................... 173

Appendix A - Survey of user perceptions (snapshots) ....................... 175
Appendix B - Experimental data (Chapter 4) ........................................... 184
Appendix C – Experimental data (Chapter 6) ............................................ 200
List of Tables

Table 3.1: Results of public acceptability of different initial login techniques, where 1 is very unacceptable and 5 is very acceptable. ................................................................. 66

Table 4.1: Description of prototype parameters (r is the tolerance distance from the actual click-point). .............................................................................................................. 87

Table 4.2: Calculated data of the total time to register a new account in Prototype 1 (in seconds) and the incorrect password submissions. Confirmation time is the time spent to re-enter the 4 click-points (First trial N = 40, Second trial N = 34) ........................................ 88

Table 4.3: Calculated data (in seconds) of the password entry time for those who successfully signed in to the system using Prototype 1 (N = 17). ........................................ 89

Table 4.4: Calculated data (in seconds) of the VP creation time for successful registrations, the number of incorrect password submissions, successful rates on the second registration attempt (SR) and the first attempt only (SR-F). (N = 40). ....................... 89

Table 4.5: Calculated data (in seconds) of the VP input time for successful signed in attempts. The number of incorrect password submissions and success rates (SR). (Prototype 2: N = 37, Prototype 3: N = 35, prototype 4: N = 39, Prototype 5: N = 37) ... 90

Table 4.6: Means (in seconds), standard deviations (SD) and success rates (after one attempt) for Passpoints (Wiedenbeck et al., 2005c), CCP (Chiasson et al., 2007) and HybridPass (Prototype 4). Registration (Reg.) in this table stands for the time need to successfully create and confirm a new VP. ........................................................................ 95

Table 5.1: The number of false accepts click-points in the Centred Discretization tolerance grid square. .............................................................................................................. 101

Table 5.2: The number of possible clicks (N) while using Centred Discretization (CD) and True Discretization (TD) in PassPoints. ........................................................................ 103

Table 6.1: Examples of the textual cues used in CCBS with their relevant click-cells. . 114
Table 6.2: Calculated data of the total number of attempts and time to register and confirm (in seconds) a new account. Confirmation time is the time spent to re-enter the 5 click-points. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$). ................................................................. 120

Table 6.3: Questions about the registration phase. A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$). ................................................................................................. 120

Table 6.4: Means and Standard Deviations (SD) of the total practice time (in seconds) and the number of incorrect password submissions in the learning phase. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$). ........................................................................................................ 121

Table 6.5: Means and Standard Deviations (SD) of questions about the learning phase. A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$). ........... 122

Table 6.6: Means and Standard Deviations (SD) of the time for correct submissions (in seconds) and the number of incorrect password submissions in the retention phase. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$). ........................................................................................................ 123

Table 6.7: Means and Standard Deviations (SD) of questions about the retention phase. A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$). ............ 125

Table 7.1: Authentication schemes used in different banks. ........................................ 155

Table 7.2: The number of basic phishing attacks required to recover the full user credentials in different authentication schemes. ................................................................. 157

Table 7.3: The number of phishing attacks required to recover the full user credentials in different authentication schemes, assuming the user ID is known and the error-feedback-trick is implemented. ........................................................................................................ 158

Table 7.4: The authentication schemes resistant to different password attacks. ........ 160

Table B.1: Prototype 1 registration data. (*) is used to highlight data for successful registrations only. ................................................................. 184

Table B.2: Prototype 1 sign-in times and calculation details. Data are from participants who successfully sign-in. (*) is used to highlight data for successful registrations only. 186
B.3: Prototype 2 and Prototype 3 registration times, success rates and other calculated information. .................................................................................................................................................. 187

Table B.4: Prototype 4 and Prototype 5 registration times, success rates calculated details. .................................................................................................................................................................................................................. 189

Table B.5: Sign-in times, success rates and calculation details. ...................................... 191

Table B.6: The images used in HybridPass experiments to create user portfolios and their reference codes. .................................................................................................................................................................................. 193

Table B.7: The VPs (user clicks) collected throughout the experiments from the 5 prototypes. .................................................................................................................................................................................................................. 195

Table C.1: The images used in the CCBS experiment to create users portfolios accompanied by their reference codes. .................................................................................................................................................................................. 200

Table C.2: The VPs (user clicks) from the CCBS experiment. Each click is represented by two numbers (image coordinates) and an image reference code. Each element is separated by a semicolon. .................................................................................................................................................................................. 201
List of Figures

Figure 2.1: Input of a graphical password in DAS (Jermyn et al., 1999) ........................................ 27
Figure 2.2: Pass-Go sign-in form .................................................................................................. 29
Figure 2.3: Passlogix, implementation of Blonder’s scheme. Taken from (Suo et al., 2005) ................................................................................................................................. 31
Figure 2.4: Screenshot of the Inkblot Authentication scheme experiment (Stubblefield and Simon, 2004) .................................................................................................................. 32
Figure 2.5: Randomly generated inkblots used in (Stubblefield and Simon, 2004) .............. 32
Figure 2.6: Snapshot of proof of 3D Password concept. A virtual art gallery hosts 36 pictures and 6 computers. Users can navigate and interact with virtual objects (Alsulaiman and ElSaddik, 2006) ........................................................................................................ 35
Figure 2.7: In CCP, each click point leads to a different sequence of images (Chiasson et al., 2007) .......................................................................................................................... 36
Figure 2.8: Passfaces (Real User Corporation, 2001) ................................................................. 38
Figure 2.9: Déjà Vu, portfolio selection window (Dhamija and Perrig, 2000) ......................... 39
Figure 2.10: VIP example (De Angeli et al., 2005) ................................................................. 41
Figure 2.11: VIP example (De Angeli et al., 2005) ................................................................. 41
Figure 2.12: Man’s shoulder-surfing resistant scheme (Man et al., 2003). The case where all user icons are on the right hand side of the screen ......................................................... 42
Figure 2.13: Hong’s spyware resistant scheme, sign-in page (Hong et al., 2004) .......... 43
Figure 2.14: Story, sample of image selection window (Davis et al., 2004) ....................... 43
Figure 2.15: Weinshall’s Cognitive Authentication Scheme Query Panel (Weinshall, 2006) ............................................................................................................................................. 45
Figure 2.16: CHC, Illustration of the invisible convex hull with 3 pass-icons (Wiedenbeck et al., 2006) ....................................................................................................................... 46
Figure 2.17: DynaHand, example of one of the authentication stages (Renaud and Olsen, 2007)

Figure 2.18: User Your Illusion - Image portfolio assignment. The original images are shown along with their distorted versions (Hayashi et al., 2008)

Figure 3.1: Demographic information (age)

Figure 3.2: Demographic information (place of residence)

Figure 3.3: Number of accounts

Figure 3.4: How often passwords of critical accounts are changed

Figure 3.5: How often participants avoid simple words available in dictionaries

Figure 3.6: How often a combination of random letters (a,b, c ...), numbers (1, 2, 3...) and symbols (?, !, @...) are used together to create a password

Figure 3.7: How often a password is written down after creating a new account

Figure 3.8: How often the participants check their new password to be at least 6 characters long when creating a new account

Figure 3.9: Comparison between participants’ passwords at work and the ones they use for internet banking

Figure 3.10: Comparison of public acceptability of different initial login techniques, where 1 is very unacceptable and 5 is very acceptable

Figure 3.11: Comparison between the public acceptability to mouse and keystroke dynamics to be used for continuous monitoring

Figure 3.12: Acceptable frequency of false rejection

Figure 3.13: Acceptable frequency of automatically logging a user account out after period of inactivity

Figure 3.14: Comparison between the four age groups in the survey regarding the number of accounts they deal with
Figure 3.15: Comparison between students’ and non-students’ responses regarding the number of accounts they deal with. ................................................................. 72

Figure 3.16: Comparison between responses from Europe and Asia regarding the number of accounts they deal with. ................................................................. 72

Figure 4.1: Illustration of the HybridPass sign-in form. ........................................ 78

Figure 4.2: Schematic view of the sign-in process .............................................. 79

Figure 4.3: Sign-in form after entering a valid ID and any text password ............. 79

Figure 4.4: Phase one in the password retention process, to enter the user ID and Text Password. ................................................................. 86

Figure 4.5: Phase two in the password retention process, to enter the VP .......... 86

Figure 4.6: Analysis of the rejected click-points during the retention phase from all prototypes. ................................................................. 91

Figure 4.7: Scatter chart illustrating 83 clicks on the most employed cartoon image in Prototypes 2 and 4 ................................................................. 92

Figure 4.8: Scatter chart illustrating 87 clicks on the most employed photograph image in Prototypes 3 and 5 ................................................................. 92

Figure 4.9: Line chart illustrating the probability distribution of 83 user clicks on the most employed cartoon image in Prototypes 2 and 4. The ‘Equal Distribution’ line represents the case if all clicks are distributed equally (all squares have the same probability). 92

Figure 4.10: Line chart illustrating the probability distribution of 87 user clicks on the most employed Photograph image in Prototypes 3 and 5. The ‘Equal Distribution’ line represents the case if all clicks are distributed equally (all squares have the same probability) ................................................................. 93

Figure 5.1: False rejects and false accepts in Robust Discretization (Chiasson et al., 2008c). ................................................................. 99

Figure 5.2: Centred Discretization. The continuous line L is divided into segments of length 2r+1, where the value of 1 represents the original click-point.............. 100
Figure 5.3: Occurrence of false accepts. Each square represents a tolerance grid in centred discretization, and the circle inside contains the true tolerance space in a 2-D environment. 

Figure 6.1: The original image used in PassClicks. 

Figure 6.2: An overlay of 157,090 points. 

Figure 6.3: The original image. It was taken from (Wiedenbeck et al., 2005b). 

Figure 6.4: Hotspots and clustered areas are highlighted. (Thorpe and Oorschot, 2007). 

Figure 6.5: Screenshot of the PCCP Create Password interface with the randomly highlighted area. Taken from (Chiasson et al., 2008b). 

Figure 6.6: Number of participants who failed to submit a valid password on the first attempt. R1 represent instant retention of the VP, R2 represent the retention phase after one week and R3 after 4 weeks from R2. The figure has been taken from (Wiedenbeck et al., 2005c). 

Figure 6.7: In addition to the visual cues, in CCBS each click-cell has a unique textual cue appearing on ‘mouse hovering’. 

Figure 6.8: The Sign-in page of the prototype employed in the CCBS experiment. The mouse is highlighting a particular click-point hence the relevant textual cue is displayed: ‘Tom and Jerry is an animated show watched by children, teenagers and adults’. 

Figure 6.9: An illustration of a single click-point. The click-point is visually highlighted inside the image and the textual cue is displayed next to it. 

Figure 6.10: The number of incorrect submissions in the learning phase. 

Figure 6.11: The number of participants in each retention trial who failed to submit their correct password on the first attempt. 

Figure 6.12: The number of participants in each retention trial who failed to submit their correct password after five attempts.
Figure 6.13: Line chart illustrating the probability distribution of 100 clicks. As such, \( P(p_{01x}) \) is the probability of the image: \( p_{01x} \). The ‘Equal Distribution’ line represents the case if all clicks are distributed equally (all squares have the same probability). .......... 126

Figure 7.1: Sample of a phishing message targeting PayPal users. ......................... 133

Figure 7.2: Sample of a phishing message targeting HSBC users. ......................... 134

Figure 7.3: Snapshot of HSBC (UK) Internet Banking Personal logon page.......... 135

Figure 7.4: Snapshot of a spoof page of the HSBC (UK) logon page. ................. 135

Figure 7.5: In Google Chrome, clicking the ‘lock’ icon indicator of SSL/TLS opens a security information window. Details provided include: website identity verification, CA name and encryption type. ................................................................. 137

Figure 7.6: In HybridPass, the design of the scheme forces user authentication to fail before recovering the VP during the first attack................................................................. 145

Figure 7.7 : In PassMark, users are asked to enter their text passwords only after authenticating the server by recognising a shared secret/image. However, most users are found to submit their passwords anyway (Stone, 2007) as illustrated. ............... 147

Figure 7.8: Bank of America SiteKey challenge questions........................................... 150

Figure 7.9: An example of a shared secret, an image and its accompanying title. .... 150

Figure 7.10: User authentication when the client’s computer is recognized by the server in the anti-phishing implementation of HybridPass............................................. 153

Figure 7.11: User authentication when the client’s computer is not recognized by the server in the anti-phishing implementation of HybridPass............................................. 153

Figure 7.12: In a MiTM-phishing, the attacker creates two separate connections pretending to be the bank’s website in the user’s browser and then interacts with the bank’s website using the user’s data. ................................................................. 162

Figure 7.13: A demonstration of a MiTM-phishing attack against BoA showing the first screen of the sign-in procedure secured with SSL......................................................... 163
Figure 7.14: A demonstration of a MiTM-phishing attack against BoA. After feeding the forms with correct user details. ............................................................... 164

Figure 7.15: Altering HTML code to run Phishing attacks. ................................. 166

Figure A.1: Snapshot 1/9 showing the welcome page of the online application survey presented in Chapter 3 ......................................................................................... 175

Figure A.2: Snapshot 2/9 of the online application survey presented in Chapter 3....... 176

Figure A.3: Snapshot 3/9 of the online application survey presented in Chapter 3....... 177

Figure A.4: Snapshot 4/9 of the online application survey presented in Chapter 3....... 178

Figure A.5: Snapshot 5/9 of the online application survey presented in Chapter 3....... 179

Figure A.6: Snapshot 6/9 of the online application survey presented in Chapter 3....... 180

Figure A.7: Snapshot 7/9 of the online application survey presented in Chapter 3....... 181

Figure A.8: Snapshot 8/9 of the online application survey presented in Chapter 3....... 182

Figure A.9: Snapshot 9/9 of the online application survey presented in Chapter 3....... 183
Acknowledgments

First and foremost, I owe my deepest gratitude to my supervisor and Director of Studies, Prof. Carsten Maple, for his insightful criticisms, advice and encouragement. This research was possible due to his ideal guidance and support.

I am also very grateful to my second supervisor, Dr. Marc Conrad, for his precious comments and support. My appreciation also goes to my examiners, Prof Tony Solomonides (University of the West of England) and Dr Wei Huang for provided valuable feedback on the content and structure of the thesis. My colleague and friend, Dr. Peter Norrington, for all the valuable discussions and for proof reading my thesis.

Furthermore, it is a pleasure to thank my colleagues in the Institute for Research in Applicable Computing (IRAC), at the University of Bedfordshire and all my friends for their company, moral support, comments and participation in some of the experiments. In particular, I would like to mention Hossam Ali, Mudhar al-Najjar, Divya Rao, Zahraa al-Attiya, Rua al-Sheikh, Shaima Abukhater, Ghazal Zayad and Dr. Hanin Omar.
Declaration

I declare that this thesis is my own unaided work. It is being submitted for the degree of Ph.D. at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other University.

Name of candidate: Haider Malik al-Khateeb

Date: 18/02/2011
Chapter 1  Introduction

Technology improvement is challenging the security of current user authentication methods at its best, text passwords in particular. When you select a new password similar to: ‘h5?2&sj8’ which is a good randomly mixed string of upper and lower alphanumeric characters and symbols, you expect it to resist cracking tools for years to come, but modern personal computers employing multiple video cards could reach the capacity of searching 1 billion passwords/sec (net-security.org, 2008, whatsmypass.com, 2008) which implies that it could crack its hash in approximately 71 days. In addition, connecting multiple computers to divide the task among them decreases the attack time effectively. For instance, the ongoing project RC5-72 (distributed.net, 2002) employs large number of distributed computers all over the world to break encryption algorithms, reaching a sustained rate of searching 743 billion keys/sec on 3/5/2010. Power like this can recover the same hash in only two to two and a half hours.

Other challenges facing text passwords include guessing and dictionary attacks. These issues remain a problem because maintaining text password security collides with the human lack of ability to remember good passwords. Security and usability are hard to balance since what is easy to remember is not safe enough, and a well selected password is hard to remember.

This work recognises the major impact of the human factor in user authentication security. It researches the employment of a different type of knowledge-based methods called visual passwords (VPs). Hence, a survey of different VPs is conducted followed by an evaluation of the best approach to develop novel ideas aiming to solve authentication security problems while maintaining usability at the application level. Various prototypes were developed to support the study with empirical evidence.

1.1. Context

The human factor is a fundamental reason behind selecting weak text passwords, and since secure user behaviour is hard to maintain, new schemes must be carefully designed to aid and influence users toward better password selection with the least
possible effort on the users' part. HCI-Sec is the term emerged to reflect this area of intersection between Human-Computer Interaction (HCI) and computer security. HCI aims to make systems more usable, while computer security in this scenario is concerned with keeping the system and all related data safe against unauthorised access. Hence, HCI-Sec is the research area of investigating and developing new design technologies to produce a system that can maintain both security and usability (Garfinkel, 2005).

While early literature shows that security and usability issues were recognised, they were addressed and discussed separately (Reid, 1990) until this research area received more attention during the 1990’s from a number of researchers such as (Whitten and Tygar, 1999) and (Zurko and Simon, 1996) who introduced ‘user-centred security’.

1.2. Problem statement

Text passwords are vulnerable to many security attacks due to the insecure practices of end users who select weak passwords as a mechanism to maintain their long term memory. A secure text password should be a long random mixture of characters. These are hard to remember and in addition, they can be easily shared, written on paper or reused on multiple systems. There are no methods for system administrators to guarantee that passwords are not used again for other accounts on other systems. Reusing passwords implies that recovering a particular account’s password exposes other systems protected by the same password as well.

1.3. Motivation

Despite all security challenges to text passwords, they remain ubiquitous and hard to replace due to their usability, hence a potential replacement must provide end users with a comparable level of flexibility and usability in addition to security. What is encouraging about researching VPs is the fact that their implementations can easily adapt to new ideas. For instance, Déjà Vu (Dhamija and Perrig, 2000) and Passfaces (Real User Corporation, 2001) are both recognition-based systems, but one was designed with abstract images to make social engineering hard to succeed and the other one used human faces to maintain a higher level of memorability. Similarly, Passpoints (Wiedenbeck et al., 2005c) was developed based on Blonder’s patent (Blonder, 1996) to provide a click-based solution with a very large password space in comparison to other implementations such as PassLogix (Boroditsky, 2002).
VP schemes have the advantage of triggering users' memory in different ways to recall their passwords, which implies that they provide aid to end users' long term memory, the real problem behind having weak passwords. Much research over the last decades shows that humans can remember visual elements and meaningful pictures better than text. (Madigan, 1983) (Nelson et al., 1976) (Bower et al., 1975) (Paivio et al., 1968) (Shepard, 1967) (Calkins, 1898).

It is very critical to take the human factor into consideration and VPs support that. In addition, the environment of VPs gives researchers the opportunity to select and develop schemes to fight different security attacks such as guessing and phishing.

1.4. Thesis statement (aims and objectives)

Aims
The first aim of the study is to develop a new VP authentication mechanism for web applications. The authentication scheme must be user-friendly, cost-effective and provide novel techniques to resist some of the security threats text passwords are vulnerable to, such as phishing, brute force and dictionary attacks.

Nevertheless, and because text passwords remain the de facto approach to user authentication and are still used in the vast majority of systems (DTI, 2006), the study aims secondly to investigate users' awareness of, and attitudes toward, text password security and usability and provide an up-to-date information

Objectives

1. Investigate users' attitudes toward text passwords, their perceptions of other knowledge/secret-based mechanisms and the concept of continuous monitoring.

2. Identify the best VP approach based on a security and usability evaluation to develop a substantial authentication solution suitable for web application.

3. Build the new authentication scheme to be more resistant to brute force and dictionary attacks.

4. Develop an effective anti-phishing technique.

5. Evaluate the scheme against different attacks analysing its strength and weakness to achieve better security and usability.
1.5. Scope

The scope of this research covers knowledge-based or secret-based authentication schemes, especially VPs. Any technology requiring additional hardware resources to the standard computer input/output devices (monitor, mouse and keyboard) is beyond the scope.

A web server will be the host of the developed system prototypes to imitate the environment of a real internet application. Various effective web development technologies will be used to enhance the HCI. For instance, PHP and SQL are used for the server side scripting and JavaScript is employed to perform the required interactivity on the client’s side.

1.6. Constraints

The results of the experiments developed throughout the thesis are based on laboratory studies. Although countermeasures were taken to maintain an environment of real world characteristics, it is possible that long term field studies will challenge these results. Due to time restrictions while performing experiments to analyse the usability of the system, it was not possible to maintain an effective long term field study which could last for months and require hundreds of participants.

The scheme’s behaviour towards different type of security attacks is hypothetically discussed, or tested and evaluated in the laboratory, publishing its design or using it as part of a live system could disclose new vulnerabilities by attackers or by other researchers.

This research focuses on developing the system at the application level, hence, low level protection (e.g. using network encryption to resist sniffers) is not discussed in detail, but rather required and expected to exist as part of the infrastructure for any authentication system. In addition, the prototypes are designed and coded to serve the purpose of adding improvements and testing new features for click-based VP systems and may not be used directly in live systems without further enhancements.

1.7. Research methodology

To achieve the objectives of this research, the following are constantly used:
a. Prototyping

A vertical prototype approach is adopted to eventually test and evaluate the implementation of the scheme. This helps to implement changes related to usability issues and develop the HCI. The development environment (web hosted) has been registered online where PHP database-driven applications can be hosted.

PHP, MySQL, JavaScript, AJAX and other scripting languages are used to develop the new authentication tool.

b. Experiments

The experiments serve as the point of departure to start the discussion with end users to discover practical problems, gather feedback and subsequently analyse how the application is being used. Participants access the application prototype online, fulfil the requirements of the experiment and answer a short survey.

c. Surveys and interviews

An open source survey application (limesurvey.org, 2010) has been chosen and developed to meet the research requirements. To collect and control data integrity from a wide range of participants, the online survey system is critical to moderate the data collection process.

Question types could be multiple-choice questions, comment box or Likert scale. Responses are date stamped and IP addresses are logged because participants are required to complete the survey online and this helps to put restrictions on how many times a survey can be accepted from a specific party. Cookies are also used for access control so that participants can save partially-finished surveys and finish them at a later time.

Surveys and questionnaires are useful to gather feedback at the end of every experiment as well as achieving some of the study's objectives, for instance, to understand users' perceptions of existing online authentication mechanisms, the concept of continuous monitoring or even their attitudes towards text passwords security.
**d. Exhaustive reviews and literature investigations**

In addition to performing surveys and experiments, exhaustive literature investigations take place continuously to make use of other research work in the field.

Existing research work could save time and effort. For example, it could help to understand a particular problem from different perspectives, analyse the strengths and weaknesses of proposed solutions and support relevant arguments with the necessary evidence.

**1.8. Thesis overview**

The remainder of the thesis is organised into seven further chapters. *Chapter 2* introduces user authentication then investigates the text password problem, possible attacks and available countermeasures. It partially addresses *Objective 1* through discussing user attitudes towards text passwords, while the remaining sections address *Objective 2* with an up-to-date survey of VP schemes available in the literature, where security and usability elements are evaluated to identify the most convenient approach. HCI-Sec is covered in detail. The chapter then concludes with a list of the terminology used throughout this thesis.

*Chapter 3* is dedicated to addressing *Objective 1*. It presents analyses of a survey designed to cover the public’s awareness of, and attitudes toward, password security and usability, different forms of initial user authentication mechanisms and the concept of continuous monitoring.

*Objective 3* required designing and developing a new VP scheme. The contribution to achieve this objective is presented in *Chapter 4* with a new click-based system called HybridPass. Both security and usability issues are theoretically discussed, followed by an analysis of a laboratory study in which a number of issues were covered such as training time, the types of image used, hotspots and dictionary attacks, tolerance, success rates and others.

Further, technical solutions are developed to support the application of the system. For instance, discretization is used to accept clicks within an area of tolerance around the original click-point in click-based systems. Proposals from the literature such as (Birget et al., 2003) and (Chiasson et al., 2008c) allowed false accepts and/or false rejects. *Chapter 5* discusses these methods in detail and provides an extension method to achieve zero false accepts/rejects to increase the password space.
Another scheme based on the original HybridPass prototype is then introduced in Chapter 6, called Cued Click-Based System (CCBS). It is developed to achieve a number of objectives. For example, it resists dictionary attacks through giving each element (click) used to create a visual password an equal chance of being selected by the system while maintaining memorability. As such, the scheme was designed to support password retention in the long term. In addition, it provides additional cues during password entry to reduce the number of incorrect password submissions.

Phishing attacks against authentication systems are investigated in Chapter 7 where Objective 4 is addressed. The chapter starts with insights to the problem and includes discussions supported by sample attacks followed by a comprehensive analysis of the existing proposals to resist phishing. The analysis helped to develop an anti-phishing approach which was then employed to develop two existing schemes (one of them is the HybridPass/CCBS scheme) to provide better protection against this type of attacks. Nevertheless, a new generation of phishing attacks referred to as Man-in-The-Middle Phishing is also covered with a potential proposal to stop its threat.

Objective 5 is addressed in Chapters 4 to 7 with the discussion of different potential attacks. Chapter 8 concludes this research with an evaluation of the usability and security of the scheme in its final prototype (i.e. CCBS with Phishing resistant). Finally, future work is addressed in the same chapter.

1.9. Research contributions

The research contributions to the field of HCI-Sec and authentication systems are:

1. A comprehensive survey of visual password schemes with a security and usability discussion supported by up-to-date references from the literature. The schemes were categorized and evaluated based on the type of memory to select the best approach for this work (Chapter 2).

2. Investigated public attitudes and perceptions towards different forms of visual passwords (Chapter 3).

3. Investigated public attitudes and perceptions towards implementing continuous monitoring in online authentication systems to maintain security for a logged in session based on user behavioural biometrics (Chapter 3).
4. Provide evidence that the number of frequently used password-protected systems is continually increasing (Chapter 3).

5. A flexible scheme (HybridPass) was designed with advantages over older click-based system such as Passpoints and CCP in terms of flexibility and providing better resistance to security threats such as shoulder-surfing and phishing. In addition, it was designed to eliminate denial of service attach since it replaces the traditional technique of controlling the number of incorrect login attempt to resist guessing with an alternative novel technique (Chapter 4).

6. Identified criteria and addressed challenges applicable to click-based systems such as image types, tolerances, accuracy and dictionary attacks (the hotspots problem) (Chapter 4).

7. A scheme (CCBS) with a novel design to resist dictionary attacks while maintaining the password space, accuracy of password submissions and user memorability for the short and long term (Chapter 6).

8. True Discretization is proposed in Chapter 5, a method to accept clicks within an area of tolerance with zero false accepts/rejects in click-based systems. Older methods from the literature are proved to allow false accepts and/or false rejects if applied with no further modification.

9. An anti-phishing approach is developed and presented in Chapter 7 following an extensive security and usability discussions of various proposals from the literature.

10. Existing schemes such as PassMark (currently used for internet banking) and HybridPass/CCBS were developed. Their superior resistant to phishing attacks was then analysed and evaluated against schemes currently used in various banks including HSBC, Lloyds TSB, Bank of America and others.

11. A new scheme to resist the new generation of phishing attacks (Man-in-The-Middle phishing) is proposed (Chapter 7).
1.10. **Related publications**


Chapter 2  Background and Related Work

This chapter introduces user authentication then presents the text password problem, possible attacks and available countermeasures, followed by an up-to-date survey of VP schemes available in the literature where security and usability elements are discussed and evaluated. The results suggest click-based systems to be promising; hence they are covered in further detail to support the following chapters where new techniques and ideas are tested on a number of click-based system prototypes developed for this purpose. HCI-Sec is presented and discussed. The chapter then concludes with a list of the terminology used throughout this thesis.

2.1. Authentication

In computer security, authentication is defined and presented as the process of confirming the identity of the originator. This process starts with ‘identification’ wherein the originator posts claim to his identity. The process then continues to verify and confirm the claim (e.g. a person proves who he claims to be); if the originator is proved as authentic, he is authorised with access rights to resources.

Identity confirmation is achieved based on one authentication factor or more. These factors are generally classified into three main classes:

a. **Something the user knows or recognises (knowledge factors):**

This works by sharing a secret which users need to recall, this –for instance– can be a password, passphrase or a personal identification number (PIN) between an access control system and a user. Every time users attempt to gain access, the system asks them to recall the shared secret correctly.

Another way is to recognize a secret provided to users during their first enrolment. The secret can be a pre-selected image (as in recognition-based VPs). Then to confirm identity a user must recognise and select the same image or an image associated to it, from a set of images generated by the system.
Some systems combine both: recall and recognition in what is known as cued-recall. These systems provide the users with a cue to recall the secret. An example to this is the Question-and-Answer approach. Click-based VP solutions are another example.

b. **Something the user has (ownership factors):**

Users provide a unique token they own or associated to them to authenticate. This varies from one system to another, some of these identity confirmation methods are: ID cards, security tokens and cell phones.

c. **Something the user is or does (biometrics)**

Users’ identity can be verified by reading and analysing their biometrics. Biometric characteristics can be divided into two classes:

- **Something the user is (physiological characteristics)**
  Also called static biometrics, these include fingerprints, hand geometry, eye features: retina & iris scanning, DNA sequence, face recognition and other biometric identifiers.

- **Something the user does (behavioural characteristics)**
  Also called behavioural biometrics or behavioumetrics, examples of behavioumetrics are: voice verification, written signature, mouse dynamics and keystroke dynamics.

Unlike other factors, physiological biometrics cannot be forgotten (they are part of the user) but, they require special hardware (readers) to capture these physical or behavioural traits. The UK Border Agency started a biometric data collection for visa applicants. They claim that enrolling fingerprint data helps protect applicants and their families from identity theft, crime and terrorism and gives them fast passage through automatic gates of entry at UK ports and airports (ukvisas.gov.uk).

However, from the use of people’s biometrics arise many privacy and discrimination issues. Misuse of the technology might put owners of important items in danger, for instance, the BBC reported in 2005 that the police in Malaysia were after a violent gang who chopped off a car (a Mercedes S-class) owner’s finger to get around the vehicle’s hi-tech security system (Kent, 2005).
In addition to the above, other authentication factors exist and can be grouped into the following additional classes:

d. Location-based authentication

The system allows access to connections from a specific approved location only. For example, connections to a server from an approved office are accepted, the rest are rejected.

Knowing the physical location of a particular user or network node at any instant time can be achieved by relying on a location signature that is created by a ‘location signature sensor (LSS) from the microwave signals transmitted by twenty-four satellite constellation of the Global Positioning System (GPS)’. In a hypothetical example, this could prevent a hacker based in Russia to connect to a funds transfer system in the US while pretending to come from a bank located in Argentina (Denning and MacDoran, 1996).

e. Time-based authentication

The system allows access to resources during a specific scheduled time. Granted privileges can also be terminated after a certain amount of time using a time-based authentication system.

f. Social networking

In this approach, to authenticate, the user is expected to successfully respond to a random challenge based on social questions. An example to this is a photo-based framework called Lineup, presented by (Yardi et al., 2008).

Lineup uses the social network graph in Facebook with additional information such as the ‘tagged user photos’ to dynamically build and present social questions (e.g. the system presents a photo and asks the user to identify a subject or a person in it). Hence, these questions are used to authenticate people or verify their membership to certain community groups, but never used to restrict access to sensitive data such as bank accounts. Lineup is discussed more within section 2.5.3.
g. Web of trust

The concept was first introduced by (Zimmermann, 1995) and implemented in Pretty Good Privacy (PGP) to verify whether a certain Public Key (PK) belongs to the originated username or not.

The traditional way to do that was to contact the originator directly, verify his identity and then get the properties of his key, but that is exhausting and time consuming.

With web of trust, it is possible to authenticate/trust the key when there are enough other trustworthy key-holders who signed the key before. In other words, if A trusts B and C, A doesn’t know if D is reliable yet, however if B and C trust that D is reliable, then A has a reason to trust D as well. (Caronni, 2000) discusses and evaluates the web of trust in comparison with a centralised hierarchically oriented approach.

Strong authentication

The U.S. Government's National Information Assurance Glossary defines strong authentication as: ‘Layered authentication approach relying on two or more authenticators to establish the identity of an originator or receiver of information’ (CNSS, 2006).

Authentication schemes are typically built on one factor. But, to strengthen the authentication mechanism, many solutions combine factors from two of the previously mentioned classes together in what is known as two-factor authentication or strong authentication. An example is Citibank Tokens (WS&T, 2006), where the customers are authenticated with something they know in addition to something they have, a security token in this case.
2.2. Text passwords

A text password is a secret string of letters, numbers and symbols shared between two parties to establish authenticity between them. Passwords are sometimes called passphrases if formed from multiple words or PINs if it was based on a short number of digits (usually 4 to 6 digits). In computer systems, this is used in combination with a username to control access to resources.

2.2.1. The text password problem

Text passwords fall into the first category (something the user knows). No additional hardware is required which makes this solution cost effective and more usable. This feature, in addition to its simplicity, makes it still the preferred approach to control access to web applications and the vast majority of systems (DTI, 2006). However, the use of text passwords is problematic and has many drawbacks, which undermine the effectiveness of the approach (Jobusch and Oldenhoeft, 1989).

Many problems were revealed and pointed out by previous investigations, for example: (Charoen et al., 2008), (Kessler, 1996), (Adams and Sasse, 1999), (Klein, 1990), (Furnell, 2005) and (Spafford, 1992). These drawbacks are presented below:

a. **Memorability**: One of the core problems with text passwords is that users often forget them, and to avoid this, they develop insecure practices (e.g. write them down or select a password that is easy guess).

b. **Passwords are badly selected**: To overcome memorability problems, users choose passwords which are easy to guess or predict by other people. For example, using personal information such as the name of someone's daughter which can be guessed easily or cracked by a dictionary attack (vulnerable to cracking tools). Cheswick and Bellovin pointed out how weak passwords are the most common cause for system break-ins (Cheswick et al., 2003).

c. **Users write down their passwords**: Users save their passwords written somewhere accessible to overcome a possible memorability problem. This approach could imply that the written password can be found by other people as well.

d. **Passwords can be shared**: People share them with their colleagues. Furthermore, studies highlighted that office workers are willing to give their
passwords in exchange for simple things such as pens (Leyden, 2003) or sweets (BBC, 2004). Sharing passwords at work should be against companies’ policies.

e. **Social engineering can succeed:** Text passwords can easily be shared with other people. An example for this is an employer exploited by an attacker with a convincing ability to recover their password. In this scenario, the attacker can pretend to be a privileged person (e.g. from the company’s IT department). As such users must be trained well on security policies.

f. **Passwords are infrequently changed:** If the password remains the same for a long period of time, then cracked or previously shared passwords will open the system to illegal access for a long time as well.

g. **Passwords are reused:** The same password is reused in multiple systems. People started to deal with many passwords on different systems as computer driven devices became part of their daily life. A survey of user attitudes in 2000 showed that 26% of respondents claimed to deal with five different systems or more, with 18 people claiming in excess of ten systems (Furnell et al., 2000). Thus people find it more practical for them to use the same password again on other systems. This is dangerous because it means that a breach on one system will cause other systems with the same password to be vulnerable too for that particular attacker.

h. **Cost:** Rather surprisingly cost is an issue as well. While the use of text passwords is virtually free, many companies encounter costly procedures in relation to password reset and related administrative issues. For example, studies have calculated that every call made to the IT help desks costs an organization approximately $25 (£15) and it has been estimated that up to 80 percent of the technical support calls made to IT help desks are in relation to forgotten passwords. (Hayday, 2003)

More statistics to prove the validity of the points explained above are in a survey on how easy systems based on text passwords can be compromised (Furnell et al., 2000). The survey shows that in many cases, the respondents themselves were compromising password protection, with 15% admitting to writing them down and 29% willingly sharing them with colleagues. In addition to this, 31 (21%) of the 151 respondents who used computers at work claimed to have used another person’s password without their
consent or knowledge’. Further statistics from an up-to-date survey (part of this thesis) are presented in Chapter 3.

2.2.2. Countermeasures and guidelines

The following countermeasures have developed over time to keep text password vulnerabilities under control. They fall into the following main categories:

1) Preventative measures

That is, taking proactive security measures to identify weak passwords before an attacker can reveal them with a cracking tool. However, this approach is time and resource consuming. Examples of the proactive password techniques are: (Bergadano et al., 1998) and (Blundo et al., 2004).

2) Increase the computational overhead of cracking passwords

Utilizes techniques to make cracking passwords harder, for instance, (Manber, 1996) scheme can be used to make guessing passwords based on one-way functions 100 to 1000 times harder.

3) Single sign on solutions

This kind of solution appeared to reduce the number of passwords users need to recall for several systems. For example, (Passlogix, 2008) v-go Single Sign-On saves users’ credentials and then automatically logs the user to password-protected Web sites and applications.

4) User training and strict policies & guidelines

Help and direct guidance to end-users is crucial to assure building secure passwords especially with the growing number of online services and their importance, (Adams and Sasse, 1999) and (Furnell, 2007).

This can be achieved by adopting strict password policies and educating end users to enhance their security awareness.

Proper guidelines can be found online to help adopt good security manners. The following points of advice are discussed in (Furnell, 2003) and (Belgers, 1993) as they are usually given to users with useful short explanation of why is it important to do (or not to do) things:
a. **Do not choose dictionary words**: This includes all words of a particular language, derivatives, given names, geographical names and encyclopaedia terms. Reversed dictionary words should also be avoided. Password cracking tools have built-in dictionaries and can match these within seconds.

b. **Do not choose patterns**: For instance, 123456, 24816, zyxwvu etc. Cracking tools can match these too.

c. **Do not choose passwords that can be guessed**: Avoid passwords that can be guessed by someone who knows you or by someone prepared to do research about you. Examples here include names of partners, children, pets, the license plate of your car, room number, phone number, elements of your address, etc. Also make sure you do not include a substring of the username or company name in the password.

d. **Do not share passwords**: Do not share your password(s) with anyone, this includes friends, family and colleagues. Anyone who knows your password can get access rights and privileges on the system using your name.

e. **Do not write down passwords**: Unless you are storing your password(s) details in a secure location, do not write them down. Someone else could find and use them as well.

f. **Do not use the same password for more than one system**: This way if an attacker cracks one password, he doesn’t get access to more than one of your accounts.

g. **Do not choose passwords less than eight characters in length**: This is important to increase the password space and protect the password against brute force attacks.

h. **Do not choose passwords consisting of alphabetical letters only**: To increase the effective password space, it is also critical to include a combination of alphabetic (mixed case is advised), numeric and punctuation symbols all together.
i. **Do change your password on a regular basis:** Changing passwords regularly, once a month for instance, helps to protect your account(s) in case your current password(s) have been obtained without your knowledge. Nevertheless, you should not use a previously used password again to avoid exposing your account to possible illicit access.

j. **Do consider using a technique to recall your passwords more easily:** It is advised that a scheme is adopted to help remember text passwords. For example, a user can choose the first letters of a phrase that is easy to remember; thus if the phrase is ‘Learn from yesterday, live for today, hope for tomorrow’ then the password is ‘lfylfthft’. This is called Mnemonic phrase-based password. However, (Kuo et al., 2006) shows that most users select phrases from music lyrics, movies and television shows which can easily be found on the internet and included in lookup dictionaries, so it will be a good idea to apply some changes such as replacing ‘for’ with the digit ‘4’. This will change the password to ‘l4yl4th4t’. Incorporate a symbol as well somewhere in the string, and you have a random and totally meaningless looking password: ‘lfyfthft?’.

Formulas to generate personalised passwords have been proposed such as the Simple Formula for Strong Passwords (Thomas, 2005) and Magic Formula for Passwords (Anctil, 2009). They suggest different ways to reform an easy to remember string to make it more random by containing a variable that is logically linked to the username or the name of the website. For example, “u08n05iGversity” is based on the keyword “university”, including a personal date of birth 08/05 after the first and second letter, while G is consequently used after the third letter taken from the word Gmail, because this password is being used in a Google mail account. “G” in this case is a simple example of a logical variable used inside a static string. Nevertheless, some algorithms such as (Lee and Lee, 2007) are implemented within an application to help users generating strong passwords for every account out of a simple user-specific salt. These are called: password stretching algorithms and the word salt is used to refer to any simple string that is insecure if used alone as a password.

All the above classes of solution are critical and important to control the problem, but do not remedy the main cause of text password insecurity, which is the human limitation of memory for secure passwords as shown in this section. Chapter 3 of this thesis presents a survey showing how the majority of end users still do not adopt the majority of
these guidelines. Nevertheless, these solutions are not convenient at times as a big company with a large number of employees will need to invest time, money and human resources to guide their users into good security practices. After all, even if done properly, this will not guarantee that users are going to comply with security polices at all times.

A strict password policy such as the one forcing users to periodically change their passwords could increase the number of users who write them down which oppose another guideline from the same policy. The password problem has been a main research area in computer security since the early use of time-sharing systems (Morris and Thompson, 1979). However, the issue is still a serious security concern because it is part of human nature to choose what is easy and more suitable to one’s daily use. Hence, people will always prefer usability and convenience over security (Dhamija and Perrig, 2000).

2.3. Password strength

Password strength is a measure of a string’s resistance to guessing, dictionary and brute force attacks. This depends on the length of the password, the number of available symbols to form the password and finally the randomness of selecting these symbols.

Some authentication systems force a minimum length of password to maintain an acceptable level of password strength. The number of possible symbols is increased by using symbols from a bigger set such as using ASCII symbols instead of relying on alphanumerical values only. Randomness is achieved if there is a uniform distribution of the available symbols among passwords, which implies that the probability to select each symbol is equal.

2.3.1. Password space

The password space is the number of password combinations available for use by users in a system. For instance, the PIN system used in ATM machines consists of 4 Arabic numerals (0-9); hence, the password space is $N^L = 10^4$ where $N$ is the number of possible symbols used to create the password and $L$ is the length. Similarly, if $N = \{a, b, c, \ldots, z\}$ the password space is $26^L \approx 2 \times 10^{11}$ this is called ‘The Theoretical Password Space’, while in practice these combinations are not equally used by end users (the distribution of symbols is not uniform). For example, assuming that most users will be selecting a combination of letters reflecting an English word (since they are easier to recall), the number of possible symbol combinations will decrease in this scenario
from $2 \times 10^{11}$ to the number of words in the English dictionary which is only about three
quarters of a million words, as stated in the Oxford Dictionary (AskOxford, 2009). In our
example where $N = 8$ this is less than 0.001% of the available space, hence the vast
majority of selected passwords are contained in a much smaller subset of the theoretical
password space, called: ‘The Effective Password Space’.

Password systems, including VP schemes have both theoretical and effective
password spaces. Hence, in systems where the symbols are poorly distributed, the
number of symbols available in the effective space $N_e$ is much less than the number of
symbols available $N_T$, i.e. $|N_e| \ll |N_T|$. 

If the probability of $A = P(A) = \frac{1}{|N_T|}$ where $A \in N_T$

And $P(B) = \frac{1}{|N_e|}$ where $B \in N_e$

Then $P(B) \gg P(A)$

Also $P(B) \approx 1$ while $P(A) \ll 1$

2.3.2. Entropy (bit strength)

The entropy or bit strength translates the language (the number of possible symbols) into
binary digits (0 or 1) (Shannon, 1951, Brown et al., 1992). Hence, instead of saying that
cracking a password ($N = 26$ and $L = 8$) requires approximately $2 \times 10^{11}$ attempts to
exhaust all possibilities, the following formula is employed:

$$Entropy = L \frac{\log N}{\log 2} = 8 \times 4.7 \approx 37.6 \text{ bits (Burr et al., 2006)}$$

This implies that the password is as strong as a binary number of 37.6 random bits.
Therefore, the entropy is also used to measure randomness or the amount of uncertainty
in an attack to reveal the password.

2.4. Type of password attack

Different techniques are used by intruders to gain access to a password protected
system. For example, some of them rely on revealing the password of the targeted
account from the hash taking a cryptanalysis approach, while other strategies rely on
stealing the password during entry time. Existing authentication systems are designed to
prevent these attacks, but this is hard to achieve since new vulnerabilities are discovered
and attack methods are developed constantly. The human factor is another concern
because users’ failure to keep their credentials safe breaks a relatively secure system. The following paragraphs identify and discuss password related attacks in further detail:

2.4.1. Shoulder-surfing

This attack works by observing the targeted users while entering their credentials using a keyboard, touch screen, computer mouse or any other input device. A traditional shoulder-surfing attack has a better chance to succeed if the authentication session happens in a public place or beside other individuals. However, electronic recorders such as pinhole cameras (Summers and Toyne, 2003) can be used to learn passwords in private environments. These modern tools raise more concern because they are hard to detect.

Systems vary in their resistance to shoulder-surfing; vulnerable solutions reveal the password with a single successful observation attempt, while others require watching the login process many times to capture the required data.

Different techniques are used to prevent shoulder-surfing attacks. They differ from brief guidelines where users are advised to enter their PINs with one hand while using the other hand to hide the keypad to more complicated techniques such as (Roth et al., 2004) where users are never asked to enter their PINs directly, instead, the authentication challenge is a set of questions which change with every login attempt and which can only be answered by someone familiar with the original password. Many other resistant methods have been proposed such as (Wiedenbeck et al., 2006) which is a VP solution resistant to shoulder-surfing and (Takada, 2008) (Shi et al., 2009) schemes to prevent observation attacks using video cameras.

2.4.2. Guessing

A text password or the security questions used for password recovery can sometimes be guessed by attackers. Text passwords vulnerable to human guessing are all the personal and impersonal predictable terms such as: “password”, “123456”, the username or part of it, child’s name, date of birth, home telephone number, ZIP code etc. Default passwords in system setups should immediately be changed by end users too, as there are lists of default passwords available online for all known applications and services sorted by models and version codes, e.g. (phenoelit-us.org, 2010). Personal information that can be searched or guessed by friends and family should be avoided as well. This includes good passwords which have been shared with people before. A survey questioned 1,200 employees found about half of them were using family-related
information as their login such as their own name, nickname or names of their children, pets and parents (McAuliffe, 2001).

An example to show how personal information can be used to gain unauthorised access is the hacking of the US Vice Presidential candidate Sarah Palin’s email address, in which the attacker claimed to use Google and Wikipedia to collect and successfully answer her account’s security questions (Birthday, ZIP code and “Where did you meet your spouse?”) in order to reset the password (Danchev, 2008).

2.4.3. Dictionary attacks

In a dictionary attack, the password is automatically looked up in a list of possibly used words. For example, a computer programme like L0phtCrack (l0phtcrack.com, 2009) and John the Ripper (openwall.com, 2009) could perform a fast guessing process using the English dictionary by trying every word as the password until the right one is reached. If the password belongs to the English language then it will be guessed correctly. It is also possible to perform a hybrid dictionary attack where multiple dictionaries (e.g. dictionary of default passwords, dictionary of the most used computing terms etc) are used together to crack the password. (Klein, 1992) performed a dictionary attack on nearly 15,000 account entries and successfully cracked 25% of the passwords at the end of the test.

Dictionaries can also be used to crack password hashes. These are known as Rainbow Tables and are used to offer a time-memory trade-off solution to reduce cryptanalysis time to recover passwords from the cipher (Oechslin, 2003) (Avoine et al., 2008). They contain pre-computed hashes for all possible combinations along with their original values. To crack a hash, it is compared to the hashes starting with the closest possible block of data until a match is found throughout the algorithm, and the password will be the value used to compute that hash.

The best way to stop dictionary attacks is to use a completely random password to give the password a large effective password space. If the password is not found in any dictionary, this attack fails.

2.4.4. Brute force

In a brute force attack (also called exhaustive search), every possible string combination is tested until the right password is found. Hence, in theory, cracking the password is guaranteed after a certain amount of time if the whole password space is covered, while in practice, if the password space is large enough, the attack becomes useless due to
time and processor speed constraints. A fast computer having the speed of checking 10,000,000 string/sec will take around 20 years to crack an 8 character long password consisting of a mixed upper and lower case alphabet with numbers and common symbols. However, the possible cracking speed currently is much more than this as mentioned at the beginning of 0.

Online brute force attacks can be fought by implementing human verification methods (e.g. CAPTCHAs) and other challenge-response techniques to block computer bots and malicious scripts. In addition, some systems lock the account after few rejected attempts. Although, defeating this attack by locking accounts can be exploited to perform a Denial of Service (DoS) attack against the service provider (Pinkas and Sander, 2002), in which attackers enter incorrect passwords on purpose to postpone the service.

The other possible scenario is when the password hash is obtained, in this case, the attacker could perform powerful offline attacks using multiple processors and super computers to reduce time. Hence, having a large password space is the best approach to defend the password.

2.4.5. Social engineering

In social engineering, the attacker communicates with people working on the targeted system to obtain confidential information. A possible tricky scenario is if an intruder pretends to be working for the IT department in a big company, he might be able to convince an employee over the phone to give or reset his/her password to a default one.

Social engineering techniques rely on the attacker’s ability to convince as well as the target’s common sense and awareness of security policies (if there are any). However, some systems are more resistant to social engineering than others; for example, sharing a text password over the phone is possible and can be easy, while sharing VPs is a more complicated procedure because they can only be described. A further detail about VPs’ resistance to social engineering is discussed in the coming sections of this thesis.

2.4.6. Phishing

Phishing is a fraudulent attempt to obtain sensitive information such as credit card numbers and user credentials by emulating electronic services familiar to the targeted user (Dhamija et al., 2006). It is an example of a social engineering technique where technology is used rather than direct human networking. In the most famous form of the attack, a clone of the sign-in page of a popular website (e.g. PayPal) is used to trick users into entering their credentials. The fake website can then display an error page
and forward users to the original website to avoid raising their awareness of what happened. Another way is to send an email pretending to be from the person’s bank and acquire private information. However, new techniques emerge as people grow wiser about older fraud methods, for instance, it has been reported that phishers are using voice-response systems through VoIP instead of email addresses warning customers about a problem with their bank account and asking them to enter their credit card numbers to resolve the problem (Evers, 2006).

People should be very cautious while sharing their private information, common sense is the first line of protection against forgery. In the case of bank accounts, it is always good to dial the bank official number to use their services rather than calling or responding to unknown numbers and ID callers.

Some authentication systems require a random part of the password at a time to resist phishing. However, such systems are still vulnerable to a more advanced attack called Man-in-The-Middle (MiTM) phishing. An example of this is when the fake sign-in page upon request establishes a connection with the legitimate site and forwards its challenge to the user, the user’s response is then communicated back until authentication succeeds. If the response is correct, the attacker then gains direct access exploiting the end user’s information directly during a live session.

2.4.7. Malware

The word malware is short for ‘malicious software’, which is designed to penetrate computer systems to perform malicious actions without the owner’s permission (Skoudis and Zeltser, 2003). Malware programmes are categorised based on their usage. For example, rootkits (Butler et al., 2008) are used by attackers to maintain access to a hacked system. Spyware (e.g. keyloggers) on the other hand can be used to collect information about the user; these programmes can secretly record the keystrokes of a keyboard, the mouse motion or grab snapshots of the screen. (Ross et al., 2005) demonstrate JavaScript code used to steal passwords from internet pages and provide a browser extension to defeat such attacks.

Malware takes different shapes and approaches, so it is hard to detect at times. However, countermeasures can be taken by installing anti-malware programmes (or anti-virus coming with an anti-malware feature) and installing genuine applications from trustworthy companies only.
2.5. Introducing visual passwords (VPs)

VPs can be used to authenticate users in computer systems. The VP is formed of a user portfolio, click-points on images or a user drawing (signature-like) on a specified background.

Since most text password problems are memory related, VPs were proposed as a possible alternative. The idea emerged based on many psychology studies presented over the last century suggesting that humans can remember pictures and visual elements better and more accurately than text (Madigan, 1983) (Nelson et al., 1976) (Calkins, 1898) (Shepard, 1967) (Paivio et al., 1968) (Bower et al., 1975). For instance, (Calkins, 1898) experiments showed that people’s ability to recall words reduced by half or more over three days, while recalling objects dropped by 20% or less only for the same period of time.

Requesting users to select random (complex) text passwords raises the problem of the limitations of human long-term memory (LTM). Hence, the main objective of VP schemes is to maintain that memory while assisting users to select strong passwords. VPs are categorised and discussed in details consequently in the following sections, but in an introductory summary, some of the possible advantages of using VPs are:

1) Large password space: e.g. click-based systems.

2) Memory aid: provide cue to recall the password.

3) Unique passwords (singularity): passwords are related to the user’s portfolio images; hence if system administrators employ unique images, users won’t be able to reuse the same password on different systems.

4) Hard to describe: some VPs are hard to share without exchanging actual copies or snapshots, e.g. abstract images.

5) Unlike biometrics and tokens they do not require special hardware to work.

While the possible challenges are:

1) Time: they take more time to enter than text passwords, especially recognition-based schemes.

2) Shoulder-surfing: password entry steps are more visible on the monitor.

3) New approach: end users are not familiar with them.
2.5.1. Categorisation and listing of VP schemes

In the literature as in (Suo et al., 2005) and (Davis et al., 2004), VPs are grouped based on the type of memory task performed by people while using each scheme. The main categories are: pure-recall, cued-recall and recognition. This approach is adopted in the following sections to demonstrate available VPs. Inside each category, the schemes are sorted based on the year of publication from oldest to latest.

Main usability and security issues for each scheme are briefly discussed and evaluated in Section 2.5.5.

2.5.2. Pure-recall

Pure-recall schemes provide no cues to end users while entering their password, so they are quite similar to text passwords in that sense. The general idea behind this type of solution is to enable users to create a personal signature of drawings, which they must reproduce in future authentication attempts. The following are examples of proposed pure-recall VP schemes:

Draw-A-Secret (DAS)

Draw-A-Secret (Jermyn et al., 1999) is similar to signing papers in real life to prove the identity of the originator.

The authors were primarily motivated by personal digital assistants (PDAs) such as the Palm Pilot™, Apple Newton™ and others. DAS frees users from having to remember any kind of alphanumeric strings.

The scheme works by drawing a simple picture on a 2D visible grid and memorizing it for future sign-in attempts. Figure 2.1 illustrates a DAS password on a 4 x 4 grid. Using a proper graphic input device such as a stylus or a mouse, the user can sign-in by drawing lines accordingly. The system maps the user’s line-path continuously to the grid’s cells using their coordinates, hence, the line should avoid following grid lines and cells corners. For instance, the coordinate sequence generated for the user in Figure 2.1 is: (2, 2), (3, 2), (3, 3), (2, 3), (2, 2), (2, 1), (5, 5). Where (5, 5) is used to record a "pen up" event.

This is how the system records the event when the user stops the line to end his drawing session or to start another line. In conclusion, the DAS password consists of a
sequence of strokes separated by pen up events and each stroke is actually the sequence of cells on the pen path separated by pen up events.

![Diagram of a 5 x 5 grid with a path drawn through it.](image)

**Figure 2.1:** Input of a graphical password in DAS (Jermyn et al., 1999).

The theoretical password space for DAS is related to the number of cells included in the grid used. It can also be increased by making the signature longer (drawing more lines) by the user.

If a 5 x 5 grid is used, where the password length consists of a maximum of 8 lines, the theoretical password space is $\log_2(25^8) = 38$ bits (Jermyn et al., 1999). However, the entropy of the DAS password could be reduced if predictable characteristics are proved to exist. (Nali and Thorpe, 2004) examined user choice in DAS and found that most users focused on the centre of the grid where they draw certain symmetrical shapes such as crosses and rectangles in addition to letters and numbers. Their results also highlighted usability challenges as 29% of the passwords collected in their user study were found to be unsuitable for DAS as they followed grid lines or crossed through cell corners.

In an attempt to guide users toward better DAS passwords, (Dunphy and Yan, 2007) suggested adding background images instead of using blank canvas overlaid with a visible grid, so they proposed BDAS (Background Draw-A-Secret). Their user study shows that using background images helped to create more complicated passwords and reduced the predictable password characteristics in comparison with the original approach. However, the study did not discuss the possibility of attacking the system using image analysis techniques to build a dictionary of predicted passwords based on attractive hotspots.
All DAS user studies used simple paper prototypes, so more research work and tests are needed to evaluate the system properly in a computer environment. Nevertheless, some security concerns can be outlined such as phishing, social engineering, guessing and shoulder-surfing. Similar to writing text passwords on a paper, a user can draw his DAS signature and share with others. In addition, if used for web application for example, the interface can be copied to a fake host in order to perform phishing attacks, which implies that DAS alone does not provide effective solutions to stop security threats.

**Passdoodle**

The idea of Passdoodle (Goldberg et al., 2002) is generally quite similar to Draw-A-Secret, but it does not use a grid, instead, it depends more on free-hand drawings. The authors tested the approach using paper prototyping and reported that most users perceive the doodles as easier to recall than text passwords if they are not asked to follow the same order in which they draw the doodle. Nevertheless, they couldn’t redraw their doodles perfectly again.

Different techniques such as velocity and pen colours are suggested by the authors to add variability to the doodles and increase the password space.

In a small user study (Varenhorst, 2004), 10 participants were trained with a lightweight Passdoodle prototype to trace a drawing on a touch screen using their fingers, consequently the screen was cleared and they were asked to repeat the same doodle as nearly as possible. The results showed the system to be feasible with extremely accurate results, but the time needed by the users was not reported. Security was not discussed either.

A Passdoodle prototype for web application (Govindarajulu and Madhvanath, 2007) was then developed. In this system, users can sign-in to websites by drawing a single doodle (known as the master doodle) using a touchpad connected to their computers. The preliminary test of the system was done using Tamil symbols (handwritten Tamil characters) as examples for doodles. 10 participants were asked to register with one initial symbol (doodle) to be their master doodle, and then handwriting recognition techniques were used to compare future sign-in attempts with the relevant master doodle. Although one of their recognition techniques achieved an average accuracy of 90.56%, this might change with a large scale user study.

Using additional hardware (touchpad) restricts the system to those who own one. In addition, security and possible attacks against passdoodle were not discussed.
Available research papers focus on presenting techniques to solve usability and technical issues. However, by reviewing how Passdoodle works, possible security challenges can be expected such as shoulder-surfing and social engineering attacks. Similar to Draw-A-Secret, many users could draw predictable symmetries.

**Pass-Go**

The idea and name of this scheme was inspired by an old Chinese game called Go (Tao and Adams, 2008). To create a password, users draw dots and/or lines on the grid’s intersections using different colours. The direction and order of the drawing is recorded, users should then recall it correctly to sign-in.

Figure 2.2 shows a Pass-Go sign-in form. The password in this example consists of two red dots, two black dots and two vertical blue lines. This can be represented by text as follows:

Red (3,7) | (3,6) | Black (4,7) | (4,6) | Blue (3,6) (3,5) (3,4) (3,3) | (4,6) (4,5) (4,4) (4,3) |

Where " | " is used to indicate a “pen up” event.

In a comparison with text passwords and other VP systems, Pass-Go has a large theoretical password space equal to 256 bits for the most basic scheme (Tao and Adams, 2008).
To protect the password against shoulder-surfing attacks, the authors suggested a variety of techniques. For example, random lines and dots can be displayed in response to each user entry during the sign-in session to confuse attackers. In addition, entering a line or dot can be associated with their hidden events such as a button on the keyboard. While these techniques can be useful to fight shoulder-surfing, their efficiency has not been tested.

The results of a field study show that 78% of 6800 login attempts were successful, so the system’s usability was assumed acceptable by the authors. The study also shows that 49% of the passwords are associated to known patterns, either alphanumeric or other well known symbols, hence, patterns is a problem. (Oorschot and Thorpe, 2008) developed a method to predict such weak passwords for drawn graphical passwords.

An interesting advantage of the system is that, it has keyboard and textual display support. By using the number keypad on any computer keyboard, there are 8 numbers surrounding the centre where number 5 is located, these are used to highlight and move the “current intersection”. Other keys are assigned to perform other events, for example, the “Enter” key can work as the left button of the mouse. Using special encoding to represent the password textually is also possible to allow entering passwords using a command line (to connect to a SSH server for example), but such code is hard to remember.

2.5.3. Cued-recall

In cued-recall solutions, the system provides a cue to trigger the end user’s memory while entering their password. It aids LTM to repeat the task (the VP) successfully. For instance, the task could be clicking a specific area inside a picture of a playground full of children; in this case, a possible cue is the hand of one of the children. A user will find it hard to recall the coordinates of the mouse to click the same spot again, but easier to recall the child’s hand to perform the right click in future attempts. The following is a survey of cued-recall VP schemes:

Blonder’s scheme and Passlogix

(Blonder, 1996) proposed a VP scheme where users credentials are a sequence of clicks on predetermined tap regions of an image. The tap regions can be represented by a variety of easy to recognise objects. For example, in Figure 2.3 some of these objects can be the clock hanged on the mirror, a necklace on the table, the hand watch on the bed and many others.
To register, users should click a number of the predetermined regions and memorise them in sequence. To sign-in, users should replay the clicks correctly using the objects as cues to where they have clicked before.

According to (Suo et al., 2005), Passlogix (Boroditsky, 2002) is an implementation based on Blonder’s idea. Figure 2.3 is a snapshot of Passlogix. However, the system is no longer available as Passlogix Inc. stopped developing this particular method and shifted their business in another direction.

![Passlogix](image)

**Figure 2.3:** Passlogix, implementation of Blonder’s scheme. Taken from (Suo et al., 2005).

The theoretical password space of the system is relatively small due to the limitation of the number of objects which can be contained in a single picture. With 30 predefined objects and 5 clicks, the password space is $30^5 = 243 \times 10^5$. The entropy is $5 \times \left( \frac{\log 30}{\log 2} \right) = 24.5$ bits.

The password can be easily shared, hence social engineering has a good chance of success as the objects can be easily described.

**Inkblot Authentication**

This system uses inkblots images to help users select and remember random passwords. The idea was originally taken from the Rorschach inkblot test, a well-established psychological test in which people are shown different inkblots and asked to name what comes to their minds after observing the blot. This test aims to examine personality characteristics and emotional functioning. However, since there has been an evidence from psychological literature that people tend to associate inkblots with dissimilar objects in their minds and remember the associations for a long time as part of
their long term memory (LTM), Inkblot Authentication employs inkblots to generate random yet easy to remember text passwords (Stubblefield and Simon, 2004).

In the authentication scheme there is an algorithm responsible for feeding the system with new inkblots for new users. Figure 2.5 shows randomly generated samples. During registration, users are presented with a number of inkblots and asked to associate them with objects in their minds, thereafter; letters of each object is used to create the new password. For example, if a user mentally associated the inkblots with the following objects: bat, skeleton, lion and a flower, the password can be: btsnlfr (the first and last letter of every object combined together in a string).

To sign-in, users either recall the password alone, or rely on the inkblots images as a cue to recall their password correctly as seen in Figure 2.4.

In a user study, the authors of the system reported that users missed at most one associated object out of 10 after one week’s time. A possible suggestion to solve this problem is to accept a password with a single association mistake, while asking users to enter associations for more blots which will increase the password entry time and could affect memorability.

The theoretical password space of the system is $26^{20}$ if the English alphabet is used to name 10 associated objects where two letters are taken from every object. And the password entropy will be: $20 \times \left(\frac{\log 26}{\log 2}\right) = 94$ bits. This large theoretical password space
makes brute force attacks hard to perform. However, dictionary attacks have an advantage since users are limited to the first and last letters of an English word in the proposed scheme. Results analysis of the user study revealed that some letters and character pairs are used more than others due to their frequency in the English language.

Observing the inkblots on the screen is not useful to recover the password. Hence, shoulder-surfing attacks should focus on the keyboard to succeed.

**Passpoints**

Passpoints (Wiedenbeck et al., 2005c) is another implementation of Blonder’s idea. But, instead of restricting the possible click-points to predetermined regions as seen with Passlogix, Passpoints allows users to click any pixel of an image and accept any replay of the click within a tolerance distance of the original click-point since it is very hard to locate the exact pixel again. Hence, this approach grants a larger number of possible click-points than that of Passlogix, which results in a bigger theoretical password space. For instance, a 451x331 pixels background image with 20 pixels tolerance distance accepted around the original click, gives a \((451 \times 331)/(20 \times 20) = 373\) possible click square, hence, 5 clicks on that image give a theoretical password space of \(373^5 \approx 7 \times 10^{12}\). The entropy in this situation is \(5 \times \left( \frac{\log 373^5}{\log 2} \right) = 213.5\) bits.

To increase the effective password space, feature-rich images should be used to help provide the end users of the system with an acceptable number of memorable click-points.

In a user study of 40 participants (Wiedenbeck et al., 2005a, Wiedenbeck et al., 2005c), the results show that with Passpoints users had more invalid password entries and spent more time than users with text passwords. The average VP creation time was 64 seconds in addition to 171 seconds used to train users to memorise their new password. The approximate sign-in time was between 9 and 19 seconds only. Success rates varied between 55% and 90% on different sign-in events throughout the experiment.

In an attempt to study the effect of the background images employed and the tolerance distance length in Passpoints, (Wiedenbeck et al., 2005b) shows that usability is extremely reduced when using a tolerance as small as 10 pixels, hence, allowing enough tolerance is advised. The results also suggest that the differences in
performance between different background-images were very little, which indicates that many images can be easily found to be employed in Passpoints-like systems.

More security and usability issues about Passpoints and other click-based systems will be discussed in further details in the following sections and chapters of this thesis.

SFR Password (visKey)

SFR Password (SFR IT Engineering, 2009) is a new commercial version of a previous application known as: visKey, reported in (Li, 2009). The system provides visual key authentication in addition to the traditional methods: PIN and Text Password.

The developer claim that it has a finger-friendly user interface and supports Windows Mobile. The system is similar to Blonder’s scheme and Passpoints as discussed before, but with more flexibility and user control, for instance, it allows users to employ their own pictures and then create the tap regions. Users can even decide the tolerance or the size of their click-cell while generating their VP. Despite being advised not to use big cell-sizes, people select usability over security which reduces the password space and makes password entry more vulnerable to shoulder-surfing attacks. Accepting personal images implies that the same image (hence password) can be reused on multiple systems and that poor images can be employed, which makes the system more vulnerable to guessing and dictionary attacks.

3D Password

The 3D Password (Alsulaiman and ElSaddik, 2006) is a multifactor authentication virtual environment. It enables users to navigate in a 3 dimensional building to perform actions and interactions with different objects. For example, it is possible to locate a biometric or token reader inside a room in the \((x_1, y_1, z_1)\) position to authenticate, or use a specific virtual computer to type a password (PIN, text or VP) in the \((x_2, y_2, z_2)\) position. Nevertheless, there could be pictures and white boards hanging on the wall and other locations to be selected and used in cued or recognition visual authentication systems. Figure 2.6 is a snapshot of proof of the 3D concept.

To authenticate, users’ password will be the sequence of all actions, interactions and inputs with the available objects inside the virtual environment.
Figure 2.6: Snapshot of proof of 3D Password concept. A virtual art gallery hosts 36 pictures and 6 computers. Users can navigate and interact with virtual objects (Alsulaiman and ElSaddik, 2006).

The password space is very large because the system is a host to various existing authentication schemes combined together. However, password security depends on user behaviour and the objects implemented in the system. For example, if a single recognition-based system is used, shoulder-surfing is a problem, while the chances of this attack succeeding are reduced if the authentication process includes a fingerprint reader. The authors’ user study did not report authentication time, but it is expected that loading the graphical environment is an issue especially if the connection is relatively slow. In addition, navigation and using multiple authentication methods is a time consuming process which raises serious usability questions.

CCP and PCCP

Cued Click Points (CCP) (Chiasson et al., 2007) is a click-based system proposed as a secure alternative to Passpoints. The password is a sequence of clicks on a number of images, but users are limited to a single click on each image. In addition, each click-point leads to a path of a different sequence of images as illustrated in Figure 2.7.

The authors claims that increasing the number of images intensifies the workload and makes attacks against the system time consuming for malicious parties. Also, user perception from the experiment conducted concludes that remembering a single click per image is easier than selecting more click-points. Nevertheless, the association between user clicks and the upcoming sequence of images triggers their memory and helps users to recall their click-points correctly.
The results of CCP laboratory study experiment (with 24 participants) show that high success rates have been achieved (98% for profile creation, 83% for password confirmation and 96% of successful sign-in attempts). It took users below 10 seconds on average to enter their CCP password, while registration time was more (around 24 seconds in average).

Persuasive CCP (PCCP) (Chiasson et al., 2008b) was then introduced where a new method has been developed into the original scheme to encourage and influence user choice to achieve randomness in click-point distribution.

Whilst CCP users can choose any click-point, the system in PCCP randomly highlights and suggests a specific area inside the background image during registration. However, if that area has no memorable click-points, a ‘shuffle’ button can be clicked to relocate the highlighted area.

PCCP performance has been tested in a lab study with 39 participants. The authors’ evaluation suggests that end users had to spend more effort learning their password because they were selecting less attractive click-points. However, they were able to remember their passwords. In a comparison with CCP, there was no significant difference in success rates and sign-in time. Hence, PCCP increased security (randomness in password selection) and maintained an acceptable level of usability. More security and usability issues of click-based systems are discussed in the coming sections and chapters of this thesis.

**Lineup**

Lineup (Yardi et al., 2008) exploits the ‘social network graph’ in Facebook aided by information such as the ‘Tagged user photos’ to build a lightweight web authentication mechanism. The system is implemented as a Facebook application widget that can be
embedded into external websites. And it dynamically generates relevant social questions, which can be answered correctly by some authorised group users.

This approach is not meant to restrict access to sensitive data; instead, it aims to provide a reasonable level of control to restrict resources to the members of a certain community group, and provide multi-access levels: low, medium and high. For example, if a website wants to dedicate one of its services to a particular university team, the system could present photos from the team’s private Facebook group and authenticate those who can identify the tagged people and objects correctly. On the contrary people who are not connected with the team are not expected to pass this challenge. Hence, in this system, access rights can be given to certain Facebook group, network or event. As such this can be used to maintain an acceptable level of authentication to provide quality services (e.g. fast download) to the targeted people and protect privacy (e.g. share pictures with friends only) without requiring initial sign up process or remembering passwords. The system gives no guarantee that unwanted individuals will not gain access, especially if Facebook privacy settings are not set accordingly.

As reported by the authors, the Denial of Service (DoS) attack is a real concern to this scheme, because a malicious user could upload irrelevant pictures and perform misleading tags to confuse legitimate users and prevent them from being authenticated. To prevent this from happening, there should be a mechanism to approve the pictures and tags by an administrator before passing them to the system.

2.5.4. Recognition

In recognition-based systems, people’s ability to recall what they have seen before is employed to authenticate users. To register, users are asked to select a subset of images from a pool and memorize them. These objects are used to create a unique portfolio for every user. To sign-in, users are expected to recall these objects correctly and select them among other randomly displayed images. However, different schemes have been proposed with different implementations of the approach as seen below:

**Passfaces**

Available for personal and commercial use, the Passfaces (Real User Corporation, 2000) scheme authenticates users by employing the human ability to recognise a face they have seen before, as in Figure 2.8. Authors claims that human faces are used based on research work suggesting that to a great extent the brain’s ability to recognise previously
seen human faces is better than recognising other shapes and objects. Hence, using faces reduce the chances for users to be locked out of their account.

The authors assumed that passfaces passwords can not be written down, copied or given to another person, but in practice, the password can be described. In fact, describing a human face is something people do all the time as part of their conversations. However, (Dunphy et al., 2008) shows how passfaces vulnerability to description can be reduced. Nevertheless, a print screen image is always an option if someone decides to share his passfaces portfolio.

To enrol, users select a face from a grid of multiple images provided to them by the system. To complete the enrolment, they should practise identifying the correct face a number of times (at least 4 times) successfully. Enrolment time is between 3 and 5 minutes (for 5 rounds using grids of 9 images) as reported by Professor Tim Valentine, Head of Department of Psychology at Goldsmiths College, University of London (Real User Corporation, 2000). The same reference suggests that data collected from the Passfaces Web site shows that users who sign-in successfully within two weeks of registration can also sign-in again after six months even if they are not very familiar with the system.

Figure 2.8: Passfaces (Real User Corporation, 2001).
In a user study (Brostoff and Sasse, 2000) of 34 users, the results proved that passfaces are easier to remember and have better success rates, but take more time to use in comparison with text passwords.

(Davis et al., 2004) implemented a clone of the system and conducted a large field study. The results show that user-choice can be predicted. For example, many people tended to select beautiful female faces of their own race. This implies that personal information like gender, race and the attractiveness of the face can be used to guess the passface.

To examine VPs' vulnerability to shoulder-surfing, (Tari et al., 2006) tested passfaces with 20 participants and found the system to be vulnerable when the computer mouse is used to select images and significantly less vulnerable when a keypad is used. With a keypad, an attacker should capture both keystrokes and the screen, which is harder than attacking text passwords where key logging can succeed alone.

Déjà Vu

The Déjà Vu (Dhamija and Perrig, 2000) approach is similar to that of Passfaces. To register, users are asked to create a portfolio (as shown in Figure 2.9) by selecting Random Art (Perrig and Song, 1999) images from a pool generated by a computer programme called Random Art developed by (Bauer, 1998). Random Art images are abstract images and unlike photographs, they are hard to describe.

The authentication process consists of a number of steps where a number of images are displayed; in each step, users should recognise and identify the correct image that belongs to their portfolio in order to sign-in successfully.

![Figure 2.9: Déjà Vu, portfolio selection window (Dhamija and Perrig, 2000).](image-url)
A user study with 20 participants conducted by the authors shows the system to maintain memorability. Users can remember their Déjà Vu passwords better than text passwords after a week, although they took longer to enter. Users needed 45 seconds in average to create Déjà Vu passwords and 32 seconds to sign-in compared with 15 seconds to use a text password in the same environment. After a week time, users were asked to login again to the system, they needed 36 seconds to complete the process achieving 90% success sign-in rate.

The password space is small compared to text passwords. With a portfolio of 5 selected images and a pool of 25 images there are 53,130 possible combinations, which is equivalent to using a 4 to 5 digit PIN (Dhamija and Perrig, 2000).

Using abstract images benefited the system in many ways, for example, it makes it hard for end users to describe the password and share it with others over the phone, and this could help to prevent social engineering to a certain level. In addition, the images are easy to get because they are generated by software.

Phishing and shoulder-surfing attacks cannot succeed from the first attempt, because only part of the password is used to sign-in every time. Hence, more attacks are required to recover the password.

**Visual Identification Protocol (VIP)**

The main objective behind developing VIP (De Angeli et al., 2002) is to replace the traditional Personal Identification Number (PIN) approach with images as a means for user authentication in ATMs (Automatic Teller Machine) and similar systems.

VIP can be implemented differently to test various criteria, for instance (De Angeli et al., 2005) conducted user studies on 3 implementations based on the VIP approach. One implementation replaced each number with a pictorial equivalent from a subject-oriented category. For example, 4 is always replaced by any flower picture selected randomly from the flowers category and 9 is replaced with a picture from the animals category as seen in Figure 2.10. To sign-in, users should recall the images in order. Another implementation had a grid of 16 pictures instead of 10 as seen in Figure 2.11.
By testing these implementations and comparing them against a PIN system, data reported in (De Angeli et al., 2005) shows that on their first attempts users were more successful to sign-in with a PIN, but then VIP achieved better results within and after a week. Another study by (Moncur et al., 2007) also concluded that systems like VIP maintain memorability and have better success rates than PIN systems.

Time needed to authenticate a user varied from one implementation to another, for instance users needed less time using a visual system with fewer images. However, it was always more than that of the PIN system. Nevertheless, based on the results the authors argued there is no superiority for using images over PINs.

**Man et al.’s shoulder-surfing resistant scheme**

To explain the idea behind (Man et al., 2003) scheme, assume having the following password: “Chocolate2Dollars30Cents”. The authors suggest that this type of password might look long, but is easy to recall considering that it can be logically divided into 3 parts: “Chocolate”, 2Dollars” and “30Cents”. Complicated strings can be used as well, but should always be logically divided into 3 parts in the user’s mind.

To protect the system against observation attacks, the system is going to ask for a single part of the password at a time in a way only the legitimate owner of the account can understand. To make this possible, a graphical interface is attached to the sign-in page where different icons and objects are displayed. Some of these objects are part of a previously selected portfolio recognised by the relevant user only. If the portfolio objects are gathered on the right hand side of the screen, as seen in Figure 2.12, then the system is asking of the right hand side part of the password, that is: “50 Cents”.

![Figure 2.10: VIP example (De Angeli et al., 2005).](image)

![Figure 2.11: VIP example (De Angeli et al., 2005).](image)
if the objects are divided on both sides of the screen, then the required part is the middle part: “2Dollars”. The final case is if the objects were gathered on the left, then, the correct answer in this example is: “Chocolate”.

![Image of icons](image-url)

**Figure 2.12:** Man's shoulder-surfing resistant scheme (Man et al., 2003). The case where all user icons are on the right hand side of the screen.

The system has been tested with a group of participants and assumed to be working by the authors, but no data regarding success rates, registration and sign-in times were reported. As for security, they mathematically and theoretically discussed that if a shoulder-surfing attack is launched 480 times on a user, the system will prevent the attacker from acquiring the password with a confidence level greater than 0.9.

**Hong et al.’s spyware resistant scheme**

This scheme (Hong et al., 2004) is designed as a challenge response system to confuse spyware. Part of the idea has been proposed by the same team in (Man et al., 2003) to resist shoulder-surfing attacks. However, in Hong’s approach each user selects a unique portfolio of icons and objects, each of these icons is linked to a string. Users are advised to choose a string they can logically link to the object, for example, “Feb14” refers to Valentine’s Day and could be associated to a “Heart” icon.

During the sign-in session, the system displays some icons from the user portfolio among others as seen in Figure 2.13, and users should first recognise these icons correctly and then write the relevant strings (in order) in the password field. This way, the password string for every login session is different.
The implementation of the scheme was described as promising by the authors and that it took them approximately 15 minutes to register and then use the password to sign-in. But no laboratory or field study results have been mentioned to support their usability arguments.

**Story**

While conducting a field study to examine Passfaces, Story was developed as a comparative system by (Davis et al., 2004). In Story, images represent people, places and different objects, as in Figure 2.14. Users are expected to select images logically connected together as a small story to maintain memorability. But in practice, most users selected images randomly.
In the study (154 participants) it was found that patterns exist in Story as well. For example, differences between male and female choices were highlighted based on what they are more attracted to. However, user choices in general were more varied than passfaces.

The system requires that users remember the images in order, which was a problem for many participants, hence, success rate dropped to about 85%.

**Weinshall’s Cognitive Authentication Scheme**

Weinshall proposed a cognitive authentication scheme (Weinshall, 2006) with the objective of fighting eavesdropping over potentially compromised computers and insecure networks such as public connections. The scheme was developed to rely on human perspective and recognition capabilities alone.

During registration, a portfolio is selected. Users should be trained very well to distinguish all the images assigned to them from a bigger set because for the authentication to succeed, users are required to identify some of these images among others to answer a multiple-choice question which can only be answered by someone familiar with the portfolio. The system repeats a number of trials and if the correct answers are gathered to a certain level, user login is accepted. This implies that the system has a degree of tolerance as it does not reject users due to a single mistake.

The sign-in task requires users to mentally compute a path usually starting from the top-left image in the panel in such a way that users should move down if they stand on a picture that belongs to their portfolio, or else move right until they reach a number on the right-most end of the grid as seen in Figure 2.15. That number is then reported as the login answer using a keyboard to reduce shoulder-surfing attacks.

The main drawback of the system is the amount of time used to authenticate users, for instance, 3 minutes were required while using the high complexity approach of the system and 1.5 minutes while using a simplified version. Training users is time consuming too, the author reported three day training for participants until the system achieved a success rate over 95%. Long training was suggested in this scheme to replace the user-choice approach with system forced images to avoid patterns.
The scheme's was then successfully attacked by (Golle and Wagner, 2007) and it was found that the password can be revealed in a few seconds after watching a small number of successful sign-in attempts, hence the system is not secure against eavesdropping (shoulder-surfing).

**Convex Hull Click (CHC)**

CHC (Wiedenbeck et al., 2006) is a visual authentication approach which has been developed to resist shoulder-surfing attacks by both human observation and electronic recording devices.

During registration, portfolios are assigned to user accounts, each portfolio consists of small images (icons). For users to complete the authentication task, they should recognise at least 3 correct icons and click anywhere within the area between them (within the convex hull of the icons) as seen in Figure 2.16. This way, users can prove knowledge of the available portfolio icons without clicking, highlighting or pointing at them directly.

![Figure 2.15: Weinshall’s Cognitive Authentication Scheme Query Panel (Weinshall, 2006).](image-url)
The sign-in process includes multiple rounds, so the icons are redistributed again with every challenge. CHC authentication time is longer than many other VP schemes because users need to properly scan the window for pass-icons. In addition, rearranging the icons for the next challenge takes about 3.4 seconds, which adds 17 seconds per password in total to the sign-in process time (Wiedenbeck et al., 2006). Hence, the system resists shoulder-surfing attacks at the cost of longer authentication time.

![Figure 2.16](image-url)

**Figure 2.16:** CHC, Illustration of the invisible convex hull with 3 pass-icons (Wiedenbeck et al., 2006)

The authors suggested increasing the number of pass-icons to increase the theoretical password space, but unfortunately, the system has already been proven to be time-consuming and increasing the number of challenges or pass-icons will cause usability issues.

**DynaHand**

Dynahand (Renaud and Olsen, 2007) requires users to identify their drawings or handwriting in order to sign-in. During registration, a range of handwriting samples are submitted to the system and then to authenticate, users should be able to distinguish their own samples from other images displayed by the system as seen in Figure 2.17.
The system resists shoulder-surfing attacks because it does not rely on the information itself (the number) as seen in the example above; instead, it depends on the way it has been drawn. This also implies that if an attacker is able to recognise a unique drawing pattern in the selected number, the attack might succeed. In addition, it might be possible to attack the system by predicting the password using analysis techniques after collecting handwriting samples belonging to the targeted user.

Dynahand’s user study shows 97.4% success rate with an average login time of 28.1 seconds (for 3 rounds). Registration is time consuming as users are required to create and submit unique and different handwritten samples. Considering the system has been proposed with 3 rounds, the number of possible guesses is 729, that is 1 out of 9 three times (Renaud and Olsen, 2007).

**Use Your Illusion**

This scheme relies on the human ability to recognise distorted versions of a previously seen image (Hayashi et al., 2008). Figure 2.18 shows an example of what a user can see during registration. The original image helps users to easily recognize the distorted version during future authentication.

Converting images this way serves the purpose of preventing social engineering and shoulder-surfing attacks. The authors suggest that it is difficult to mentally revert a deformed image without previous knowledge of the original, which implies that legitimate users could easily recognise the correct portfolio, although it will be hard to describe such images to other people.
Figure 2.18: User Your Illusion - Image portfolio assignment. The original images are shown along with their distorted versions (Hayashi et al., 2008)

Within the same study, web-based and cell phone prototypes were developed to test the system with a total number of 99 participants and the result shows that (from a usability perspective) using distorted images has an equivalent error rate to traditional images, but only when the original images are known to the user at the time of registration. After 4 weeks of registration, users spent 17.9 seconds on average to authenticate, time increased to 24.7 seconds when the portfolio images were imposed on them during registration. Authentication success rates after a week are as high as 100% with self-selected portfolios and 73% for users with randomly imposed portfolios.

2.5.5. Category-based evaluation

References and debates from the VP survey presented in the last section are used to discuss and evaluate the main three categories (pure recall, cued recall and recognition) based on the main usability and security aspects.

Memory: The challenge in visual pure recall schemes is similar to that of text passwords as both methods require users to depend on their minds only to recall the shared secret. Cued recall is an easier challenge since the system is designed to trigger users’ memory to enter the correct answer, whilst research work such as (Dhamija and Perrig, 2000) and many others quoted in section 2.5 clearly acknowledge recognition-based (imprecise recall) in particular to be easier to remember for the long term than other type of memory tasks. Nevertheless, recognising pictures is easier than recognising written materials (Shepard, 1967). Recognition is less confusing than cued recall because the user is asked to distinguish the image as a whole, while in cued recall, details inside the image should be remembered correctly to locate the right click-points in order.
**Time:** Users in the Détéà Vu study took about 45 seconds to create the VP and 32-36 seconds to sign-in (Dhamija and Perrig, 2000). Time increased to 3-5 minutes (for 5 rounds) to register a new user in the commercial scheme of Passfaces as reported by Professor Tim Valentine. Weinshall’s scheme user-study reported 1.5 to 3 minutes used to sign-in to the system only (Weinshall, 2006). Registration and sign-in time is relatively long in recognition-based schemes due to the number of rounds forced by the system to maintain a reasonable password space. Cued recall (click-based) schemes took less time, Passpoints for example reported 64 seconds (Wiedenbeck et al., 2005c, Wiedenbeck et al., 2005a); while CCP users took 20 seconds to register and less than 10 seconds to sign-in (Chiasson et al., 2007). Pure recall schemes are hard to evaluate because no time has been reported by the available case studies. However, since they do not require multiple rounds as seen in recognition-based solutions, a well designed scheme should be expected to require time close to this needed by click-based systems.

**Success rates:** The results are found to be symmetrical with the level of memory challenge. For instance, recognition solutions, as the easiest to recall among other categories, are found to have high success rates reaching 100% after a weeks time as in Use Your Illusion and over 95% as in Weinshall’s scheme and Passfaces. However, the implementation of the system is important and could affect success rates either in a positive or a negative way. A perfect example of this is how the success rates decreased to 73% after imposing random portfolios on end users instead of relying on a user choice approach (Hayashi et al., 2008). Cued recall results are mostly available from user studies of click-based systems. Rates of successful sign-in attempts vary between 55% and 90% in Passpoints and 96% in CCP. Not enough data has been reported in the case studies of pure recall schemes, except for Pass-Go, which has a login success rate of 78%.

**Theoretical password space:** The bigger it is, the harder to perform a successful brute force attack against the system. The theoretical password space in pure recall and cued recall schemes can be increased by enlarging the background grid size (or the employed image as in click-based systems). Other parameters exist such as: supporting multi colour choice to draw more complicated passwords; decreasing the accepted tolerance distance; or using longer passwords. However, some schemes are different, for instance 3D Password has a very large password space due to its complicated 3D environment as explained earlier. On the other side, recognition-based systems can only support a relatively small password space due to the usability limitation of implementing a recognition scheme with a large number of rounds and recognition grid pictures. In addition, the size of the user’s portfolio should be limited to avoid memorability problems.
**Effective password space:** The bigger it is, the harder to perform a successful dictionary attack. A dictionary attack is a fast automated guessing attack searching the password within the effective password space. A vulnerable password can have a predictable symmetry (Nali and Thorpe, 2004) in pure recall, Attractive images in recognition (Davis et al., 2004) or clicks within hotspots in click-based systems. Hotspots are discussed in detail in Chapter 6. Therefore, all schemes can be vulnerable to having a small effective password space.

**Human guessing:** An attacker with the essential knowledge has a good chance to guess the password if it is strongly based on personal preferences. In recognition-based systems for example, a Passfaces study showed people to be more attracted to select faces of their ethnicity or selecting beautiful women over others (Davis et al., 2004). The same scenario can be true in the case of objects (e.g. selecting a dog over other animals) or colours and shapes (e.g. selecting the abstract image with the most red colour). For the other two categories and due to their bigger password space, the chance for a human guessing attack to succeed is less. While this was possible with an automated dictionary attack, it is more difficult to predict someone’s drawing on a grid or clicking the correct set of objects inside different images in the same order by a person. This might vary from one scheme to another, but the general perception is as discussed.

**Shoulder-surfing:** Good observation of a single login has a reasonable chance to succeed in revealing the password in the vast majority of schemes discussed so far regardless of their category (type of memory). Systems resistant to shoulder-surfing or hidden cameras are those requiring part of the password at a time to authenticate, hence revealing the full password forces attackers to repeat a successful attack many times. However, some schemes such as Déjà Vu have a type of VPs hard to remember by a human from the first few login attempts in comparison with other systems employing normal pictures (or objects). In conclusion, there is no evidence that one VP category is more resistant to shoulder-surfing than others.

**Offline attack (database):** Pure recall and cued recall VPs (signatures, drawings, doodles, click-points etc) can be translated into text, which can then be hashed in the database. To reveal the password, either a brute force or dictionary attack should first succeed. Whilst, in current recognition-based systems, the user portfolio must be available to the system at all times, which implies that accessing the backend leads to obtaining the password immediately as it will be written in plain text (or in reversible encryption).
Social engineering: Despite the scheme category, if the VP is well selected it can be difficult or sometimes impossible to share in an oral manner. Imagine describing an abstract image, a click-point on a particular apple in a picture full of apple trees or a complicated signature. However, that is still possible by taking snapshots and drawings sketches to guide other people toward the correct password. Nevertheless, sharing such passwords in an oral manner can be possible in some cases, e.g. selecting or clicking unique items in a picture, or drawing a popular symmetry.

Phishing: Assuming that a user trusted a phishing site and started entering his password, in case of pure recall systems, the attacker can mock a clone of the interface and perform a bulk phishing attack to capture passwords without further interaction with the original system. With cued recall, the attacker should first retrieve the right portfolio for each targeted user, otherwise the phishing site must be designed to perform a MiTM attack between the original system and the user to request the relevant images correctly and then capture the password. Recognition-based schemes require part of the password at a time hence it takes multiple logins to reveal the password. However, a successful MiTM attack could lead to a successful login attempt at the time of the attack.

Malware: If the computer is infected with a malware, pure recall VPs are possibly captured by either a single snapshot of the interface or by recording the mouse motion during a sign-in session. For a cued recall system, several screen shots are required to record the portfolio in addition to the mouse to record click-points. Recognition-based schemes require multiple records of logins to reveal the password in which screen shots are gathered in addition to mouse clicks and keystrokes if the scheme supports the keyboard to select images.

Singularity: Unlike pure recall schemes, passwords in cued and recognition-based solutions are connected to the images employed. Using unique images means there is a good chance the generated passwords cannot be reused on different systems by the same users.

In conclusion, no category can be selected to resist all type of attacks while maintaining usability better than others. However, cued-recall is found to serve this research because it has the potential to resist more security threats as discussed earlier while maintaining usability. The main drawback of pure-recall is memorability; it has no cues to help end users while in this research we argue that memorability is the main problem behind user insecure practises. The main drawback of recognition-based solutions is their lack of
resistance to guessing and offline attacks in addition to being slower as they require more authentication time.

2.5.6. More insight into click-based systems

Passpoints and CCP are ideal examples of click-based systems. Their user studies show they maintain large password spaces, passwords are encrypted in the database and sign-in time is much less than that of recognition-based schemes. In addition, they provide cues for users to recall their passwords correctly.

Research studies specify recognition-based systems to be easier to remember nevertheless this conclusion can be challenged in practice since the password space of the selected password is much less than that of cued recall systems. This research found development opportunities for click-based systems to enhance their performance and resistance to many security threats such as phishing and dictionary attacks.

As explained earlier, the VP in click-based systems is a sequence of clicks on a picture where every pixel is a possible click-point, but since relocating these particular pixel-based click-points is hard a serious usability issue was raised. The solution is to accept clicks within an area of tolerance around the original click-point, which results in multiple possible inputs instead of a single value. This implies that producing a single hash for each password is not possible unless discretization is used. This matter is covered in detail with further development in Chapter 5.

The password space in click-based systems is large but can be affected by hotspots. These are click-points contained within size-limited areas on an image, which users are attracted to if their choice was not disturbed. Hotspots can be exploited to perform dictionary attacks against the system. To maintain randomness and increase the effective space, click-based systems should be developed with techniques to help users select random passwords while maintaining memorability. Chapter 6 discusses this problem then presents a method to increase the effective password space in click-based VPs.

Another challenge is that the password entry relies entirely on the mouse or a similar device, if these are not available, it will be impossible to authenticate using the keyboard alone unless a proper HCI technique is developed.
2.6. Other proposals

2.6.1. Audio passwords

Long time after proposing a variety of VP schemes, some research papers such as (Chiasson et al., 2008a) (Conrad et al., 2006) mentioned the possibility of using music for authentication. This idea was then implemented by (Gibson et al., 2009). The prototype is called Musipass because it uses music as passwords. Similar in principle to the VP recognition-based schemes, during enrolment it asks users to go through 4 rounds, in each they are asked to select a music clip of their own choice from a grid of 9 clips to create their portfolio. To authenticate, users are asked to recognise their music clips among others.

Audio passwords are not considered in this research due to major usability and security flaws. For instance, similar to recognition-based VPs their password space is too small to resist guessing and offline attacks. In addition, the authentication time is significantly long. As such, they are slow even if compared to recognition-based VPs because selecting a Musipass requires playing and listening to the file. Nevertheless, they require headsets to listen to the passwords in private. This is a serious security concern since users with speakers but not headsets might choose to authenticate and expose their password to all people in the nearby.

2.6.2. Behavioural biometrics

The third category of authentication factors (something the user is or does) includes keystroke dynamics and mouse dynamics which are behavioural characteristics that does not require special external device(s) to input their data to the computer, which is why they fall within the scope of this thesis.

Keystroke dynamics identify users based on the analysis of their typing metrics (Dowland et al., 2002). This can be achieved through:

- Their use of special keys: the habit of using specific additional keys on the keyboard and the order of pressing them
- Periodic dynamic: analysis of keystrokes for a specific session or amount of time.

Mouse dynamics is based on the way users move their mouse (Ahmed and Traore, 2007). It depends on extracting the behavioural features related to mouse movements.
Statistical techniques are then used to compute and analyse the movement characteristics.

In conclusion, keyboard and mouse are input devices, which can be used differently by different people (Joyce and Gupta, 1990). Thus, they can be employed to provide an additional level of authentication and unlike other systems; they can be used to track users continuously during a live session to verify identity. This implies that the system will be able to recognise if the account is still being used by the original user who initially signed in or is replaced by someone else. However, this technique cannot be used at the beginning of the session (e.g. sign-in page) since the process of data capturing and analysis requires a relatively long time.

2.7. HCI-Sec

Developing an authentication scheme with the highest levels of security is good, but it will not succeed without maintaining usability. Vice versa, developing a usable scheme that is not secure will eventually be hacked and become unused. While security and usability conflict with each other, HCI-Sec is the research area of investigating and developing new design technologies to produce a system that can maintain both security and usability.

Before HCI-Sec was recognised as a discrete field, researchers were aware of the problem, but used to outline security or usability issues separately. For example, while studying computer break-ins in 1987, Reid argued that Unix was created for researchers and not as a commercial operating system, hence it should be less usable and more secure (Reid, 1990). The topic has long been pointed out, Saltzer and Schroeder in 1975 introduced the ‘psychological acceptability’ factor and argued that it is fundamental for secure system developers to design an easy-to-use system, so that users can be expected to follow the security mechanisms correctly and consistently (Saltzer and Schroeder, 1975). However, until recently research on security has tended to ignore performing proper usability studies. (Adams and Sasse, 1999) note that many authentication schemes were proposed to control access to systems, but very few studies exists to investigate the usability of these proposals. The VP survey have proves that usability has rarely been reliably tested in a laboratory or field studies for these schemes. Nevertheless, in his review of the history of security and usability in the literature (Garfinkel, 2005) concludes that ‘while many security researchers have long considered usability issues, and usability researchers have long considered security issues, the topic has only rarely received significant attention as a subject of primary study’.
From the above, HCI-Sec might not be a new field, but this research area started to receive more attention during the 1990’s starting with researchers such as (Karat, 1989) who did a usability testing of a security applications, then more publications followed such as (Whitten and Tygar, 1999) and (Zurko and Simon, 1996) who introduced ‘user-centred security’.

HCI alone has various rules and principles to be followed in order to enhance the usability of the system; examples to these are:

- Supporting feedback: the system should keep the user alert to the task being performed.
- Availability of shortcuts: to aid advanced users to perform certain actions fast.
- Providing proper help tips and documentations.
- Using clear and simple dialogue for error messages and alerts.
- Speed: Users should be able to perform the required tasks in the minimum amount of time possible. Also system response time should be fast.
- Maintain user's memory: users should find it easy to continue using the system without relying much on their memory to recall instructions on how to do things.

These principles are general to all computer systems, while others are more specific to serve certain type of software such as security software. To examine an authentication system for instance, it is important to perform user studies in which participants are aware of the security nature of the system they are asked to test to observe their behaviour and analyse authentication time, training time and success rates.

In addition, HCI-Sec includes essential security properties that must be considered while designing the system which makes it different from other software. (Whitten and Tygar, 1999) developed a list of these properties as summarised below:

- Security features must not be too complicated, because for end users it is a secondary goal at its best. If they find it too difficult to perform or time consuming, it will be ignored.
- In security systems, it is difficult to provide good and full feedbacks regarding the system all the time since the technicality involved in the design can be too complex to summarize. However, reasonable feedback alerts should exist.
- Attached security policies and guidelines are usually abstract and can be well understood by professionals, but not normal end users. If the targeted audience is end users, technical jargon should be avoided or explained.

- Once a secret is revealed on one occasion, the relevant damage cannot be reversed. Even worse, there is no usually method to verify that protected data has been compromised or not.

- The security of the system is measured by its weakest link. That is because exploiting a single error can open further doors to crack the system.

These properties have been criticized of being applicable to many software rather than secure systems in particular (Garfinkel, 2005). However, while this can be the case, it must be much more critical to security software (e.g. an authentication mechanism) to sustain such properties, failing to maintain HCI-Sec in security software causes more damage than other software. For examples, a security breach to change the content of a showcase web application causes less damage than that of a web authentication mechanism protecting confidential data for a large number of users.

2.8. Terminology

In this section, the core terminology used throughout the thesis is defined.

Text password: is a password consisting of letters, digits and symbols. It is used to distinguish this kind of password from visual passwords. Other previously known terms were passed over because they might not give the same meaning correctly. For example, “alphanumeric password” could imply that symbols are excluded.

Visual Passwords (VPs): A visual password consists of a number of images or clicks performed on these images. To authenticate, a user must perform an action in a VP authentication scheme. These actions can be: selecting images to create a login portfolio, clicking objects or click-points inside images or drawing a doodle. The term ‘graphical password’ is used as well by some researchers in the literature and refers to the same meaning.

Click-based systems: VP authentication schemes in which the VP is a sequence of click-points on one image or more.

Click-Cell: In click-based system, a click-cell is all the area of tolerance surrounding and including a click-point performed by a user.
Continuous monitoring: a potential security enhancement technique for authentication systems. It is the process of tracking users’ behaviour during a logged-in session to verify identity. If a user leaves his/her email account open and another person tries to use it, the system will log the account out automatically and activate the initial authentication method again. Continuous monitoring is presented in this thesis because it can be used as a complementary method in online authentication schemes.

Key logging: also called keystroke logging is the action of monitoring and recording what a user is typing on a keyboard.

Mnemonic phrase-based password: is a password in which the user chooses an easy to remember phrase then selects a letter from each word to form a secret string.

CAPTCHA: Is a challenge-response test used in some computer applications to distinguish humans from bots and scripts.

LTM: Abbreviation of Long-Term Memory, while human short term memory stores information for a few seconds, the LTM can recall information longer (days or as long as decades).
Chapter 3  Survey of User Perceptions

Online service providers require a method to identify users on the web. This is critical to force different levels of access rights on individuals. Legitimate members must gain access with the appropriate privileges smoothly whilst malicious login attempts must be banned from the system.

Text passwords are used for authentication and end users are familiar with them because they have been employed in computers since 1960 when Corbato designed the first access control solution for time sharing systems based on passwords and usernames login (Smith, 2001). However, these are not the only ways to electronically verify that people are who they say they are; various methods have been suggested and developed based on different factors. This chapter introduces a survey of user perceptions to measure and analyze the acceptance of different VP categories. The concept of continuous monitoring is also presented as a new security enhancement technique.

The survey proposed solutions within the scope of this thesis. VPs for example do not require any additional hardware text passwords do not use. As such, a successful candidate to replace text passwords must be cost-effective and suitable to cover a wide area of application. In addition, it is critical for any proposed scheme to maintain usability by enhancing user memorability through simplifying password entry and providing cues to trigger user LTM. In theory, a method with these characteristics should gain user acceptance and be described as a user-friendly solution.

Up-to-date information about users’ awareness of, and attitudes towards, text password security practices is also presented and discussed because the password problem covered in Chapter 2 remains an issue despite all policies and guidelines being taught to end users. For example, the survey studies the demand for more password-protected accounts in people’s lives which encourage insecure password practices. Further, a question to investigate is whether people’s demographic information such as age, major or being a student affects the number of these accounts. Studying these practices with up-to-date information helps research in this area with evidence.
3.1. The survey

The survey aims to assess the following issues to achieve the first objective of this thesis:

- Public awareness of, and attitudes toward, password security and usability.
- Public attitudes toward different forms of initial user authentication for web applications.
- Public attitudes toward the concept of continuous monitoring.

The assumptions before performing the survey were:

- System administrators and security consultants encourage policies and guidelines such as the ones presented in Section 2.2.2 to resist insecure behaviours and control the text password problem. However, a minority of users succeed in following all these guidelines while the majority fail.
- The number of frequently used password-protected systems is continually increasing.

There are 46 questions in this survey presented in five groups, many of them were presented as multiple-choice and the rest employed a Likert scale. Some questions were answered using a comment box.

Responses were date stamped and IP Addresses were logged because participants were required to complete the survey online (Appendix A includes snapshots for the survey) and this helped to put restrictions on how many times a survey can be accepted from a specified party. Cookies were also used for access control so that participants save partially-finished surveys and finish them at a later time.

Group one questions gathered general details to provide demographic information such as age, gender, education and place of residence. These were collected to study their effect on some results such as the number of accounts and to describe the sample of people who participated in this survey.

Whilst group two was more about measuring the level of computer usage in general as well as some selected technologies such as Single Sign-On (SSO).
Password related questions were covered in group three to understand how people deal with security measures currently, their awareness of possible methods of breaches to confidential accounts and usage of any alternative authentication mechanisms.

After that, the participants were introduced to possible alternative techniques for initial login with group four. The questions were supported with help tips and images to explain what and how each technique works (e.g. explanation of what recognition-based graphical password solutions are, how they work and then supporting that text with an image or snapshot) followed by a five point Likert scale to let them rate the acceptability of the technique compared to others where 1 is “Very Unacceptable” and 5 is “Very Acceptable”.

Finally, group five introduced the participants to Continuous Monitoring to test the acceptability of the idea initially, the impact of its possible methods on them and finally some other important questions regarding the frequency of false rejection, trust and privacy issues.

The survey was hosted online for four months on a server and the link was distributed to individuals and groups in different countries. To reach many people the survey’s link was sent to members of existing university groups. It was also sent to mailing lists, online forums and social networking groups such as Facebook. This resulted in an overall number of 204 responses where 157 responses were completed and submitted successfully while 47 responses were excluded because they were not completely filled out by the day the survey questionnaire was taken offline.

3.2. Analysis of results

3.2.1. General

Figure 3.1 illustrates the percentage of responses from different age groups. These are used to compare with other result from the survey. For example, Figure 3.14 illustrates the number of accounts for each age group.
56.05% of the overall responses were completed by females. Computing, engineering and business were selected as the major backgrounds for the majority (61%) of the participants and 62% of them claimed to be students or enrolled in educational courses. Furthermore, 45% completed or still enrolled in a postgraduate study.

Figure 3.2 shows that the highest number of participants came from Asia and Europe, and then Africa and North America.

3.2.2. Level of computer usage

In terms of computer usage, the vast majority of participants have a fair to very good experience with computers and IT with 99.36% of them having and using a computer either at home or at work, and 96.18% having internet service.

In addition, 91.72% claimed to have used computers for more than 5 years and 79.62% have used the internet for the same period of time.
Due to the increased numbers of systems with access control, a question asked about the number of systems frequently used by participants during their daily life, whether it is a bank account, email or a website as long as it requires text passwords for authentication. The results showed that 47.77% use passwords for more than 10 systems and 8.28% use them for over 30 systems. Figure 3.3 illustrates the results in further detail.

![Figure 3.3: Number of accounts.](image)

One of the proposed solutions to help users who deal with many user accounts is Single Sign-On (SSO). It enables users to log in with a single password to their different email accounts, websites and windows applications without being prompted to enter each system’s password individually. The SSO software does that by saving all user passwords in a secure place (encrypted database). These passwords are then used to open related accounts after the user enters the software master password.

The participants were asked if they use any SSO solution at all and 20.38% replied with a yes.

### 3.2.3. Text password related questions

Public awareness of text passwords security and usability has been examined because it is still adopted by the vast majority of systems for authentication purposes (DTI, 2006).

Our survey showed that only 39.49% of the participants never shared their own passwords, as for those who shared it 16.56% of them shared their work passwords with other people.
74.52% said that friends and colleagues have given them their passwords and an interesting percentage of 22.94% claimed to have or used other people’s passwords without their knowledge or permission.

At work, 68.37% of whom have jobs (117 participants) said the system at work enforces a minimum length for passwords and 41.52% claimed it shows them a scale to measure how secure the password they create is.

The results also showed that 56.69% of the users do not change their passwords if the system does not force them to do that, even if the account is considered to be important or critical some way or other, and 21.02% are those who are always careful to use a combination of random letters (a, b, c, ...), numbers (1, 2, 3...) and symbols (?, !, @...) together to create a password. Nevertheless, 20.38% sometimes feel their passwords can be easily guessed.

Figure 3.4: How often passwords of critical accounts are changed

Figure 3.5: How often participants avoid simple words available in dictionaries.
Figure 3.6: How often a combination of random letters (a,b, c ...), numbers (1, 2, 3...) and symbols (?!, @...) are used together to create a password.

Figure 3.7: How often a password is written down after creating a new account.

Figure 3.8: How often the participants check their new password to be at least 6 characters long when creating a new account.
An expectable result also agreed with other studies such as (DTI, 2006) by showing that 89.81\% of the participants use text passwords to access all of their accounts, leaving 10.19\% only of those who happened to use finger-prints and security tokens to access one or more of their user accounts.

When another question focused more on the work environment, 95.08\% claimed to be using text passwords only.

As for what text passwords consist of, Figure 3.9 illustrates the details and shows that 47\% to 52\% of the participants use only letters and numbers to access their accounts at work and something similar applies to their internet banking accounts.

The number of participants who claimed to have internet banking accounts is 100 where 92 depends on text passwords and the rest use tokens.

![Figure 3.9: Comparison between participants’ passwords at work and the ones they use for internet banking.](image)

The final text password related question asked the participants how often they use the same password for more than one account, 31.21\% of them stated that they Frequently do that and a similar percentage of 31.85\% selected Sometimes. 8.92\% thought they always reuse the same password. However, 28.03\% responded saying they rarely or never reuse their passwords.
3.2.4. Alternative techniques for initial login

To study public attitudes and perceptions towards different forms of initial user authentication for web applications a Likert scale was provided along with explanations, tips and images to explain how each technique works. Table 3.1 displays the results and the numbers 1 to 5 represent the Likert scale. Under each number is the percentage of people who selected it.

**Table 3.1**: Results of public acceptability of different initial login techniques, where 1 is very unacceptable and 5 is very acceptable.

<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Password</td>
<td>4.46%</td>
<td>3.82%</td>
<td>23.57%</td>
<td>27.39%</td>
<td>40.76%</td>
</tr>
<tr>
<td>Question and Answer</td>
<td>13.38%</td>
<td>17.20%</td>
<td>33.76%</td>
<td>20.38%</td>
<td>15.29%</td>
</tr>
<tr>
<td>Recognition based Graphical Password</td>
<td>22.93%</td>
<td>26.11%</td>
<td>26.75%</td>
<td>15.92%</td>
<td>8.28%</td>
</tr>
<tr>
<td>Cued Recall based Graphical Password</td>
<td>28.03%</td>
<td>21.66%</td>
<td>26.75%</td>
<td>16.56%</td>
<td>7.01%</td>
</tr>
<tr>
<td>Pure Recall based Graphical Password</td>
<td>29.30%</td>
<td>24.84%</td>
<td>26.75%</td>
<td>12.10%</td>
<td>7.01%</td>
</tr>
</tbody>
</table>

Figure 3.10 illustrates the data of the previous table in a comparative way. It shows that users are more confident to rely on text passwords whilst many users selected 3 for Question and Answer to say: they are doubtful. Consequently, the rest of the mechanisms gave us almost similar results.
Figure 3.10: Comparison of public acceptability of different initial login techniques, where 1 is very unacceptable and 5 is very acceptable.

3.2.5. Continuous monitoring

Continuous monitoring was finally introduced as the process of tracking user behaviour during a logged-in session to verify identity. This was simplified to the following: if a user leaves his/her email account open and another person tries to use it, the system will log the account out automatically and activate the initial authentication method again.

The survey shows that 44.59% were positive in accepting this feature while 28.03% rejected it and the remaining 27.39% do not have a clear opinion yet.
Mouse and keystroke dynamics and their possible usage have been introduced in brief detail. Figure 3.11 shows that public acceptability to both of them gives similar results, where 1 is very unacceptable and 5 is very acceptable.

![Comparison between the public acceptability to mouse and keystroke dynamics to be used for continuous monitoring.](image)

**Figure 3.11**: Comparison between the public acceptability to mouse and keystroke dynamics to be used for continuous monitoring.

False rejection is one of the major possible drawbacks of any decision making system. The participants were asked to decide the acceptable frequency of false rejection to them; Figure 3.12 illustrates the results.
Mouse and Keystroke Dynamics are behavioural characteristics of the users, therefore, unlike normal passwords they do not change easily. Participants were asked if they trust their organization or school to collect their behavioural data in order to provide this security feature and 60.51% decided they do not trust it. Furthermore, 63.06% consider it to be an invasion of their privacy.

If continuous monitoring techniques are not used, some web applications automatically log the users out after a certain time of inactivity and force them to log in again for security reasons. The continuous monitoring section was closed by asking about the acceptable frequency of automatically logging a user account out after a period of inactivity. The results are in Figure 3.13.

Figure 3.12: Acceptable frequency of false rejection.

<table>
<thead>
<tr>
<th>Number of Hours</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Hour or less</td>
<td>30.57%</td>
</tr>
<tr>
<td>1-3 Hours</td>
<td>19.11%</td>
</tr>
<tr>
<td>3-9 Hours</td>
<td>12.74%</td>
</tr>
<tr>
<td>9-24 Hours</td>
<td>7.64%</td>
</tr>
<tr>
<td>1-3 Days</td>
<td>7.64%</td>
</tr>
<tr>
<td>3-7 Days</td>
<td>3.18%</td>
</tr>
<tr>
<td>More than 7 Days</td>
<td>19.11%</td>
</tr>
</tbody>
</table>

Figure 3.13: Acceptable frequency of automatically logging a user account out after period of inactivity.

<table>
<thead>
<tr>
<th>Number of Days</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Hour or less</td>
<td>39.49%</td>
</tr>
<tr>
<td>1-3 Hours</td>
<td>28.03%</td>
</tr>
<tr>
<td>3-9 Hours</td>
<td>7.01%</td>
</tr>
<tr>
<td>9-24 Hours</td>
<td>9.55%</td>
</tr>
<tr>
<td>1-3 Days</td>
<td>3.82%</td>
</tr>
<tr>
<td>3-7 Days</td>
<td>3.18%</td>
</tr>
<tr>
<td>More than 7 Days</td>
<td>8.92%</td>
</tr>
</tbody>
</table>
3.3. Discussion

Figure 3.2 shows that most of the participants are residents in Asia, which could be a result of translating our survey questionnaire to Arabic in order to distribute it more easily in the Middle East and North Africa. The reason behind choosing Arabic only as a second language was the author's ability to use it effectively, thus, making sure the way the questionnaire is presented in Arabic is identical to the English version in a way that does not affect the results.

The number of access controlled systems is increasing rapidly in people's daily life as more than ever technology plays a key role. If the results from Figure 3.3 are compared with a similar study (Furnell et al., 2000) we notice how the percentage of people using more than 10 systems increased from approximately 13% nine years ago to 47.77% in this survey, though the previous study also argued to have most responses from people with technology fields such as computing, telecommunications and engineering who also happened to have familiarity with IT. This result could explain why 56.69% of the respondents do not change their accounts passwords unless the system forces them. With more passwords to be created, it is more difficult to adopt good security practices. Obviously, people are trying to keep track of their login credentials, so we end up with no more than 21.02% of them who always give extra attention to use good combinations of digits, letters and symbols while creating passwords.

To check whether our previous results apply to people regardless of their demographic information such as their age, place of residence or whether they are students or not, results are further analysed. Figure 3.14 illustrates the number of responses for every age group, this shows how having 5 to 10 accounts at a time is consistence for all ages.
Figure 3.14: Comparison between the four age groups in the survey regarding the number of accounts they deal with.

Student seems to indeed need more than 5 accounts, see Figure 3.15. However the graph shows a good percentage of each group using 15 accounts at a time or even over 30 accounts.
Most of the responses came from Asia and Europe, therefore we were able to do the comparison in Figure 3.16 between these two places. There were not enough responses from other continents to involve them as well. Again the diagram lines show that despite the location 5-10 accounts is consistence among users and many people use 10, 15 or even 30 accounts at a time.
The comparison between different initial login techniques showed that people almost had the same response in regards to the different graphical password approaches; this is probably because they did not use them in practice to make a clear decision.

In comparison with text passwords, the results predict how difficult it is for a solution integrated with graphical passwords ideas to be accepted without sufficient proof of its security and usability to users.

The 27.39% who preferred not to give a clear opinion on whether to accept or reject continuous monitoring along with the 28.03% who rejected it should be considered by software developers to keep this feature as an option rather than forcing it in any online service implementation, unless the system belongs to a company where security is favoured over end users opinion. However, there is still a majority of 44.59% who think continuous monitoring is a good idea.

3.4. Conclusion

The analysis has shown the survey’s two assumptions to be valid to a great extent. People are dealing with an increased number of access controlled accounts despite of their age, place of residence or whether they are students or not, and that, text passwords remain the most adopted technique to authenticate end users. In addition, only a limited portion of end users pay attention to adopt all secure practices as seen in the results of the text password related questions, for instance, only 5.10% change the passwords of their critical accounts on a monthly basis; only 21% always use a combination of letters, numbers and symbols; and sharing passwords is a significant phenomenon with 74.54% of them being given a password of their friends or colleagues.

Many text password related issues have been demonstrated for reference as an evidence for users’ continuous insecure practices especially with the increased number of password protected accounts.

In regards to finding a VP replacement to text passwords, the survey showed this option to be unacceptable to end users. However, these participants did not use a VP before hence their perception could change if a VP system proves itself good enough to be user-friendly and secure at the same time compared to text passwords. This is understandable and expectable, as due to inertia people resist change. For the targeted market such as banks more attention is paid to the security aspects of the systems hence they can provide training for users to learn the procedures of the new security
system. This is not a new requirement by our scheme since banks and other companies are already providing security training to their users. Gradually, more people will adapt and it is expected that general public develop better understanding and acceptance of the new system. Nevertheless, no specific VP category was proved to be more favourable by end users in this survey, possibly because of the lack of their field experience with such systems. Hence, usability and then security evaluations of the proposed approaches is the best path to develop an acceptable authentication solution.

The survey also showed that continuous monitoring was not significantly accepted since around half of the users either rejected it or did not post their opinion. This implies that software developers must keep this feature as an optional. Further, end users did not favour mouse or keystroke dynamics among each other. Apparently, what matters is the solution that proves itself to be more accurate and efficient in practice.
Chapter 4  HybridPass – A New VP Scheme

Internet banking schemes and other secure services are designed with two pure recall secrets (passwords) for user authentication. For instance, this can be two text passwords or a text password and memorable information; more examples are discussed and analysed in Chapter 7. The main password in such systems is hashed in the database to resist database attacks so it has to be entered in full whilst the secondary password is partially inquired by the system to resist other attacks such as phishing, shoulder-surfing and network-sniffing. This chapter introduces a new scheme called HybridPass for online user authentication. Its design employs two types of memory challenges into one system. As such, HybridPass combines a text password (pure recall approach) with a click-based VP (cued recall approach) aiming to deliver a new design to enhance online security through achieving bigger password spaces, resisting phishing attacks and maintaining memorability via a cued recall approach.

The main advantage of both VPs and text passwords is that unlike biometrics and tokens they do not require special hardware to work on the client’s side. That is a critical usability privilege which is a key role in gaining the trust of the end users (Cranor and Garfinkel, 2005). However, if implemented alone, they still fail to provide a satisfying solution to the usable security problems of today’s authentication systems. Both text and VPs have limitations. For example, VPs are vulnerable to shoulder-surfing attacks since a computer screen is more exposed to individuals in a shared room than a keyboard. An end user must ensure that no one is observing the screen during authentication to keep the VP secure. Another limitation is the comparatively large amount of time required to register and sign-in (Suo et al., 2005). This is in particular an issue with recognition-based approaches such as Déjà Vu and PassFaces (Real User Corporation, 2001). This scheme shows how these limitations can be minimised by combining the two methods together to provide an integrated login mechanism suitable for web applications. The design is user-friendly and response to a variety of security challenges. For instance, its password is harder to share, cannot be used on other systems, the password space is large and due to the hybrid nature of the approach, it enforces resistance against phishing scams. We argue that using a hybrid scheme combining a pure recall password with a cued recall VP can achieve better security and usability than the schemes requiring two pure recall passwords adopted by many service providers such as banks.
The VP design-in HybridPass has also been designed to be flexible, hence provide a factual proof that a click-based system can conveniently be included or embedded in a content web page similar to the forms currently used for text passwords. This is not possible with other click-based implementations due to a variety of reasons such as their large sizes and the number of login screens to capture their VP. For instance, the images from Passpoints and CCP occupy an area of 451x331 pixels alone; this is decreased to 230x100 only in HybridPass without increasing error-rates or reducing the system’s password space. If compared to other schemes from the literature, the VP design of HybridPass has the following main advantages:

- Smaller system layout (a flexible design) while maintaining the security and usability of the click-based approach: height and width of the layout and the images employed inside have been reduced to maintain system flexibility and ease of use in web applications and other screen limited environments such as mobiles. In addition, this is expected to help in reducing loading time and resisting shoulder surfing attacks since the size is smaller.

- Ease of navigation: click-based systems (e.g. Passpoints and CCP) have multiple screens to support interaction between users and their portfolios where a single image is systematically displayed at a time on each screen, whilst the VP approach in HybridPass support navigation and user control of images while remaining on the same page until the password is submitted. This allows bigger password spaces in HybridPass since each click can be located on any of the available images in a user portfolio, while in CCP and Passpoints each click can only performed on the image displayed on the screen.

In addition, the laboratory experiments of HybridPass were designed to serve this research as a test-pad to help answering a number of design questions with empirical evidence. For example, we distinguish between two main classes of images: drawings/cartoon images and photographs. It is unclear if a specific type of image can have a positive impact on the usability of the system. Other research questions include studying the effect of using smaller tolerance distance around the click-point, behaviour of first time users and finally the hotspot problem.

The remaining sections will first propose the design and implementation of HybridPass followed by an evaluation of how it performs in view of security and usability supported by the analysis of a laboratory study.
The work presented in this chapter was published in two papers at the IADIS International Conference e-Society 2009 (al-Khateeb et al., 2009) and the IADIS International Conference e-Society 2010 (al-Khateeb et al., 2010).

4.1. Description of the system

The scheme requires a username and two types of passwords: a text password and a VP. Unlike traditional text-based password schemes, its text password need not be long or complicated (consist of digits, letters and symbols). In fact any short memorable string will be appropriate without affecting the security of the login procedure because it is only an initial part of the final password hence the system will not be prone to dictionary attacks.

The system resists phishing scams, a typical attack consists of presenting the user with a fake login screen, say via a link from a phishing email. The hybrid nature of the design provides an anti-phishing technique that forces the sign-in procedure to stop. The correct set of images required to enter the VP in the scheme shows up only if a valid username and a correct text password have been entered. A fake website cannot interact with the user correctly in that way and without the right set of images the users will not be able to start entering their VPs; hence they recognise they are at a spoof site. Stealing the text password alone by this phishing technique is not enough to sign-in. The user must change their text password as a result of entering it to a spoof site.

The VP is cued recall-based where the user input consists of a sequence of a few clicks on an image displayed in a clickable area. The image dimensions are 230 x 100 pixels only and the users are provided with six images displayed as small buttons beside the clickable area. Users can control which image to display continuously by clicking the small buttons while entering their VP. The screenshot of the system in Figure 4.1 (before entering the text password) and Figure 4.3 (after entering the text password) illustrates the idea.
The system can provide a virtually infinite repository of images and the users will have six of them assigned to their account during the registration stage. These images can be changed by the users anytime after signing in.

The six images are not retrieved on the sign-in page until a valid username is entered with a correct text password. This way if a phishing site tries to capture the users’ credentials, the users would not be able to give away their VP because the spoof site wouldn’t know which images to display to a particular user in the first place as explained earlier. Another advantage of this setup is that the user is forced not to repeat the same VP on different systems. Every site adopting this scheme can maintain their own individual repository of images to create VPs.
Figure 4.2: Schematic view of the sign-in process

Figure 4.3: Sign-in form after entering a valid ID and any text password.

After recognizing the correct portfolio (step 2 in Figure 4.3) users can trust the website and start entering their VP by clicking the correct spots in sequence. Hence, our system includes here a recognition-based technique to authenticate the server.

The numbers in Figure 4.3 stand for the following:

1. Two fields to enter a user name and text password.
2. Six preselected images to recreate the VP.
3. The clickable area: clicking any image from (2) displays it within this boundary with a bigger size, thus, the user can click inside the image to create the VP.

4. VP information: keep the user informed of the number of clicks performed so far.

5. Sign-in and reset buttons

The sign-in process is illustrated in Figure 4.2. Each and every click consists of three strings, first and second values are the coordinates (x and y) of the mouse position on an image (e.g. “35;70”, “40;64” etc). Whilst the third string is a special identifier to represent the image itself (e.g. “x!”, “hd”, “?&” etc). Therefore a password of three clicks on different images could be presented with the following code:

35;70;x!;40;64;hd;110;29;?&;

The semicolon is being used as a separator between the different strings to make the example more readable.

Hashing and storing the password in the database is an issue, because users will not be able to click the exact correct spot, for that matter our scheme has some tolerance distance of 4 pixels from the actual click-point. Having tolerance distance means not having the exact same value every time the user signs in, thus, the hash comparison will fail even if the user clicks within the tolerance distance.

This problem can be solved by using Birget et al.’s algorithm using three discretization grids (Birget et al., 2003) or the more accurate Centred Discretization (Chiasson et al., 2008c) approach. Discretization is discussed in further details in Chapter 5 of this thesis in which an enhanced discretization method is developed for this click-based system.

To register, a user ID and a portfolio of six images are assigned to a new user signing up to the system where (s)he is required to choose a text password and perform the minimum number of clicks on different images to create the VP successfully.

4.2. Usability discussion

The ISO defines usability as the “effectiveness, efficiency and satisfaction with which a specified set of users can achieve a specified set of tasks in a particular environment”. These aspects are ultimately concerned with memorability, learnability, error frequency, productivity and satisfaction. HybridPass scheme requires two types of passwords instead of one, which might not be considered as a more user-friendly solution compared to the normal text-based password approach. However after adopting good security
practices text-based password are actually more complicated. For instance, Lloyds (lloydstsb.com, 2010b) secure sign-in procedure for internet banking consist of a nine digit username, a text password with minimum length of 6 characters and a piece of memorable information which must be a mixture of letters and numbers between 6 and 15 characters long. Similar requirements on the choice of the secrets can be identified in other (security critical) applications.

The registration and sign-in process in click-based and recognition-based VP solutions require multiple screens or multiple pages and are typically known to be slow (Suo et al., 2005). In contrast the HybridPass prototype proves that a system can be coded and presented on a single page for easy user interaction using smaller image sizes without affecting the password space. Whilst, the size of the clickable area is limited, in practice there are no problems with finding clickable points if the used images are suitably chosen. We noticed that photographs are indeed harder to use, but an initial prototype test worked very well with cartoon images.

With the use of Ajax (asynchronous JavaScript and XML) technology to submit data to the server and update the client side in return the page will not be refreshed, instead when the text password is entered the set of user images are requested alone. The whole setup of the system with this technology guarantees a smooth user experience.

From the above, our arguments regarding usability are based on theoretical comparisons to demonstrate a new proposal employing up-to-date technology. A prototype has been coded and it is expected that laboratory experiments will confirm the theoretical work (This has been accomplished in user study presented in section 4.4).

4.3. Security discussion

To measure security we discuss the main threats targeting authentication systems such as social engineering (Leyden, 2003), spyware, phishing and password cracking tools. (Cheswick et al., 2003) (Spafford, 1992) (Furnell et al., 2000).

**The password space.** There are 94 standard ASCII keyboard characters, hence in a traditional text based password system, given the advised length of eight characters there is a password space of $94^8 \approx 6 \times 10^{15}$ words.

Our hybrid password is a combination of visual and text passwords where the password space calculates as follows. We should however note that because the number of images to choose from is virtually unlimited we may argue that the password
space is a priori infinite. However for a sensible comparison we calculate assuming that the six chosen images are fixed. We will see that even then we obtain a larger space.

VP space:

The dimensions of our clickable area are 230x100 pixels. If we assume tolerance space 4 pixels away from the actual click-point we have a grid square size of 9 x 9 pixels. That is (230x100) / (9x9) ≈ 284 grid squares. The clickable area is used to present six images continuously (controlled by the user) at the time of signing in, then, the users have 284 x 6 = 1704 grid squares to click. The password space for four clicks will be: 1704^4 ≈ 8.4 x 10^{12}. Five clicks will give 1704^5 ≈ 1.4 x 10^{16}.

Assume the image was poorly selected and has few memorable features; the effective password space is significantly reduced. However, Chapter 6 proposes a technique to aid click-point recall for every grid square in a click-based system.

Text password space:

Four letter text password has a password space of 94^4 ≈ 7.8 x 10^7. Hence the combination of both the text based password space and the virtual password space in HybridPass gives even with the most conservative estimate in total a magnitude of 7.8 x 10^7 x 10^{12} = 7.8 x 10^{18} > 6 x 10^{15}, i.e. much greater than the password space based on an eight character ‘traditional’ password.

**Shoulder-surfing**, observing a user during VP-entry has always been a problem especially that the password entry details are exposed on the monitor. The keyboard could also be watched while typing in the text password. Our scheme uses both, so for shoulder-surfing to succeed, the attacker needs to watch both the monitor and the keyboard. In addition the system does not leave click marks on the screen to make this kind of attack more difficult. Nevertheless, the clickable area size is very small hence less visible in comparison with previously presented click-based schemes such as PassPoints and CCP. The enhanced version of HybridPass demonstrated in Chapter 6 provides better resistance against shoulder-surfing since the text password is not requested in full length.

**Spyware and key loggers**, computers that have been compromised with a keystroke logging system result in stealing passwords typed in by the attached keyboard. HybridPass makes this kind of software useless unless they capture mouse motion and clicks as well as keystrokes used at that same time.
Social engineering, while text passwords can be written down and/or given away easily, by pronouncing the letters, numbers and symbols, VPs require more time to be communicated. Given the complexity of the images used, revealing passwords over the phone or by writing them down will be far less straightforward. With the additional standard safeguards that accounts will be locked after a certain number of failed sign-ins this system will make social engineering more demanding for a prospective intruder.

Guessing and Denial-of-Service, after entering a valid username accompanying a correct text password the system will respond with the correct portfolio to start building the VP. However, if a wrong text password or a wrong username is entered, the system will display a random set of images and save the result in a log table for that particular combination which implies that:

- If a legitimate user enters the correct combination (s)he gets the correct portfolio and proceeds.
- If a legitimate user enters the wrong combination (s)he gets the wrong portfolio and realises (s)he needs to repeat the first step.
- If an attacker enters a wrong combination to perform a guessing attack the system will display a static response for that particular combination. There is no feedback for the attacker to acknowledge whether the password was guessed correctly or not. This technique is used to replace the need to lock the account after a minimum number of failed login attempts which can result in a denial-of-service.

This is an important benefit of the scheme. In addition, the image repository where the user chooses their pass images from is provided by the service provider. Hence the VP does not primarily depend on something related to the user’s personal details. Chances for people to guess someone’s password this way are fewer.

Brute force attack, the large password space of HybridPass makes the system more resistant to this attack compared to a text-based password system and other click-based systems as well.

Dictionary attack, text passwords fail sometimes under dictionary attacks but VPs in HybridPass are much more reliable since the number of images used to generate user portfolios is essentially unlimited. Hence, HybridPass will prevent this attack and remain strong even in front of tools and algorithms which try to analyze images and build
dictionaries of the most selectable points. However, if the attacker recovers the images (gains access to the image identifiers) belonging to the user, a dictionary attack can recover all the passwords contained within popular areas called hotspots. The hotspot problem is analysed in detail in Chapter 6 and a solution is demonstrated to effectively reduce its impact on the system.

*Phishing scam*, authentication system does not include a technique for end users to authenticate the server before submitting their credentials. With HybridPass the server can be authenticated before submitting the VP via a shared secret (user portfolio). If the server fails to show the correct secret, it is considered as a spoof and the submitted text password must be changed to prevent further phishing attacks. The phishing problem is analysed as in Chapter 7.

4.4. Laboratory study

Security and usability are key elements in system design. A bad design will result in inverse proportion between the two, while a good design must find a balance to achieve usable security. The following sections present and discuss the result of a user study to show how good application of click-based systems can produce a system people can easily use while maintaining security.

In this study, participants were asked to do trials using 5 different prototypes of the system. A comparison between them helped to find the best criteria where there is an acceptable balance between security and usability. For instance, using a relatively small tolerance distance enhances security by increasing the password space, the comparison helps to find how small the distance can be while maintaining usability.

Another objective of this study was to distinguish between two classes of images; the results revealed that using cartoon images has positive impact on usability. Nevertheless, hotspots occurred and that makes particular images more vulnerable to dictionary attacks. Experiments also show that, if they can chose, users select images more vulnerable to hotspots.

4.5. Experiment methodology

HybridPass has been tested in a user study with 40 participants who completed the experiment individually. To imitate the environment of a substantial internet application, it was hosted online on a dedicated server to make it remotely accessible from different places. Access locations varied from university labs to private accommodation, internet
cafes and different outdoor places where a wireless connection was used. Participants to a great extent were university students from several fields. All were familiar with computers and have been interacting with internet applications for several years.

The prototype was implemented using PHP, HTML, CSS, JavaScript and SQL. Internet Explorer was used to access the Internet under Windows environment and with a predetermined screen resolution of 1280 x 800 throughout the experiment. In HybridPass there is a clickable area to display the background images instead of showing one image per page, see Figure 4.5 the clickable area dimensions were 230 x 100 pixels. Passwords and click-points were not hashed in the database, so no discretization methods (Birget et al., 2003) (Chiasson et al., 2008c) were used. This was necessary to keep a record of all click-points for further analyses.

Trials that every participant was asked to do are given in the following three steps:

i. Creation: users were asked to enter a User ID, Text Password, Full Name and then still on the same page create their VP. To create a VP, there are images randomly selected by the system to choose from. These images appear as small buttons, but once clicked, they come into view inside a clickable area as a background image. The system forced VPs of four clicks, no more or less and it did not allow continuous clicking on the same click-point. After that the ‘Proceed to sign up’ button should be clicked to start step two.

ii. Confirmation: step two is a confirmation stage where users are asked to re-enter both text and VPs one more time.

Registration is successful if the passwords entered in step one and two match. Consequently, the system forwards users to Sign-in.

If any of the passwords did not match, users are asked to try registering one last time.

iii. Password retention: to calculate the time required to input VPs, the sign-in process has been divided into two phases. Users were first asked to enter their ID and Text Password (Figure 4.4), and then proceed to enter the corresponding VP (Figure 4.5).

If the passwords match the database records for that user ID, user is authenticated, otherwise authentication fails. After a successful registration, Users are asked to enter their credentials to sign-in, one time only.
Five prototypes were developed, with identical code but some parameters set differently to compare results. Hence, participants had to do five trails at least (more than five if registration failed on the first trial), one for each prototype. These parameters were:

i. Image class: researchers have studied memorability of images in the context of recognition-based or in comparison to text passwords but to our knowledge there is no study analysing the effects of different image classes on the usability of click-based systems. (Wiedenbeck et al., 2005b) investigated the impact of four samples under Passpoints and concluded that there were no major effect on the performance, but there were still few significant differences among the individual images. In this study we recognise two main classes of images through dividing them into cartoon (drawing images) and photograph images (real life snapshots taken by camera) to investigate if one class gives better results than the other. Table B.6 in Appendix B shows all the images used in this experiment.

ii. Tolerance: 4 and 6 pixels distance from the actual click-point were used. This implies that, if 4 pixels are permitted, and the original click-point is \((x, y)\) then \((x', y')\) is accepted if \(x' \leq x + 4\) and \(x' \geq x - 4\). And \(y' \leq y + 4\) and \(y' \geq y - 4\).
Table 4.1 shows which parameters were used with which prototype. Some events were recorded throughout the trials such as: click-point coordinates, registration time, sign-in time and the number of sign-in attempts.

Table 4.1: Description of prototype parameters (r is the tolerance distance from the actual click-point).

<table>
<thead>
<tr>
<th>Prototype</th>
<th>Image type</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototype 1</td>
<td>Mixed (cartoon + photographs)</td>
<td>4 pixels</td>
</tr>
<tr>
<td>Prototype 2</td>
<td>Cartoon</td>
<td>4 pixels</td>
</tr>
<tr>
<td>Prototype 3</td>
<td>Photographs</td>
<td>4 pixels</td>
</tr>
<tr>
<td>Prototype 4</td>
<td>Cartoon</td>
<td>6 pixels</td>
</tr>
<tr>
<td>Prototype 5</td>
<td>Photographs</td>
<td>6 pixels</td>
</tr>
</tbody>
</table>

Prototype 1 was used to study how first time users deal with the system, after that, users proceed to prototypes 2, 3, 4 and 5. The objective for prototype 1 was to investigate the number of participants who are able to understand the system by reading the simple instructions provided. Time was logged to calculate the average required by new users to understand how this new authentication scheme works. They had to deal with the system alone without help or training. As such, prototype 1 is considered as the training stage for the following prototypes. Informative discussion followed to explain the system briefly.

After prototype 1 users were expected to be more familiar with the system and use it faster after every trial, therefore, to make the result comparison fair, prototypes 2, 3, 4 and 5 running sequence was random.

4.6. Analysis of results

The participants generated an overall number of 2166 click-points on 15 background images. The number of successful trials was 165, which is the number of times participants were able to register and then try to sign-in to the system. t-test is used for the analysis since they compare variance of the means between two groups. The confidence interval for the difference was always 95%.
4.6.1. Training (prototype 1)

Prototype 1 was used for training and to introduce the system to the participants. Time and the number of incorrect password submissions were measures as shown in Table 4.2 and Table 4.3. Since the prototype was the user's first interaction with the system, they submitted a significantly higher number of incorrect passwords compared to Prototype 2: \( t(78) = 7.22, p < .001 \), Prototype 3: \( t(78) = 5.67, p < .001 \), Prototype 4: \( t(78) = 10.09, p < .001 \), and Prototype 5: \( t(78) = 7.87, p < .001 \). This is reflected by a low registration rate of 42.5%, 15% of them registered on the first trial and the remaining 27.5% on the second trial.

In the first registration trial, it took the users significantly longer to create a new account compared to other prototypes. Creation time in the second trial was shorter: \( t(72) = 6.21, p < .001 \). The minimum time recorded in the second trial decreased as well, as such the number of users who finished their registration in less than one minute rose from 10% on the first trial to 65% on the second one which implies that the scheme is easy to learn.

<table>
<thead>
<tr>
<th></th>
<th>Registration (first trial)</th>
<th>Registration (second trial)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Create</td>
<td>Confirm</td>
</tr>
<tr>
<td>Mean time</td>
<td>153.12</td>
<td>30.5</td>
</tr>
<tr>
<td>SD</td>
<td>89.37</td>
<td>18.34</td>
</tr>
<tr>
<td>Minimum time</td>
<td>50</td>
<td>11</td>
</tr>
<tr>
<td>Number of incorrect submissions (SD):</td>
<td>1.42 (0.74)</td>
<td></td>
</tr>
</tbody>
</table>

Sign-in times are shown in Table 4.3. There was no significant difference between the VP submissions time in Prototype 1 compared to other prototypes.
Table 4.3: Calculated data (in seconds) of the password entry time for those who successfully signed in to the system using Prototype 1 (N = 17).

<table>
<thead>
<tr>
<th></th>
<th>Text password + ID</th>
<th>VP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean time</td>
<td>14.69</td>
<td>19.15</td>
</tr>
<tr>
<td>SD</td>
<td>4.47</td>
<td>7.17</td>
</tr>
<tr>
<td>Minimum time</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

4.6.2. Registration phase

Registration is complete when the user ID, VP (4 clicks) and full name are entered in the first step and the VP is correctly repeated in the following step. Table 4.4 shows the time of successful registrations, success rates and the number of incorrect VP submissions for Prototypes 2, 3, 4 and 5. Significant difference was found in the number of incorrect submissions between Prototype 4 and Prototype 2: \( t(78) = 1.99, p < .05 \) and between Prototype 4 and Prototype 3: \( t(78) = 2.83, p < .01 \). There was no significant different between other prototypes. Nevertheless, time differences are not significant between any of the prototypes.

Table 4.4: Calculated data (in seconds) of the VP creation time for successful registrations, the number of incorrect password submissions, successful rates on the second registration attempt (SR) and the first attempt only (SR-F). (N = 40).

<table>
<thead>
<tr>
<th></th>
<th>Prototype 2 Create</th>
<th>Prototype 2 Confirm</th>
<th>Prototype 3 Create</th>
<th>Prototype 3 Confirm</th>
<th>Prototype 4 Create</th>
<th>Prototype 4 Confirm</th>
<th>Prototype 5 Create</th>
<th>Prototype 5 Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean time</td>
<td>54.51</td>
<td>25.37</td>
<td>50.11</td>
<td>23.51</td>
<td>48.46</td>
<td>24.25</td>
<td>47.45</td>
<td>24.45</td>
</tr>
<tr>
<td>SD</td>
<td>27.62</td>
<td>9.58</td>
<td>19.08</td>
<td>7.53</td>
<td>17.93</td>
<td>10.86</td>
<td>24.08</td>
<td>11.87</td>
</tr>
<tr>
<td>Minimum</td>
<td>21</td>
<td>13</td>
<td>24</td>
<td>11</td>
<td>19</td>
<td>11</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>SR</td>
<td>92.5% (37/40)</td>
<td>87.5% (35/40)</td>
<td>97.5% (39/40)</td>
<td>92.5% (37/40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR-F</td>
<td>75% (30/40)</td>
<td>67.5% (27/40)</td>
<td>95% (38/40)</td>
<td>82.5% (33/40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of incorrect submissions</td>
<td>0.325 (0.61)</td>
<td>0.475 (0.75)</td>
<td>0.1 (0.37)</td>
<td>0.25 (0.58)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.6.3. Retention phase

After a small break following a successful registration, users are asked to retain their passwords and sign-in to the system. Table 4.5 shows the time of successful sign-in attempts, success rates and the number of incorrect VP submissions for Prototypes 2, 3, 4 and 5. No significant differences were found in the number of incorrect VP submissions between any of the prototypes. Further, time differences did not reach significance either.

Table 4.5: Calculated data (in seconds) of the VP input time for successful signed in attempts. The number of incorrect password submissions and success rates (SR). (Prototype 2: N = 37, Prototype 3: N = 35, prototype 4: N = 39, Prototype 5: N = 37).

<table>
<thead>
<tr>
<th></th>
<th>Prototype 2</th>
<th>Prototype 3</th>
<th>Prototype 4</th>
<th>Prototype 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>19.06</td>
<td>19.14</td>
<td>18.86</td>
<td>18.03</td>
</tr>
<tr>
<td>SD</td>
<td>6.74</td>
<td>10.16</td>
<td>8.71</td>
<td>7.04</td>
</tr>
<tr>
<td>Minimum</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>SR</td>
<td>81% (30/37)</td>
<td>80% (28/35)</td>
<td>94.8% (37/39)</td>
<td>86.4% (32/37)</td>
</tr>
<tr>
<td>Number of incorrect submissions</td>
<td>0.18 (0.39)</td>
<td>0.2 (0.40)</td>
<td>0.05 (0.22)</td>
<td>0.13 (0.34)</td>
</tr>
</tbody>
</table>

4.6.4. Accuracy

The rejected VPs during the retention phase from all prototypes were analysed to address the reasons behind the submissions of incorrect VPs. The relevant data shows that 70% of the passwords were rejected because the clicks are 1 to 4 pixels away from the accepted tolerance. This indicated that users are able to recall the chosen area of their click-based passwords in sequence but at times fail to include their click within the tolerance of the system.
Figure 4.6: Analysis of the rejected click-points during the retention phase from all prototypes.

Other reasons include 20% of users who succeeded in reselecting their click-points within tolerance but failed to repeat the original sequence of clicks. The remaining 10% of users failed to use the correct image; hence they submitted a random VP.

4.6.5. Hotspots

In click-based systems, hotspots are areas that comprise most user clicks. If users were not interrupted by the system to select a good password, then there is a high probability that most users consciously or unconsciously select them. This particularly implies that attackers can do image analysis to locate hotspots and perform effective dictionary attacks on click-based systems. Previous research such as (Thorpe and Oorschot, 2007) (Chiasson et al., 2007) (Wolfe, 2000) addressed this problem.

Similarly, this study proves that most clicks are contained within hotspots; and every image has a unique hotspot map. Figure 4.7 illustrates clicks on the most employed cartoon image in prototypes 2 and 4 with 83 user clicks. It shows clusters of clicks in some areas, fewer clicks in other areas and absolutely no click on the rest. Figure 4.9 illustrates the probability distribution of these clicks with a line chart.
Further, an interesting result was that users tended to unconsciously favour images which are more vulnerable to hotspot analysis. Previous implementations forced images on users to click on, but since our approach gave users a choice, this behaviour was observed. This was the case for both types of images across the prototypes; participants selected images with fewer figures. For instance, the most employed photograph image (in prototypes 3 and 5) was the one with the very least features. It got 87 clicks; Figure
4.8 shows how these clicks were highly focused on very few and area-limited hotspots inside the image and Figure 4.10 illustrated the probability distribution of that image in a line chart. People select what is clear and easy to remember. Hence, they created weak VPs due to their lack of experience and training on click-based system security.

Figure 4.10: Line chart illustrating the probability distribution of 87 user clicks on the most employed Photograph image in prototypes 3 and 5. The ‘Equal Distribution’ line represents the case if all clicks are distributed equally (all squares have the same probability).

4.7. Discussion

As expected, users took more time to register when they used the system for the first time. Prototype 1 showed how time was diminished on the second trial and then with the consequent prototypes. After training, the number of incorrect VPs submissions was significantly reduced as well which indicates that the system is easy to learn hence users became familiar with the scheme. Training was important because participants were using the type of visual passwords employed in HybridPass for the first time.

The study distinguished between two types of images: cartoon (drawings) and photographs (camera photos) because the colours in cartoon images are more contrasted so the details are clear, a question to investigate was whether drawings are a better choice for click-based systems than photographs. This study shows that image
type alone causes no significant differences in performance, but however it proves that drawings/cartoon images can reduce the number of incorrect VP submissions. As such, in Table 4.4 and Table 4.5 Prototypes 2 and 3 are identical except for the class of images used. However, prototype 2 shows 5% success rates improvement. The same scenario repeated between prototypes 4 and 5 for the benefit of prototype 4 where cartoon images were used again to conclude that employing drawing images in a click-based system helps to reduce human errors during authentication. Registration rates are measured of those who successfully created and confirmed their credentials on the first or second trial.

Similarly, there was no significant difference caused by the two values of system tolerance. However, using six pixels tolerance distance (13x13 click-square) had a positive impact on the success rates. Registration rates increased around 5% between prototypes 2 and 4 and similarly between 3 and 5. There was more effect on password retention as the success rates elevated from 81% in prototype 2 to 94.8% in prototype 4. Similar impact occurred between prototypes 3 and 5.

Although, image type and the value of system tolerance alone did not result in a significant difference in this study, selecting cartoon images together with six pixels of tolerance caused significant positive impact. This is reflected in significantly higher success rates and less incorrect password submissions in Prototype 4.

Analysis of click-points shows that 70% of the incorrect VP submissions were rejected because users missed the correct click-point by 1 to 4 pixels outside tolerance as illustrated in Figure 4.6. This implies that users were able to recall the area containing the correct click-point but failed to select it. Increasing tolerance distance is a possible solution but doing that will reduce the password space. Hence, a technique must be developed for click-based systems to help users selecting the intended click-points when tolerance distance is small. This problem has been investigated and addressed in Chapter 6. Further, analysis of click-points distribution proves that each image has limited areas which are more attractive to participants. This scenario is similar to text passwords when users tend to select dictionary words. This issue has also been addressed in Chapter 6 with a solution for this matter.

To evaluate HybridPass, the results from Prototype 4 are compared to other click-based systems from the literature as shown in Table 4.6. Registration and password retention rates are 95% for registration and 94.8% for retention. These are high rates compared to other click-based systems. Registration time in HybridPass is 43 to 48
seconds to create a user account and additional 21 to 24 seconds to confirm the VP, thus full registration would take between 64 and 72 seconds. Note that unlike Passpoints and CCP, HybridPass registration procedure included extra time to input a text password, username and the user’s full name. Nevertheless, considering the minimum recorded values, full registration was possible within 30 seconds.

Inputting the VP in HybridPass took 16 to 19 seconds which is longer than the comparative schemes. This comparison should take into account that the majority of users were pausing to comment while entering their passwords. The minimum recorded time shows that inputting the VP was possible with 8 seconds only which is very close to the password retention time in Passpoints and CCP. Another reason can be that users spent more time navigating images in the clickable area.

**Table 4.6:** Means (in seconds), standard deviations (SD) and success rates (after one attempt) for Passpoints (Wiedenbeck et al., 2005c), CCP (Chiasson et al., 2007) and HybridPass (Prototype 4). Registration (Reg.) in this table stands for the time need to successfully create and confirm a new VP.

<table>
<thead>
<tr>
<th></th>
<th>Passpoints</th>
<th>CCP</th>
<th>HybridPass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg. mean time (SD)</td>
<td>64.03 (21.93) / 1.10 = 58.2</td>
<td>24.7 (16.4) + 10.9 (13.1) = 35.6</td>
<td>48.46 + 24.25 = 72.71</td>
</tr>
<tr>
<td>Reg. success rates</td>
<td>95%</td>
<td>82.8%</td>
<td>95%</td>
</tr>
<tr>
<td>Login mean time (SD)</td>
<td>8.78 (4.40)</td>
<td>7.4 (5.5)</td>
<td>18.86 (8.71)</td>
</tr>
<tr>
<td>Week 2</td>
<td>24.25 (15.21)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Week 6</td>
<td>19.38 (17.57)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Login success rates</td>
<td>85%</td>
<td>95.7% (246/257)</td>
<td>94.8% (37/39)</td>
</tr>
<tr>
<td>Week 2</td>
<td>55%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Week 6</td>
<td>55%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
4.8. Conclusion

HybridPass is an application that combines VPs and text passwords in a single authentication solution to increase the password space without forcing long passwords on users. Further, we discussed its resistant to a variety of attacks such as shoulder-surfing, spyware, social engineering, guessing, denial-of-service, brute-force, dictionary and phishing. The analysis of a laboratory study shows the system to be usable with high success rates (95%) and easy to learn but sign-in time is of course longer than a system with a single password. Nevertheless, the VP input time was longer than that of Passpoints and CCP. In conclusion registering a new account including passwords confirmation in HybridPass took between 64 and 72 seconds and the VP input time was between 16 and 19 seconds. However, considering the minimum time recorded it was possible to register a new account in 30 seconds and input the VP in 8 seconds only.

Flexibility of the scheme was also discussed as it has smaller layout and support image-navigation compared to the other click-based systems from the literature. In addition, the study investigated the type of images and tolerance distance (to accept user clicks) most suitable for this application. The results revealed significant increase in success rates while employing cartoon (drawing) images instead of photographs together with six pixels of tolerance. Users recalled their VPs easier when the tolerance distance is bigger, but increasing tolerance decrease the number of possible clicks in the system yet the password space. The objective behind testing relatively small tolerance in the experiment was to identify a usable yet security convenient distance.

Passwords retention was instant i.e. during the same day of registration. Asking users to retain their password after a week or more is expected to decrease success rates. However, our objective in this experiment was to test HybridPass system and not users’ ability to remember click-based VPs in general but this is an important research question to cover hence it is addressed in chapter 6.
Chapter 5  True Discretization

In click-based VPs, discretization is used to accept clicks within an area of tolerance around the original click-point. Robust Discretization (Birget et al., 2003) was first presented as a flexible method to achieve this objective, but it allowed false accepts and false rejects. Centred Discretization (Chiasson et al., 2008c) was then proposed to eliminate the problem and guarantees centred-tolerance.

This chapter shows that implementing centred discretization without modifications in a 2-dimensional space in click-based systems allows false accepts. Hence, True Discretization is introduced as a method to reject clicks outside the tolerance distance followed by a discussion of how this method could increase the theoretical password space.

5.1. Introduction

The normal password authentication procedure works after users enter a string (password). The entered value is hashed and compared to what is stored in the database, if the system finds a match then access rights are granted to user. This is because after registration, passwords are usually hashed with a one way algorithm before storing them in the database of any authentication system.

In click-based VP systems users are expected to sign-in with a VP string that consists of approximate coordinates values within a specific tolerance distance. That is necessary because people’s chances to recall and click the exact pixels all together are very rare. Implementing a system with a tolerance distance implies that the system should accept all string hashes within that tolerance area. A 6 pixels tolerance from the actual click-point results in a 13x13 possible hashes for every click individually. The numbers of hashes multiply with additional clicks, and these are too many to store, thus discretization is used to solve this problem.
Discretization refers to the process of transferring interrelated values of a large set into an equivalent set of limited size. Therefore, it can be used in click-based systems to obtain a stable output (click value) in order to produce a unique password string.

A simple solution is to overlap a static invisible grid upon the background image. In this solution, clicks on any pixel within a specific grid square return a single value. But, a serious usability problem occurs if a user clicks on the edge of the grid square because in this case the tolerance distance will not be equal around the original click-point when the user tries to sign-in. This is known as the ‘edge problem’ and it results in false acceptance and false rejections.

Robust Discretization (Birget et al., 2003) was proposed, but proven to be vulnerable to both false accepts and false rejects. Thereafter, Centred Discretization (Chiasson et al., 2008c) was presented to provide a better solution. However, both methods produced grid square tolerance in the 2-dimensional space. We show how this can result in an effective number of false accepts, then introduce True Discretization as an additional method to reject clicks outside tolerance, compare it to Centred Discretization and discuss how this can affect the security and usability of click-based systems.

5.2. Robust Discretization

Birget et al. addressed the edge problem and proposed the use of multi-discretization grids as an improved solution called Robust Discretization (Birget et al., 2003). They suggested that in a \( d \)-dimensional image, \( d + 1 \) grids are necessary and sufficient to make sure that all points of an image are at a safe distance from grid edges. Hence, in a 2-dimensional environment relying on 3 grids is sufficient. For each click-point, the system assigns the best grid by storing its identifier in the database. This approach guarantees that every pixel in the background image is at a ‘safe’ distance away from the edge of at least one of the grids. However, using three grids only partially prevents the edge problem, because it still does not imply that the tolerance distance is equal around the original click-point in all cases. Figure 5.1 illustrates how this can still result in false rejections and false acceptance. In Figure 5.1 the small circle is the original click-point. The centered-tolerance square is the evenly distributed tolerance likely expected by a user. The dotted square is the grid-square used by Robust Discretization in the worst-case. The non-overlapping region of the centered tolerance square is the area where false rejects would occur in Robust Discretization, while the non-overlapping region of the Robust Discretization square indicates false accepts in Robust Discretization. For further details about the limitation of this method refer to (Chiasson et al., 2008c).
5.3. Centred Discretization

After addressing the limitation of Robust Discretization, Chiasson et al. proposed Centred Discretization (Chiasson et al., 2008c) which offers a centred-tolerance solution. The authors argued ‘the rate of false accepts and false rejects is zero by definition since centered-tolerance implies that the system will only accept click-points that are within $r$ from the original point’.

In this approach each coordinate value is discretized individually along a continuous line $L$ that is the corresponding axis. $L$ is divided into equal segments of length $2r + 1$ where $r$ is the tolerance distance desired, 1 is the original click-point and variable $i$ as the segment identifier. $L$ is moved until $x$ falls in the centre of the segment containing it, where $x$ is a real number representing a coordinate point as shown in Figure 5.2.

Parameter $d$ is the offset where $L$ (the first segment, $i = 0$) should start to centre $x$ in its own segment (where $0 \leq d < 2r + 1$).
Figure 5.2: Centred Discretization. The continuous line L is divided into segments of length $2r+1$, where the value of 1 represents the original click-point.

$d$ is stored as plaintext in the database and can be computed by $d = (x - r) \mod (2r + 1)$, while $i$ is hashed $h(i, d)$. The offset $d$ is included in the hash to uniquely identify the segment. $i$ is computed by $i = \left\lfloor \frac{x-r}{2r+1} \right\rfloor$.

To sign-in, the user is authenticated if the new click-point $x'$ is within $r$ of $x$ or in other words if $x'$ is on the same segment where $x$ is. This can be checked by computing $i' = \left\lfloor \frac{x'-d}{2r+1} \right\rfloor$. If both are on the same segment then $i' = i$ and hence $h(i', d) = h(i, d)$. Otherwise, if the segment $i'$ is not the same as the original segment, the two hashes are not equal and authentication is rejected by the system.

For a 5 click-points VP $(x_1, y_1), \ldots, (x_5, y_5)$, the grid identifiers $(d_1^x, d_1^y, \ldots, d_5^x, d_5^y)$ of these clicks are stored as plaintext, while the hash consists of $h(d_1^x, d_1^y, i_1^x, i_1^y, \ldots, d_5^x, d_5^y, i_5^x, i_5^y)$. Note that $d_1^x, d_1^y$ refers to the grid identifier of the first click $(x_1, y_1)$, $d_2^x, d_2^y$ refers to the grid identifier of the second click and so on.

5.4. Occurrence of false accepts

Centred discretization rate of false accepts and false rejects is zero by definition in a 1-dimensional space as the authors argued because the system will only accept click-points that are within $r$ from the original point. However, since the same approach was suggested for the 2-D environment of VPs without further improvements, false acceptance occurs.

When implementing centred discretization in a 2-D space, the tolerance shape is represented as a grid square. Hence, the tolerance distance ($r$) is not equal around the original click-point, which in the worst case means a click-point 8.48 pixels distant from the original click-point towards the edge of a 13x13 grid square (where $r = 6$) is falsely accepted. And a click-point 12.72 pixels distant will also be considered within tolerance in a 19x19 grid square despite $r = 9$ pixels.
Figure 5.3 illustrates the problem and also shows that while \( r \) is a consistent value, pixels within tolerance around the original click-point shape a circle inside that square, hence further formulas should extend centred discretization calculations to only accept click-points that are within \( r \) from the original point.

![Figure 5.3: Occurrence of false accepts. Each square represents a tolerance grid in centred discretization, and the circle inside contains the true tolerance space in a 2-D environment](image)

Increasing the tolerance distance (\( r \)) increases the size of the grid square, which in return increases the area of false accepts in centred discretization grid square as shown in Table 5.1. In the table, tolerance within (\( r + 0.5 \)) is calculated to include pixels partially located outside tolerance, however they fall along the tolerance circle line as seen in Figure 5.3. Including these pixels is suggested to enhance usability and make the application more convenient to end users.

**Table 5.1:** The number of false accepts click-points in the Centred Discretization tolerance grid square.

<table>
<thead>
<tr>
<th>( r ) (pixels)</th>
<th>Grid Square size</th>
<th>False accepts points within grid when:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Tolerance ( \leq r )</td>
</tr>
<tr>
<td>4</td>
<td>9x9</td>
<td>32 points</td>
</tr>
<tr>
<td>6</td>
<td>13x13</td>
<td>56 points</td>
</tr>
<tr>
<td>9</td>
<td>19x19</td>
<td>116 points</td>
</tr>
</tbody>
</table>
To reject click-points outside tolerance in a 2-D space, we propose True Discretization as an extension to centred discretization where the rate of false accepts equal to zero by definition is achieved.

5.5. True Discretization

After implementing the standard centred discretization approach as in the previous 5 click-points example \((x_1,y_1),..., (x_5,y_5)\) in section 5.3, if \(i' = i\) (the two hashes are identical), a second step follows to check that all click-points are within \(r\) (or \(r + 0.5\)) of the original click. For instance, to verify that \((x'_1, y'_1)\) is within \(r\) of \((x_1,y_1)\) we compute

\[
x_1 = d^x_i + r + ((2r + 1) \times i'_x)
\]

and similarly

\[
y_1 = d^y_i + r + ((2r + 1) \times i'_y).
\]

Now that we have retrieved the original click-point, we calculate the distance using the Euclidean distance formula:

\[
\sqrt{(x_1 - x'_1)^2 + (y_1 - y'_1)^2}.
\]

If

\[
\sqrt{(x_1 - x'_1)^2 + (y_1 - y'_1)^2} \leq r
\]

then \((x'_1, y'_1)\) is within tolerance and system accepts the entry. If

\[
\sqrt{(x_1 - x'_1)^2 + (y_1 - y'_1)^2} > r
\]

then \((x'_1, y'_1)\) is outside the accepted tolerance \(r\), although it is located inside the grid square. Note that in an actual implementation \((x_1 - x'_1)^2 + (y_1 - y'_1)^2 \leq r^2\) can be used instead to avoid using the computations of the square root and therefore floating point arithmetic.

To illustrate the above computation here is a full example, assume that \((x_1,y_1) = (16,12)\) and \(r = 9\). We first apply Centred Discretization normally, so we compute the following:

\[
i^x_i = \left\lfloor \frac{x_1 - r}{2r+1} \right\rfloor = \left\lfloor \frac{16 - 9}{19} \right\rfloor = 0 \quad \text{and} \quad i^y_i = \left\lfloor \frac{y_1 - r}{2r+1} \right\rfloor = \left\lfloor \frac{12 - 9}{19} \right\rfloor = 0
\]

Also,

\[
d^x_i = (x - r) \mod (2r + 1) = 7 \mod 19 = 7
\]

Similarly,

\[
d^y_i = (y - r) \mod (2r + 1) = 3 \mod 19 = 3
\]

Now, \((d^x_i, d^y_i) = (7,3)\) is stored in the clear while the hash is: \(h(d^x_i, d^y_i, i^x_i, i^y_i)\).

If a user clicks \((x'_1, y'_1) = (24,21)\) to sign-in, the system calculates:

\[
i'^x_1 = \left\lfloor \frac{x'_1 - d^x_i}{2r+1} \right\rfloor = \left\lfloor \frac{24 - 7}{19} \right\rfloor = 0 \quad \text{and} \quad i'^y_1 = \left\lfloor \frac{y'_1 - d^y_i}{2r+1} \right\rfloor = \left\lfloor \frac{21 - 3}{19} \right\rfloor = 0
\]
We notice that $i_1^x = i_1'^x$ and $i_1^y = i_1'^y$ which imply that both clicks are on the same segment (first segment) and the two hashes match. True Discretization now checks if the click is within the real tolerance or not. To do that, the original click point is retrieved:

\[
x_1 = d_1^x + r + ((2r + 1) \times i_1^x) = 7 + 9 = 16
\]

\[
y_1 = d_1^y + r + ((2r + 1) \times i_1^y) = 3 + 9 = 12
\]

The distance between the two points is now calculated using:

\[
\sqrt{(x_1 - x_1')^2 + (y_1 - y_1')^2} = \sqrt{(16 - 24)^2 + (12 - 21)^2} \approx 12 \text{ and that is greater than the accepted tolerance, so the click is rejected although it is located inside the grid square.}
\]

5.6. Security analysis

From a security perspective, using true discretization increases the theoretical password space by increasing the possible number of clicks on the background image. For instance, images used in PassPoints (Wiedenbeck et al., 2005c) were 451x331 pixels, with centred discretization the number of possible clicks while $r = 9$ is: $(451 \times 331) / (19 \times 19) = 413$. Using true discretization, this number increases by around 19% to 509 clicks when tolerance is $\leq (r + 0.5)$ and by around 32%, that is up to 609 clicks while tolerance is $\leq r$. For more comparisons, see Table 5.2.

For end users, the accuracy of true discretization becomes more effective and obvious when $r$ is large as in a 19x19 grid square rather than a 9x9 grid square.

<table>
<thead>
<tr>
<th>$r$ (pixels)</th>
<th>N (CD)</th>
<th>N (TD)</th>
<th>N (TD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1842</td>
<td>3046</td>
<td>2163</td>
</tr>
<tr>
<td>6</td>
<td>883</td>
<td>1321</td>
<td>1089</td>
</tr>
<tr>
<td>9</td>
<td>413</td>
<td>609</td>
<td>509</td>
</tr>
</tbody>
</table>

Table 5.2: The number of possible clicks (N) while using Centred Discretization (CD) and True Discretization (TD) in PassPoints.
5.7. Conclusion

Representing the tolerance area as a grid square in click-based authentication systems reduces the number of possible clicks on the background image, which in return reduces the theoretical password space of the system. Application of True Discretization adds to the system’s accuracy by allowing discretization with zero false accepts rate since all clicks outside the accepted distance are rejected.

True Discretization efficiency in an application is in direct proportion to the distance of tolerance (r) and since schemes use large grid squares to enhance usability (e.g. PassPoints was implemented with a 20x20 grid square) implementing True Discretization should be the preferred method.
Chapter 6  CCBS – A Method to Maintain Memorability, Accuracy of Password Submission and the Effective Password Space

Hotspots reveal a serious vulnerability in click-based systems. They occur because users are consciously or unconsciously attracted to specific parts of an image and neglect other areas. Undertaking image analysis to identify these high probability areas can assist dictionary attacks to reduce the effective password space and fool the system.

Another concern is that click-based systems do not guide users towards the correct click-point they are aiming to select. For instance, users might recall the correct spot or area but still fail to include their click within the tolerance distance around the original click-point which generates more incorrect password submissions. Increasing the tolerance can eliminate this problem but it reduces the overall effective password space of the system.

Nevertheless, the Passpoints study inspected the retention of their VP in comparison with text passwords over the long term. Despite being cued-recall the successful rate of their VP submission was not superior to text passwords as it decreased from 85% (the instant retention on the day of registration) to 55% after 2 weeks. This result was identical to that of the text password in the same experiment. The successful submission rates after 6 weeks were also 55% for both VP and text passwords.

This chapter investigates and addresses these concerns, then presents a novel method as a usable solution. Hence, the remaining sections will consequently introduce a new prototype, HybridPass, where the new method is implemented to provide empirical proof for our work. For comparison and evaluation purposes we refer to the new prototype as Cued Click-Based System (CCBS). A user study is then conducted and the results are evaluated against a comparative study.
6.1. Introduction

Password strength is a measure of a string’s resistance to guessing attacks until it is successfully recovered. This is estimated by the entropy or the bit strength of the password. To find the bit strength of a password, the number of possibly used symbols to create the password is translated into binary, either 1 or 0. Hence, a password with an entropy (or strength) equal to 50 bits is same or as strong as a binary number of 50 random bits (Burr et al., 2006, Shannon, 1951).

Randomness is a key of password strength. Random passwords consist of a string of symbols of a designated length randomly selected from a pool of symbols where each symbol is equally likely to be used. The ASCII character set for instance can be employed to generate passwords using a random selection process. In practice, randomness is not effectively achieved when selecting the symbols is a human-driven process. People tend to follow predictable patterns to select passwords because these are easier to recall in the future.

Attackers exploit possible patterns to reduce the number of possible string combination in a password (the password space) and perform efficient dictionary attack against the system. Hence, the effective password space is much smaller than the theoretical password space of the system if the password was not selected carefully.

Due to the limitation of current technology, text passwords are relatively secure against guessing, dictionary and brute-force attacks when their length is eight characters or more, consist of a complex mix of characters (digits, letters and symbols) and are absolutely random, but that is hard to achieve. (Furnell, 2003) and (Belgers, 1993) discuss secure practices to create good text passwords in further detail.

ASCII keyboards have 94 printable characters, hence in a traditional text based password system, given the advised length of eight characters there is a password space of \( 94^8 \approx 6 \times 10^{15} \) words (that is 52.4 bits of strength).

\( 6 \times 10^{15} \) is the number of possible random strings we obtain from the ASCII character pool. However, the distribution of the strings is not uniform since users tend to use a well-known set of strings and ignore the rest. Therefore, cracking tools such as L0phtCrack (l0phtcrack.com, 2009) and John the Ripper (openwall.com, 2009) exploit this human vulnerability by building lists and dictionaries of known or possible words to crack passwords instead of performing a time consuming brute-force attack. (Klein, 1992) for example, performed a cracking test on several UNIX /etc/passwd files, making
all together a database of nearly 15,000 account entries and the dictionary attack was able to reveal 2.7% of these passwords within 15 minutes, 21% (nearly 3,000 passwords) at the end of the first week and finally 25% of the passwords were successfully cracked at the end of the 12 CPU months of exhaustive testing.

Nevertheless, there are lookup tables known as Rainbow tables used to offer a time-memory trade-off to reduce the cryptanalysis time to recover passwords from the cipher. The basic approach is to search for the hash (encrypted password) in a huge database containing precomputed hashes along with their original values to find a match (the hash is compared to these values starting with the smallest possible block of data until the match is found throughout the algorithm). New techniques such as (Oechslin, 2003) and (Avoine et al., 2008), introduce improved methods to reduce recovery time (e.g. by exploiting certain factors to reduce the number of lookup tables and highlight a smaller search area) and perform more efficient attacks against MD5 and MS-Windows password hashes. To make this attack impractical, a random value called a salt is added to the original password to make it stronger (longer and more random). The salt value should only be known to the system; otherwise, the hash will again be vulnerable. In conclusion, randomness is critical to password security, but it is hard to achieve.

Based on these results a text password, regardless of its length, has an effective password space much smaller than the theoretical space. Further example, if we assume that a group of users tend to use an English word as a password, the effective password space in this case will be equal to the number of words in the English dictionary, this is impossible to count accurately, but the number approaches three quarters of a million words only as estimated by Oxford Dictionary (AskOxford, 2009). Nevertheless, the total size of the dictionary used by Klein to reveal 25% of the passwords had no more than 62,727 words.

One effective, but time and resource consuming approach to decrease the risk of dictionary attacks is to identify weak passwords before an attacker can reveal them with a cracking tool. Examples of proactive password techniques are: ‘High Dictionary Compression for Proactive Password Checking’ (Bergadano et al., 1998) and ‘HYPPOCRATES: a new proactive password checker’ (Blundo et al., 2004). Another technique is to disallow user choice and force generated passwords on users, but this usually leads to usability complications.

Similarly, click-based passwords are vulnerable to dictionary attacks as discussed and analysed in the following section. Users find the retention of a click-based VP easier if included within specific hotspots of an image. Some click-points are also easier to
select based on their location in a particular image. For instance, recalling then selecting a click-point visually represented by the edge of a square is easier than selecting a click-point located inside the square in an empty space. As such, click-based systems require a method to support all their click-points with a memorable cue.

The remainder of this chapter is outlined as follows: Sections 6.2, 6.3 and 6.4 provide background information and discussion of related research work. Section 6.5 proposes CCBS. Hypotheses are in Section 6.6. Section 6.7 introduces the methodology used in our user study. Section 6.8 presents the results of the experiment and Section 6.9 discusses the experiments results. Section 6.10 discusses validation of hypotheses and Section 6.11 introduces potential enhancements to CCBS. Finally, conclusions are presented in Section 6.12.

6.2. The effective password space in click-based VPs

Click-based VP schemes can have a large theoretical password space. For instance, the VP space for five clicks in HybridPass is approximately $1.4 \times 10^{16}$ (Section 4.3 of this thesis). This can easily be increased by using more grid squares, which is obtained by expanding the clickable area or adding more images in the user portfolio. However, the effective password space in click-based schemes is smaller than the theoretical space. That is because, if people are not guided or interrupted they are attracted to a limited number of predictable areas when looking at an image (Wolfe, 2000, Thorpe and Oorschot, 2007).

(Erik, 2009) developed a click-based prototype called PassClicks. To register, participants are required to submit five clicks on the image shown in Figure 6.1. Figure 6.2 illustrates the locations of 157,090 click-points collected over time with a colour overlay. It shows how the vast majority of users were attracted to a number of limited spots which are referred to as hotspots.
If an attacker finds the image used to create a VP for a single user, then hotspots can be used to perform an effective dictionary attack on the system and reveal the password. For instance, (Thorpe and Oorschot, 2007) used data from a relatively small set of users to explore hotspots and popular clusters on different images. Figure 6.3 shows an example of the images used. Hotspots and clustered areas are then illustrated in Figure 6.4 for that image. Data were consequently used to feed a dictionary to attack passwords generated with the same images but from a long-term field study. The attack succeeded to correctly guess 36% of the passwords within $2^{31}$ guesses. This provided empirical evidence that popular points (hotspots) do exist and they are useful in executing attacks on this type of VP schemes by guessing the passwords based on the click-points of a small set of users.
Similarly, the analysis of our HybridPass experiment in Section 4.6.5 shows that most clicks are contained within hotspots which can be used to draw a map of high probability areas for every image. Figure 4.7 and Figure 4.8 illustrate clicks on the most employed images in the mentioned user study. They show clusters of clicks in some areas, fewer clicks in other areas and absolutely no click on the rest. Further, Figure 4.9 and Figure 4.10 illustrate the probability distribution of these clicks, showing that a few spots have a very high probability of being selected while other areas do not.

To have some control over the hotspots problem, initial techniques can be used to identify and exclude vulnerable images. For example, (Dirik et al., 2007) is a model to automatically evaluate and identify hotspots in a given image. If the image is too vulnerable to hotspots, it should not be used in a click-based system for security reasons. While this approach is helpful as a primary method to identify relatively bad images, the hotspot problem still exist to reduce the effective password space in accepted images. Section 2.1 of the same paper where the model is presented shows how carefully selected images used in research work are also vulnerable to hotspots.

Another proposed solution to work around the hotspots problem was to increase the effective password space by influencing users’ choice to select more random click-points. Persuasive Cued Click-Points (PCCP) by (Chiasson et al., 2008b) does that by highlighting a random area (square) in the image being used to create a password. A user may not click outside this area, but if they struggle with finding suitable points to click, they can press a shuffle button to randomly reposition the highlighted area. See Figure 6.5 for a screenshot.
The objective of PCCP to achieve randomness is accomplished by forcing a highlighted random area on users. This solution guided users to areas they would have ignored without the software guidance, but it does not aim to add more possible click-points to the background image. Consequently, if a poor image is used or if the highlighted area was a featureless part of the image, users would need to click the shuffle button continuously to locate an area with a better figure, hence low probability areas will still be ignored. Furthermore, if the highlighted area happened to contain a high probability hotspot, users are expected to choose it and ignore other available click-points. In conclusion PCCP can achieve better distribution of clicks compared to Passpoints and CCP, but cannot achieve randomness nor increase the usability of low probability areas.

6.3. Accuracy of password submission

The laboratory study of HybridPass shows that 70% of the incorrect clicks submitted by users were rejected due to exceeding tolerance by up to 4 pixels only. (Wiedenbeck et al., 2005b) conducted a study to examine tolerance effect concluded that smaller tolerances (10x10 in their case) are harder to encode in users’ memory, hence resulting in more incorrect password submissions. Nevertheless, retaining the VPs after one week shows that the number of incorrect submissions with the smaller tolerance (10x10) was

Figure 6.5: Screenshot of the PCCP Create Password interface with the randomly highlighted area. Taken from (Chiasson et al., 2008b).
significantly higher than the larger tolerance of 14x14. Further, their original study adopted a 20x20 tolerance to support more usability.

This problem persists because password cues in click-based systems guide users towards areas and not specific click-points. Increasing the tolerance can eliminate this problem but it reduces the overall effective password space of the system. In CCP (Chiasson et al., 2007), every click results in a unique path of images until the VP is submitted. This helps the user to reselect a click-point before password submission if the consequent image is not part of their portfolio. While this is a superior technique in comparison to Passpoints with regards to correct password submissions, it can be time consuming and exposes the system to shoulder-surfing attacks. CCP vulnerability to shoulder-surfing was addressed by its authors.

Whilst, a good application can maintain high success rates (e.g. HybridPass), we argue that smaller tolerance (e.g. 9x9) can still be usable with CCBS.

6.4. VP retention in click-based systems

Cued-recall authentication such as click-based systems provides cues to trigger users’ memory while entering their password. Each cue should aid the LTM to retain a particular task successfully. However, a laboratory study by (Wiedenbeck et al., 2005c) showed that the number of participants who failed to submit valid click-based passwords during the experiment was almost identical to that of users who were asked to retain text passwords. Success rates for both type of passwords decreased from 85% (the instant retention on the day of registration) to 55% after 1 week from registration/first retention (R1) and the same percentage of 55% was achieved after 4 weeks from the second password retention (R2) as illustrated in Figure 6.6.

This implies that the visual cues in click-based systems are still not sufficient to significantly maintain users’ memory to recall passwords. Hence, we add a new layer of cues in the method we are introducing in the remaining sections of this chapter.
Figure 6.6: Number of participants who failed to submit a valid password on the first attempt. R1 represent instant retention of the VP, R2 represent the retention phase after one week and R3 after 4 weeks from R2. The figure has been taken from (Wiedenbeck et al., 2005c).

6.5. Cued Click-Based System (CCBS)

We propose Cued Click-Based System (CCBS) as a method to overcome some of the main limitations of click-based systems discussed in the previous sections of this chapter. In CCBS, two types of cues are implemented to trigger the user’s memory: graphical and textual, to retain and submit the correct click-points.

Each image is transparently divided into click-cells representing the available symbols to form a VP. The visual cues to recall these click-cells (similar to other click-based systems) consist of all or part of the figures and features of an image existing in the area of that particular cell. In addition, each click-cell is accompanied by a unique textual cue as illustrated in Figure 6.7. The textual cue appears when the relevant cell is hovered by the mouse.

Figure 6.7: In addition to the visual cues, in CCBS each click-cell has a unique textual cue appearing on ‘mouse hovering’.
In an attempt to maintain memorability, these cues are formed of short but informative sentences. Further, it is essential to avoid confusing users by locating similar textual cues next to each other. For instance, if one cue is talking about London being the capital of the UK, it is better that the cues of the click-cells next to it have completely different information that doesn’t include keywords like London, Capital or the UK. Table 6.1 demonstrates examples of the textual cues used in CCBS.

**Table 6.1:** Examples of the textual cues used in CCBS with their relevant click-cells.

<table>
<thead>
<tr>
<th>Click-cells</th>
<th>Coordinates</th>
<th>Textual cues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0,0,9,9</td>
<td>Orange is the colour of the Dutch Royal Family.</td>
</tr>
<tr>
<td>2</td>
<td>9,0,18,9</td>
<td>Spain won the 2010 FIFA World Cup.</td>
</tr>
<tr>
<td>3</td>
<td>18,0,27,9</td>
<td>The pound sterling is the official currency of the UK.</td>
</tr>
<tr>
<td>4</td>
<td>27,0,36,9</td>
<td>Venus is the Roman goddess of love and beauty.</td>
</tr>
<tr>
<td>5</td>
<td>36,0,45,9</td>
<td>Nokia is a Finnish multinational communications corp.</td>
</tr>
<tr>
<td>6</td>
<td>45,0,54,9</td>
<td>Charles Dickens is a popular British novelist.</td>
</tr>
</tbody>
</table>

In conclusion, CCBS has dual cued-recall approach to maintain memorability for each click-cell in the system in an attempt to increase the effective password space. However, to achieve a close to equal distribution of click-points, the VPs must be generated by the system and assigned to end users during registration. In the following user study, we provide empirical evidence that preventing users from creating their own passwords in CCBS doesn’t cause memorability problems in comparison with the comparative user studies (i.e. CCP and Passpoints).

In an authentication system where the symbols are equally distributed, the number of symbols available in the effective space $N_E$ is equal to the number of symbols available $N_T$, i.e. $|N_E| = |N_T|$.

Hence, if the probability of $A = P(A) = \frac{1}{|N_T|}$ where $A \in N_T$.

And $P(B) = \frac{1}{|N_E|}$ where $B \in N_E$.

Then $P(B) = P(A) = 1$

**6.6. Hypotheses**

On applying CCBS, the following hypotheses are stated:

First, in response to the ‘effective password space’ problem covered in Section 6.2:
1) A uniform distribution of click-points (no hotspots) will be achieved via system-generated passwords while maintaining memorability.

Second, in response to the ‘accuracy of password submission’ problem covered in Section 6.3:

2) Users can accurately select the intended click-points to input their VPs hence the number of incorrect password submissions will significantly reduce.

And finally, in response to the ‘VP retention’ problem covered in Section 6.4:

3) Success rates of the VP retention in CCBS will be significantly higher than the comparative schemes: Passpoints and Alphanumeric (text password) on the long term.

To investigate the efficiency and usability of CCBS, it was implemented into a new prototype of the HybridPass scheme. The results were then evaluated against comparative studies and the hypotheses were validated respectively.

6.7. Experiment methodology

The HybridPass prototype has been developed to match the requirements of this experiment. For instance, the text password field has been removed to match that of the comparative study: Passpoints (Wiedenbeck et al., 2005c).

6.7.1. Experimental design

The experiment continued for 6 weeks and consisted of 3 sessions. Session 1 was undertaken during week 1 in which participants were introduced to the system, and then asked to create a new user account using a VP that is randomly assigned to them i.e. not of their choice. This was followed by a learning task where the VP is requested multiple times. Participants were then asked to complete a questionnaire about their perceptions of the system. Finally, they were asked to perform their first VP retention trial and sign-in to their accounts. Consequently, sessions 2 and 3 took place during weeks 2 and 6 where the participants were asked to retain their VPs. Session 3 was followed by another questionnaire about their experience with the system.

The number of attempts and amount of time to create or retain and submit a valid VP was measured throughout the experiment.
6.7.2. Materials

The system was implemented based on the HybridPass prototype but the text password interfaces were excluded. The clickable area can display up to six different pictures and in addition to the visual representation of click-points, textual cues are used. As such, when a click-point is highlighted (hovered over) by the mouse the relevant textual cue is displayed as illustrated in Figure 6.8.

![Sign-in page of the prototype employed in the CCBS experiment.](image)

**Figure 6.8:** The Sign-in page of the prototype employed in the CCBS experiment. The mouse is highlighting a particular click-point hence the relevant textual cue is displayed: ‘Tom and Jerry is an animated show watched by children, teenagers and adults’.

The same six images are used to create portfolios and random VPs to all users, these are displayed in Appendix C. The size of the clickable area was 230x100 pixels and the tolerance around the original click was set to 4 pixels, which represents each click-point with a 9x9 grid square. Hence, instead of returning the coordinates of the selected click-points the system calculates an identifier of the grid square containing the click-point. Assuming that \( r = 4 \), then the square identifier of a captured coordinates \((x, y) = \left(\left\lfloor \frac{x}{r+1} \right\rfloor, \left\lfloor \frac{y}{r+1} \right\rfloor\right)\). In our JavaScript code this is translated into:
var r = 4;

var imgClickx = Math.floor(xPosition/(2*r+1));

var imgClicky = Math.floor(yPosition/(2*r+1));

The layout included two buttons: the ‘Click here to sign-in’ button was used to submit the password and the ‘Reset’ button to erase the selected click-points. The prototype was designed with screen tips to register a new account or retain a password. In addition, feedbacks are displayed in response to invalid or wrong password submission, invalid username, successful authentication, sign out etc.

The prototype was developed using PHP, HTML, CSS and JavaScript. It was hosted online and accessed through a web browser to imitate the environment and speed of a real web application. A single computer was used in this experiment with a screen resolution of 1280 x 800.

The experiment included a questionnaire in which the perception of end users towards the system is measured. The questions were the same as what have been used in the comparative study and they were answered based on a 7-point Likert scale.

6.7.3. Procedure

The experiment was completed individually. Participants were first introduced to the procedure of the experiment in a 5 minute presentation including a brief demonstration of the system.

In session one the participants were asked to follow the instructions on the screen to register new accounts. The registration form included two input fields to capture the user ID and full name and an input method to capture the VP. However, VPs were not entered based on the participant’s preference but rather randomly assigned to them. In an attempt to maintain uniform distribution, the VPs used in this experiment were randomly generated prior to session one. A unique VP formed of 5 click-points is shown to each participant during registration to adopt and use. This was accomplished via an illustration as seen in Figure 6.9.
Figure 6.9: An illustration of a single click-point. The click-point is visually highlighted inside the image and the textual cue is displayed next to it.

Participants looked at the area of each click-point and moved the mouse until the correct textual cue is returned by the system. If the relevant cue is displayed, they perform a click to select it. They were asked to memorise these click-points and their order to select them again in the future. The registration form was validated using JavaScript, thus the ‘Submit’ button can be clicked if the ID and full name fields are filled and exactly 5 click-points are selected. The following step is for password confirmation, participants are asked to re-enter their VP one more time. Registration is successful if the VPs entered in steps one and two match. Nevertheless, if any of the passwords did not match, users are asked to repeat their registration.

After a successful registration, users are asked to start a learning task. The learning task consists of multiple password submissions and it continues until the participant succeeds to submit the correct password 10 times. However, it is part of the procedure to show the correct password again to the participant after incorrect password submission. Then to distract the participants from the system, they are asked to complete a questionnaire. After around 30 minutes, they were asked to sign-in to the system to do their first VP retention trial, R1.

In session 2, users are asked to retain their passwords for the second time, R2. If the password is wrong they can try again until they submit the correct one. However, after five attempts users can see their correct password to refresh their memory and use it to sign-in.

Finally in session 3, users are asked to retain their passwords for the last time similar to session 2. Further, they are asked again to fill out a questionnaire regarding the experiment.
6.7.4. Participants

The comparative study had 20 participants taking part in the graphic password scheme. This study included the same number. Participants were computer science and business students who use computers on a regular basis. Most of them were Masters or PhD students. The mean age is 26.65 years (SD = 2.79) and the range was between 23 and 34 years. There were 11 females and 9 males in the sample.

6.8. Analysis of results

To evaluate CCBS, the results are reported and analyses below against the comparative study: Passpoints (Wiedenbeck et al., 2005c). Data from the registration phase is first demonstrated followed by the learning and retention phases. Further, click-points from the experiments are analysed and illustrated.

6.8.1. Registration phase

Registration is complete when the user ID, VP (5 clicks) and full name are entered in the first step and the VP is confirmed correctly in the second step. Table 6.2 compares the time (of all attempts) and the number of attempts required to register a new user account with the Passpoints scheme and the alphanumeric password from the comparative study. Password confirmation time is reported for CCBS only because the comparative study did not include a confirmation step.

Student's t-test was used to analyse and compare the results. The total number of attempts to register a new account was less in CCBS. The difference was significant compared to the text password: t(38) = 10.22, p <.005 and not significant compared to Passpoints. In CCBS, 19 participants were able to register from the first attempt versus 11 of 20 participants who took two or more attempts to create a valid password. However, registering a new account in CCBS took significantly more time in contrast with the text password: t(38) = 3.3, p <.005 and Passpoints: t(38) = 7, p <.005. However, considering that the difference in means is 31 seconds with the text password and 48 seconds with Passpoints, this does not imply a problem with the CCBS scheme considering that the time measured for Passpoints is for selecting 5 click-points correctly, while in CCBS it included inputting the user ID and full name as well.
Table 6.2: Calculated data of the total number of attempts and time to register and confirm (in seconds) a new account. Confirmation time is the time spent to re-enter the 5 click-points. (CCBS N = 20, Passpoints N = 20, Text N = 20).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Total attempts</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCBS</td>
<td>1.05 (0.22)</td>
<td></td>
</tr>
<tr>
<td>Passpoints</td>
<td>1.10 (0.07)</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>1.70 (0.18)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Total time to register</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCBS</td>
<td>112.47 (21.79)</td>
<td></td>
</tr>
<tr>
<td>Passpoints</td>
<td>64.03 (21.93)</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>81.10 (36.50)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Total time to confirm</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCBS</td>
<td>54.04 (23.85)</td>
<td></td>
</tr>
</tbody>
</table>

Two questions were asked after the registration phase as shown in Table 6.3. t-test analyses of the results show no significant difference between the schemes.

Table 6.3: Questions about the registration phase. A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS N = 20, Passpoints N = 20, Text N = 20).

<table>
<thead>
<tr>
<th>The question</th>
<th>Scheme</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I did not have much trouble creating a password</td>
<td>CCBS</td>
<td>2.35 (1.18)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>2.35 (1.57)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>3.30 (1.59)</td>
</tr>
<tr>
<td>It did not take me long to create the password</td>
<td>CCBS</td>
<td>2.95 (1.09)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>2.60 (1.42)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>3.15 (1.63)</td>
</tr>
</tbody>
</table>

6.8.2. Learning phase

After registration, participants were asked to undertake a learning task by entering their password to the system until they sign-in 10 times successfully. Table 6.4 shows the means and standard deviations of the number of incorrect submissions and submission time during the learning phase.
Analysing the results using $t$-test shows that the number of incorrect submissions in the CCBS scheme was significantly less than Passpoints: $t(38) = 2.9$, $p < .01$ while the difference with the Text scheme did not reach significance. In CCBS 3 participants had a single incorrect submission compared to 4 participants with 1 incorrect submission and 2 with 2 incorrect submissions in the Text password scheme. Comparisons of the details are illustrated in Figure 6.10.

**Table 6.4:** Means and Standard Deviations (SD) of the total practice time (in seconds) and the number of incorrect password submissions in the learning phase. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of incorrect submissions</strong></td>
<td></td>
</tr>
<tr>
<td>CCBS</td>
<td>0.15 (0.36)</td>
</tr>
<tr>
<td>Passpoints</td>
<td>4.80 (7.16)</td>
</tr>
<tr>
<td>Text</td>
<td>0.40 (0.68)</td>
</tr>
<tr>
<td><strong>Total practice time</strong></td>
<td></td>
</tr>
<tr>
<td>CCBS</td>
<td>37.18 (11.10)</td>
</tr>
<tr>
<td>Passpoints</td>
<td>171.89 (24.46)</td>
</tr>
<tr>
<td>Text</td>
<td>66.08 (04.92)</td>
</tr>
</tbody>
</table>

Five questions were asked after the learning phase as shown in Table 6.5. Result analysis of question one shows the participants of the Text scheme agree that it did not take them long to input their passwords 10 times with a significant difference compared to Passpoints: $t(38) = 3.49$, $p < .005$ and CCBS: $t(38) = 5.87$, $p < .005$. There was no significant difference between CCBS and Passpoints. In question two there was a significant difference between CCBS and Passpoints: $t(38) = 2.6$, $p < .02$ and Text & Passpoints: $t(38) = 2.14$, $p < .05$. There was no significant difference between CCBS and Text. Further, no significant differences found in questions three and four. Finally, in question five there was a significant difference between CCBS and Text: $t(38) = 2.88$, $p < .01$ which implies that participants found that inputting a text password is easier than using CCBS. However, the difference between CCBS and Passpoints did not reach significance.
Figure 6.10: the number of incorrect submissions in the learning phase.

Table 6.5: Means and Standard Deviations (SD) of questions about the learning phase.
A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

<table>
<thead>
<tr>
<th>The question</th>
<th>Scheme</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. It did not take me long to input my password 10 times</td>
<td>CCBS</td>
<td>4.1 (1.74)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>3.40 (2.14)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>1.65 (0.67)</td>
</tr>
<tr>
<td>2. Once I created my password I was able to input it correctly</td>
<td>CCBS</td>
<td>1.5 (0.51)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>2.60 (1.82)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>1.65 (0.79)</td>
</tr>
<tr>
<td>3. My password input got better with practice</td>
<td>CCBS</td>
<td>1.2 (0.41)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>1.15 (0.50)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>1.20 (0.52)</td>
</tr>
<tr>
<td>4. Inputting my password was easy</td>
<td>CCBS</td>
<td>2.55 (1.23)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>2.70 (2.18)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>1.90 (1.02)</td>
</tr>
<tr>
<td>5. Inputting my password was fast</td>
<td>CCBS</td>
<td>4 (1.71)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>3.05 (1.73)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>2.35 (1.14)</td>
</tr>
</tbody>
</table>
6.8.3. Retention phase

Participants retained their passwords for the first time (R1) at the end of the first session, then in session two (R2) and session three (R3) respectively. The number of incorrect submissions and time for the correct password submission are measures and compared to the results from the comparative study in Table 6.6. There were no incorrect submissions for CCBS in R1. t-test shows a significant difference with Passpoints: \( t(38) = 4.41, p < .005 \) and no significant difference with Text. In R2, the number of incorrect submissions in CCBS are significantly less than Passpoints: \( t(38) = 2.3, p < .05 \) and Text: \( t(38) = 2.28, p < .05 \). In R3, the number of incorrect submissions in CCBS is fewer but no significant difference was found with Passpoints or Text.

Table 6.6: Means and Standard Deviations (SD) of the time for correct submissions (in seconds) and the number of incorrect password submissions in the retention phase. (CCBS \( N = 20 \), Passpoints \( N = 20 \), Text \( N = 20 \)).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Mean R1 (SD)</th>
<th>Mean R2 (SD)</th>
<th>Mean R3 (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCBS</td>
<td>0.0 (0.0)</td>
<td>0.6 (1.53)</td>
<td>0.55 (1.53)</td>
</tr>
<tr>
<td>Passpoints</td>
<td>1.55 (1.57)</td>
<td>2.75 (3.88)</td>
<td>1.50 (2.80)</td>
</tr>
<tr>
<td>Text</td>
<td>0.25 (0.79)</td>
<td>2.20 (2.73)</td>
<td>1.75 (2.47)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of incorrect submissions</th>
<th>Mean R1 (SD)</th>
<th>Mean R2 (SD)</th>
<th>Mean R3 (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCBS</td>
<td>27.9 (7.63)</td>
<td>69.55 (38.22)</td>
<td>59.65 (24.05)</td>
</tr>
<tr>
<td>Passpoints</td>
<td>8.78 (4.40)</td>
<td>24.25 (15.21)</td>
<td>19.38 (17.57)</td>
</tr>
<tr>
<td>Text</td>
<td>5.23 (1.66)</td>
<td>9.42 (3.70)</td>
<td>9.24 (03.72)</td>
</tr>
</tbody>
</table>

Time for correct submissions is more in CCBS across R1, R2 and R3. In R1, the difference was significant with Passpoints: \( t(38) = 9.7, p < .001 \) and Text \( t(38) = 12.98, p < .001 \). In R2, the difference was significant with Passpoints: \( t(38) = 4.92, p < .001 \) and Text: \( t(38) = 7, p < .001 \). In R3, the difference was significant with Passpoints: \( t(38) = 6.04, p < .001 \) and Text: \( t(38) = 9.26, p < .001 \).

Further, the number of participants who failed to submit the correct password at their first attempt in each session is calculated. The result is illustrated in Figure 6.11 and compared to Passpoints and Text. Figure 6.12 illustrates the number of participants who failed to submit the correct password after five attempts.
Figure 6.11: The number of participants in each retention trial who failed to submit their correct password on the first attempt.

Figure 6.12: The number of participants in each retention trial who failed to submit their correct password after five attempts.

Four questions were asked after the final retention phase as shown in Table 6.7. t-test analyses of the results shows no significant difference in question one. In question two, there was one significant difference between CCBS and Text: $t(38) = 3.71, p < .01$. Hence, participants of the text scheme agree more that inputting their password was
The same apply to question three, there was one significant difference between CCBS and Text: $t(38) = 2.39, p < .05$. The participants in this question agree more than CCBS was pleasant to use. Finally, there was a significant difference between Passpoints and Text: $t(38) = 2.81, p < .01$ and between CCBS and Text: $t(38) = 4.09, p < .001$. Hence, there was significant agreement that remembering the password is easier in CCBS and Passpoints.

Table 6.7: Means and Standard Deviations (SD) of questions about the retention phase. A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

<table>
<thead>
<tr>
<th>The question</th>
<th>Scheme</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Inputting my password was easy</td>
<td>CCBS</td>
<td>2.45 (0.99)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>2.70 (2.18)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>1.90 (1.02)</td>
</tr>
<tr>
<td>2. Inputting my password was fast</td>
<td>CCBS</td>
<td>4.05 (1.70)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>3.05 (1.73)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>2.35 (1.14)</td>
</tr>
<tr>
<td>3. I think the password system was pleasant to use</td>
<td>CCBS</td>
<td>2.10 (1.02)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>2.40 (1.57)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>3.00 (1.34)</td>
</tr>
<tr>
<td>4. I think that the rules [about password creation] make it easy to remember the password</td>
<td>CCBS</td>
<td>3.35 (1.18)</td>
</tr>
<tr>
<td></td>
<td>Passpoints</td>
<td>3.55 (2.09)</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>5.25 (1.71)</td>
</tr>
</tbody>
</table>

6.8.4. Click-points

A PHP function called rand() was used as part of the systems script to generate random click-points for this experiment. The analysis of these clicks shows no hotspots in any of the employed images as seen in Figure 6.13 where the probability distribution is illustrated. In the illustration each image is divided into 36 squares of 27x27 pixels and the probability distribution of each square is calculated. The illustration shows no significant difference above the ‘equal distribution’ line. The equal distribution line represents the case if all clicks are distributed equally among the 36 squares of each image.
Figure 6.13: Line chart illustrating the probability distribution of 100 clicks. As such, $P(p01x)$ is the Probability of the image: p01x. The ‘Equal Distribution’ line represents the case if all clicks are distributed equally (all squares have the same probability).

6.9. Discussion

Click-based systems have security privileges over other authentication schemes (e.g. larger password spaces). However, usability is a concern as discussed earlier in this chapter. Three main problems were particularly addressed and a new application of HybridPass called ‘CCBS’ is developed with the objective of overcoming these problems as predicted in the hypothesis of the chapter.

Similar to Passpoints, CCBS users found it easy to register a new account. In CCBS, only one user needed a second attempt to register a new account because the click-points entered initially for the account were not confirmed successfully during the second step of the registration phase. Nevertheless, the total number of attempts to register a new account in CCBS was significantly less than that of Text. This implies that the CCBS scheme succeeded to maintain high registration rate while enforcing system generated passwords on users. However, CCBS users required more time to register a new account (an average of 112 seconds compared to 64 seconds in Passpoints). This difference is acceptable considering that 64 seconds in Passpoints is the time required to input 5 click-points, while 112 s in CCBS included time to enter the user ID and full name.
in addition to selecting 5 click-points. Further, in the questionnaire of user perceptions, participants agreed that it did not take long to create a password.

In the learning phase 85% (17 users out of 20) of the CCBS users input their passwords 10 times without any errors and the remaining 15% made a single error. In comparison, 70% of the Text users made no errors and the remaining 30% completed the learning phase with a maximum of 2 errors. In Passpoints, 40% made no errors, 30% made 1 or 2 errors and the remaining percentage made more errors. As such, 20% made between 17 and 20 errors. These results imply that the technique used in CCBS was very successful in guiding the users to select the correct click-points. The effect was also reflected in significantly shorter total learning time compared to Passpoints. Users of the Text scheme were familiar with entering this kind of passwords hence they made fewer errors compared to Passpoints. In addition, Passpoints lacked a technique to guide users towards the exact click-point they were intending to select. The authors of the comparative study concluded that the most common problem in Passpoints was clicking close to the correct click-point but outside tolerance which supports our explanation of the problem. CCBS resists this via a confirmation message sent to the user before selecting a particular click-point although tolerance used in CCBS is 9x9 that is less than that of Passpoints 20x20.

User perceptions after the learning phase show the Text scheme to be significantly faster. This is not surprising since users are more familiar with it. Selecting a letter on the keyboard is a daily practice for most if not all the users, while using the mouse to select a click-point inside an image is a new and a slower practice as seen from the experiment data. Users agreed that password input was easy and that it got better with practice despite the scheme being used. The users of Passpoints however significantly disagreed more that they were able to input their passwords correctly. This is a reliable perception since it is corresponding to the analysis of the number of incorrect submissions in Passpoints compared to the other schemes as discussed earlier.

In theory, cued-recall passwords should be easier to remember than pure-recall schemes since they provide hints to trigger user memory. However, password retention in Passpoints was similar to that of the Text scheme which could imply that password retention in click-based systems is similar to that of text passwords. Our results challenge this indication as there were no incorrect submissions using CCBS in R1. In R2, the number was significantly less than the other schemes and in R3 it was also less but with no significant difference. Hence, a good application of click-based schemes (e.g. CCBS) can maintain better retention than text passwords.
Time analysis during password retention shows that CCBS users were the slowest to submit their passwords. The difference was significant across R1, R2 and R3 whilst users of the Text scheme were the fastest. Hence, the technique used in CCBS to guide users to correctly select their click-points to login is significantly increasing retention time. The time difference is not due to mouse movement or familiarity with the system since Passpoints shares these characteristics, but the time must have been used to recall the correct click-points. As such, there are three main cues toward the correct click-point: first the user must retrieve the correct image, and then observing the image to locate the correct area, finally moving the mouse until the right click-point is hovered and the correct textual cue is displayed by the system. Data analysis for CCBS also shows that retention time in R1 was less than that of R2. The reason must be that R1 took place shortly after creating the password but then time in R3 was less than R2 suggesting that users are becoming more familiar with the password hence faster to login. This scenario applies to Text and Passpoints too. In conclusion, CCBS password retention is slower, approximately 1 minute in R2 and R3 but we argue that this time is acceptable since time reduces with practice, noting that time in R1 was 27 seconds. In addition, slower but acceptable password retention time with less number of incorrect password submissions conclude more usability than a scheme were users often do multiple password resetting or go through password recovery procedures due to retention problems.

Perceptions after the retention phase agree more that text passwords then Passpoints were faster to input. All schemes were easy to use but participants agreed more that click-based systems included better cues and password rules to create then recall the passwords. This is not surprising since they are cued-recall authentication schemes.

Analysis of the click-points used in this experiment shows no hotspots. This is not surprising since they were automatically and randomly selected by the system and not by the end users.

6.10. Validation of hypotheses

On applying CCBS, The hypotheses were:

1) ‘A uniform distribution of click-points (no hotspots) will be achieved via system-generated passwords while maintaining memorability’.

Hypothesis supported: probability distribution of click-points is illustrated in Figure 6.13. No hotspots emerged. Using system-generated passwords (click-points) did
not affect user memorability. Password retention in R1, R2 and R3 for CCBS was better than the comparative schemes, hence memorability was maintained.

2) ‘Users can accurately select the intended click-points to input their VPs hence the number of incorrect password submissions will significantly reduce’.

Hypothesis supported: the number of incorrect password submissions in the learning phase using CCBS was less than the two other schemes. The difference was significant with Passpoints. Further, in the retention phase there was no incorrect submission in R1. The number was also less in R2 and R3. The difference in R2 was significant.

3) ‘Success rates of the VP retention in CCBS will be significantly higher than the comparative schemes: Passpoints and Alphanumeric (text password) on the long term’.

Hypothesis supported: the illustration in Figure 6.11 shows success rates for Passpoints and Text to be 85% and 90% respectively in R1. Success rates were then decreased to 55% in R2 and R3. Meanwhile, success rates using CCBS were 100% in R1, 80% in R2 and 85% in R3. CCBS rates in R3 were identical to that of the instant retention using Passpoints in R1.

6.11. Potential enhancements (shoulder-surfing and keyboard support)

Shoulder-surfing is a concern for click-based systems since password submission is exposed on the monitor. Text passwords are less vulnerable to shoulder-surfing since password submission is performed with a keyboard which is less visible to other individuals. Further, users can use their hands to prevent people from looking at the typing area. Hence, it is a privilege to develop a method for click-based schemes to submit click-points to the system using the keyboard.

A click-based system with keyboard support is possible with CCBS. This can be achieved through adding a unique code at the beginning of every textual cue. The format will be: ‘click-point code: textual cue’, as such if the code is A3B, the result is: ‘A3B: Venus is the Roman goddess of love and beauty’. After locating the correct click-point, the user can choose to either click or else type the relevant codes in a password field to submit the password using the keyboard. Nevertheless, click-points can also be highlighted using the keyboard alone (using the left, right, up and down arrows to highlight them and the ‘Enter’ key to select). This is possible in CCBS because the
coordinates of each click-point is consistent in the image. Whilst in Passpoints, the coordinates/area of a click-point is calculated after a mouse-click event (not consistent).

6.12. Conclusion

The results of this experiment give empirical evidence to the usability of adopting system-generated passwords in a click-based system such as CCBS. This approach is critical to resist hotspots and increase the effective password space. It was found that using system-generated passwords in CCBS did not have negative impact on password retention in comparison with Passpoints and text passwords.

Researches such as (Wiedenbeck et al., 2005b) and (Wiedenbeck et al., 2005c) suggest that most of the incorrect password submissions occur because users fail to click inside tolerance around their original click-point. This is reasonable since there is no cue to confirm if they are selecting the intended click-point or the one next to it. CCBS used textual cues and successfully increased the accuracy of click-points selections, in addition, supported password retention. Hence, CCBS has two cues to recall each click-point instead of one.

Users found CCBS to be easy to use but password submission was slow especially if the password is being retained after a week’s time. Regular users of the system are expected to be faster in retaining their password but this however does not change that CCBS is slower than Passpoints or a text password scheme. It is a challenge for end users to perform fast with CCBS compared to text passwords, a scheme they have regular experience with.
Chapter 7 Protection against Phishing Attacks

Phishing or web spoofing is a social engineering technique where technology is employed to illegally obtain sensitive information and user credentials such as usernames and passwords. In a phishing attack, users of a particular service can be asked to sign-in to a clone of the original authentication system connected to a masqueraded domain name. Nevertheless, an advanced version of the attack (known as Pharming) could exploit the Domain Name Service (DNS) responsible for resolving domain names into their real IP addresses to forward the visitors of a legitimate domain name to a fraudulent server to recover their credentials. The phishing problem is widely recognised now due to the number of victims and financial loses, which have risen to more than $3.2 billion in 2007. The same survey shows that approximately 3.6 million US adults lost money in phishing attacks, up from 2.3 million adults the year before. (Gartner, 2007)

This chapter investigates and addresses the problems and limitations of the existing anti-phishing solutions and concludes answers to why phishing is still working. Further, it presents a plan of action to develop and maintain better resistance by developing user authentication schemes using VPs/challenges. Two schemes are then demonstrated to protect inexperienced users against web spoofing, one is based on the PassMark scheme provided by RSA as part of their identity protection suite and another one is based on the HybridPass scheme presented in this thesis. Further, followed by an evaluation and security analyses in opposition to a number of authentication systems currently used in HSBC, Lloyds TSB, Bank of America (BoA), Alliance & Leicester (A&L) and other banks.

Some of the recent attacks incorporated a script enabling a phishing server to act as a Man-in-The-Middle (MiTM). Based on the findings and literature analysis of this research all authentication methods used online today are vulnerable to this attack including the 2-factor authentication schemes (Krebs, 2006). MiTM-phishing potential solutions are analysed followed by a proposal of an approach to resist MiTM-phishing scripts.
7.1. Insights into phishing

Research work on online trust such as (Ang et al., 2001), (Jarvenpaa et al., 2000) and (Fogg et al., 2003) developed guidelines and models for web designers to earn consumers trust on the internet. A study of 2,684 participants to analyse how consumers evaluate the creditability of a website found that the majority of people focus on the layout’s design (Fogg et al., 2003). However, a fake website can be designed to follow these rules as well. Even worse, in a phishing attack the whole design and content of a trusted website is copied to launch an identical clone of the system. These spoof pages are so well designed that many consumers are convinced to share their information. A recent example is when hotmail users were urged to change their passwords after the appearance of a list of more than 10,000 hotmail passwords due to phishing attacks (Raywood, 2009a). In addition to capturing sensitive details, spoof pages can be used to trick users into downloading malicious software or following instructions posted by the attacker.

7.1.1. The problem

There has been a dramatic increase in the financial losses caused by phishing attacks. In the US the number reported in 2004 was $500 million (Leyden, 2004), increasing to over $2.8 billion in 2006 (Gartner, 2006) and $3.2 billion in 2007 (Gartner, 2007). The Gartner survey also shows the number of victims to be around 3.6 million US adults in 2007, rising from 2.3 million the year before.

All e-commerce customers are potential targets for phishers. The number of people receiving phishing emails doubled between 2004 and 2006 (Gartner, 2006). As such the number of reported phishing sites is mounting. There were 2870 sites as of March 2005, with a 28% increase every month since July 2004 (APWG, 2005). Then as of the second half of 2009, there were 126,697 unique phishing attacks reported worldwide using 28,775 unique domain names (Aaron and Rasmussen, 2010). Most of the recent attacks are well planned and not random, the Anti-Phishing Working Group (APWG) identified ‘Avalanche’ as a group responsible for two-thirds of all the identified phishing attacks during 2009.

Despite the existing anti-phishing solutions, fighting web spoofing remains difficult due to technical faults or the human factor (users’ lack of knowledge or lack of attention) as discussed in Sections 7.2 and 7.3.
7.1.2. Sample attacks

Phishing attacks may take several forms but the most popular scenario starts with an email sent to the users pretending to be from one of their service providers. Figure 7.1 illustrates a phishing message targeting PayPal users.

Dear PayPal member,

You have added albertbussines@yahoo.com as a new email address for your PayPal account.

If you did not authorize this change, check with family members and others who may have access to your account first. If you still feel that an unauthorized person has changed your email, submit the form attached to your email in order to keep your original email and restore your PayPal account.

If you are using Internet Explorer please allow ActiveX for scripts to perform all data transfers securely.

Thank you for using PayPal!

Please do not reply to this email.
This mailbox is not monitored and you will not receive a response.

---------------------------------------------------------------------------------------------------
Copyright © 1999-2010 PayPal. All rights reserved.

Figure 7.1: Sample of a phishing message targeting PayPal users.

The email's header had the following details:
- Apparent sender: PayPal.
- Return address: Paypal <noreply@paypal-security.com> (this is where the message apparently comes from and is easy to fake).
- Title: Your Paypal e-mail adress was successfully updated.

After the message body the following paragraph is posted followed by a data form with a submit button:

‘To confirm that you are the right owner, please complete as accurately as possible the following form’.

The form is designed to capture the following information from users: email address, PayPal password, full name, bank name, card type, expiration date, card verification number (cvv2), street, city, state, country, zip code, mother median name, telephone, social security number and date of birth.

Other phishing messages are written to convince people to following a link directing them to a clone of the service provider’s (e.g. a bank) website to sign-in. After logging
their credentials, an error message is displayed and people are forwarded to the bank's legitimate authentication page. Many users would accept what happened assuming they have entered a wrong password.

During August 2010, an extensive number of phishing emails were sent targeting the customers of different British banks. Figure 7.2 illustrates one of these messages. The email had the following details:

- Apparent sender: HSBC.
- Return address: HSBC UK <notice@hsbc.co.uk> (this is where the message apparently comes from and is easy to fake).
- Title: Dear HSBC Bank UK Cardholder.

![Phishing Email Example](image)

**Figure 7.2:** Sample of a phishing message targeting HSBC users.

The ‘Log on’ link coming with the email forwards people to a clone of the system hosted at: http://58.30.143.198/mobile/online.hsbc.co.uk/. Figure 7.4 illustrates the spoof while Figure 7.3 illustrates the genuine HSBC website. The spoof is designed to be a clone of the original system.
Figure 7.3: Snapshot of HSBC (UK) Internet Banking Personal logon page.

Figure 7.4: Snapshot of a spoof page of the HSBC (UK) logon page.

7.1.3. Key facts of recent phishing attacks

Investigating a sample of recent phishing emails and websites submitted to online phishing databases such as (PhishTank, 2010) helps to identify the main characteristics of a phishing attack. These are described in the following steps of a typical phishing scenario:

- A phishing message is sent to a list of targeted emails.
- The message’s title is written to capture users’ attention so it usually includes words like alert, update, action required, important notice etc.
- The sender’s address is filled with a forged email belonging to the service provider it claims to come from or with a similar address when the target is required to reply to the email.
- The message body directs users to send their information using an attached form requesting personal information or via a link to a clone of the system.
- The links included in the message are posted as clickable images or HTML hyperlinks to hide the real address from the user (e.g. `<a href='http://58.30.143.198/mobile/online.hsbc.co.uk/'> http://online.hsbc.co.uk/ </a>`).
- The message style imitates the design of the original service provider and includes logos and email signatures to cause visual deception to users, hence giving them a false sense of security.
- Since the message is sent to a large number of users, it is usually impersonalised and includes generic greetings. However, some of the recent phishing scam greeted users by their names or email addresses.

- A large portion of the reviewed phishing messages included spelling mistakes in the title or body of the emails.

- The phishing site is a partial clone of the system. Attackers copy the exact HTML tags of useful pages (e.g. the user authentication pages) to capture sensitive information, but not everything. Instead, they include links to the original system if the user tries to navigate.

- The domain name of the targeted service is used as part of the spoof address to trick people, mostly as a sub domain (e.g. http://paypal.com.pltx.info/).

- After logging the victim’s response, a typical phishing site generates a feedback and forward to the original system.

- Frames and pop-ups are used at times to place a spoof form on a legitimate website instead of designing a clone of the system. However, this technique is less popular.

7.2. Analysis of solutions

Due to the growth of well planned phishing attacks and the accompanying consequences on the use of online services, a number of responses emerged to control the problem, ranging from social and legal responses (training and legislation) to technical proposals.

This section demonstrates the technical solutions available in the literature and analyses their strengths and weaknesses. As such they can be divided into three categories: SSL/TLS and third-party certification, anti-phishing tools and finally all the schemes extending the traditional user authentication interface where user IDs and text passwords are employed.

7.2.1. SSL/TLS and third-party certification

Secure Sockets Layer (SSL) is the predecessor of Transport Layer Security (TLS); these two protocols utilize Public Key Infrastructure (PKI) to secure transactions between clients (web browsers) and servers (domain names). In addition to protecting sensitive data such as passwords and credit card numbers via encryption, they authenticate the server through a certificate issued by a trusted third-party (TTP) also known as
Certificate Authority (CA), such as Thawte or VeriSign. The certificate is used to verify whether the domain name truly belongs to the server the user is connecting to, hence it resists phishing attacks. When SSL/TLS is used, the browser shows a small ‘lock’ icon to indicate that the connection is secure. Clicking that icon gives authenticity and encryption details as seen in Figure 7.5. Browsers verify that the certificate is valid, signed by a trustworthy CA and contains the domain name as a proof of identity. If a certificate error occurs, the browser warns the user about the problem. For more technical details about SSL/TLS refer to (Rescorla, 2000).

Figure 7.5: In Google Chrome, clicking the ‘lock’ icon indicator of SSL/TLS opens a security information window. Details provided include: website identity verification, CA name and encryption type.

Despite all security measures in the design and implementation of SSL/TLS, there are challenges and problems, some of these are described below:

Certificates can be exploited. For instance, (Sotirov et al., 2008) shows how a vulnerability existed in PKI that helped to create false SSL certificates to impersonate any website on the internet. The exposed vulnerability was based on a flaw in the MD5 hashing algorithm used to create unique digital signature to prove the certificate’s integrity and authenticity, but the flaw made it possible to create two certificates with the same hash value. Thus, an attacker could create a fake duplicate of a genuine site’s certificate to be used for a phishing clone. Likewise, the security holes in some browsers could allow a malicious clone to use the certificate of another website (Miller, 2004). Other reported incidents include VeriSign’s erroneous issue of Microsoft code-signing certificates to a deceiver, the attacker in this case could sign code using the name
'Microsoft Corporation (Microsoft Security Bulletin MS01-017, 2003). Nevertheless, and according to a Symantec research (Westervelt, 2009), the latest cybercrime techniques include placing phishing pages onto legitimate servers to obtain valid certificate information.

*Users do not have the required knowledge or skills* (Dhamija et al., 2006) (Wu et al., 2006) (Downs et al., 2006). The human factor can be exploited in various ways to perform effective phishing attacks because users are not well trained in internet security. For example, most of them do not understand how digital certificates work, instead their anti-phishing education involves locating the ‘lock’ icon as an indicator that the page is secure to submit sensitive data. Consequently, the complex security measures are represented by a small lock icon for many users. Accordingly, more malicious sites are designed with SSL to look more trustworthy. The Netcraft Toolbar Community reported more than 450 phishing sites using SSL (Miller, 2005). In response to that, the Extended Verification (EV) SSL Certificate (VeriSign Inc, 2010) was introduced. New browsers such as IE8, Google Chrome and Firefox 3 will recognise an EV Certified website by changing the colour of the address bar to green. An EV Certificate is superior because it validates the owner's details in addition to the domain name, while the regular SSL Certificate is issued after a domain validation only. Further, it is exclusively issue by a number of trustworthy CAs. However, do users take warning messages seriously? An incident occurred to New Zealand’s BankDirect when their online banking certificate accidentally expired. During the 12 hours until the problem was solved, about 300 customers were alerted with a security warning. Only one user stopped, while the rest of the 300 dismissed the warning and signed in as the server logs prove (Miller, 2005).

Users can also be fooled by the syntax of the domain name, for instance, a long domain name starting with http://paypal.mypayaccount.com or http://www.paypai.com might go unnoticed by PayPal members. In addition, an analysis of the Anti Phishing Working Group (APWG) archive shows that many users cannot reliably distinguish hyperlinks from images of hyperlinks (a legitimate written link pointing out to a different target) or a web page content from the body of the browser (Dhamija and Tygar, 2005).

Another type of third party authentication is done through site seals. These are linked images posted on a website to prove that it has been verified by a trustworthy authority, e.g. (Thawte Trusted Site Seal, 2010) and (DigiCert® Secure Site Seal Authentication, 2010). An obvious challenge with site seals is that people could use them to verify a new website, but it is a challenge expecting them to perform this test with every sign-in attempt. Nevertheless, site seals can be easily emulated and spoofed.
7.2.2. Anti-phishing tools

They are primarily designed as toolbars and add-ons/extensions for internet browsers to identify fraudulent websites. In addition to verifying website certificates, a variety of other methods are used including black lists of known spoof URLs, white lists of known legitimate domains and algorithms to calculate the probability of a website to be a spoof or not. Examples of these are:

Netcraft Anti-Phishing Toolbar (netcraft.com, 2010), relies on a community-powered scheme to build a list of newly discovered phishing sites. The list is actively updated by experts and highly trusted users (to lower the potential of false positives) to block or flag spoofed URLs before other members access them. The scheme is also designed with the capability of:

1) Recognising susceptible URLs aiming to mislead users (e.g. replacing the ‘o’ in amazon with a zero; www.amaz0n.com).
2) Enforcing the display of browser navigation controls to defend against pop ups’ attempts to hide them.
3) Providing an enhanced display of additional information about the site’s hosting location. This is helpful, for instance a local British bank’s website is unlikely to be hosted in Russia or China, if found so, the reason should be investigated.

SpoofGuard (Chou et al., 2004), is a browser plug-in that alerts users when a suspected spoof page is identified after inspecting its domain name, URL, hyperlinks and images. For example, a domain such as http://amazon-tradewithus.com with Amazon logo will have a higher probability of being a spoof than a website without these characteristics. In addition, SpoofGuard analyse the browser’s history to make better decisions, for example it checks if a particular domain was visited regularly before and whether the referring page was from an email, a bookmark or another form of links.

Other toolbars performing similar tasks include: (EarthLink, 2010), (SpoofStick, 2005) and many others.

While these tools are designed to combat phishing, the effectiveness varies from one tool to another. (Zhang et al., 2007) experimented with 10 popular anti-phishing toolbars using newly discovered phishing URLs mixed with legitimate ones. The result shows that SpoofGuard did well at identifying spoof websites, but it also erroneously identified a large fraction of legitimate sites as spoof. This is a major usability problem which could result in removing the tool immediately by the vast majority of users and if not removed,
users might confuse and accept fraudulent sites since the tool has a high rate of mistakenly identifying genuine sites as spoof. The study also concluded that TrustWatch, Netscape 8 and eBay toolbars correctly identified less than half of the fraudulent URLs, while McAfee SiteAdvisor failed to identify any of them. However, Netcraft, Google, EarthLink, Cloudmark and IE7 performed better by identifying most fraudulent sites correctly with few false positives, but they still missed more than 15% of them.

From the example above, it is clear that some of the anti-phishing tools can help to control the phishing problem better than others. A good tool must be one step ahead of phishers who will be testing their codes against widely available protection techniques. As such, many if not all the techniques used to identify spoof websites can be defeated by attackers in one way or another. For example, identifying website logos as described in SpoofGuard is accomplished by a comparison test that can be fooled if the image is slightly changed (Chou et al., 2004). Reporting spoof URLs by expert users only as adopted by the Netcraft community restricts the speed of identifying new phishing sites. Time is important since the average time for a single phishing attack link to remain online is between 24 and 36 hours only, as of the second half of 2009 (Aaron and Rasmussen, 2010). However, if all users are involved, the service could be exploited to perform Denial of Service (DoS) attacks by intentionally reporting legitimate URLs. Another technique adopted by many tools is to display extra details about the site’s host, but despite the fact that many users aren’t skilled to perform this check, it is unlikely for users to investigate these details continually for a regular login page since they favour convenient over security, for instance they continually ignore security warnings generated by their anti-phishing tools (Dhamija et al., 2006) (Wu et al., 2006) (Downs et al., 2006). Other challenges to the use of anti-phishing tools include privacy concerns since some techniques depend on the user’s browsing history, as mentioned earlier. Nevertheless, these tools require installation, which implies that users in any case may not benefit from them on public and shared computers.

7.2.3. Extending user authentication mechanisms

The traditional user authentication approach of user IDs and text passwords can be extended to resist phishing attacks. Different solutions have been proposed, which can be grouped into two categories:

One-time passwords, the resistance mechanism in this kind of system relies on restricting the life cycle of the password to a single use only or a certain amount of time, hence if a phishing attack succeeds, the stolen password expires. Generating secure
one-time passwords is achieved with the use of tokens; something the user has, adding a new authentication factor to the process. (RSA SecureID) is an example for such two-factor authentications in which the password is changed every 60 seconds. A separate device such as the mobile phone can also be employed either to host an application responsible for generating the passwords or by requesting/receiving it via Short Message Service (SMS). Some banks such as (finextra.com, 2005) have a paper-based scheme in which every user receives a list containing the one-time passwords in advance to use them in order. The challenges of adopting these solutions include cost and the usability concerns of having to carry many tokens everywhere; if the token is not available, the user cannot use the service. To avoid using tokens, an approach is to require part of the password to be used at a time instead of the full password string. For instance, Lloyds TSB (lloydstsb.com, 2010a) internet banking service requires 3 random letters from a longer memorable information in addition to the traditional text password at every sign-in attempt. The Déjà Vu VP (Dhamija and Perrig, 2000) scheme requires end users to recognise part of their portfolio at a time. While this avoids using tokens, the password space is relatively small. A successful phishing attack could reveal a large portion of the password.

Server verification through shared secrets, in which the implementations can be divided into two sub-approaches to verify the server’s identity through shared secrets.

In the first approach, the local system (e.g. browser) is completely involved to indicate the server’s authenticity to users. The Petname Tool (Close, 2009) is an example for this in which the user customises each website with a name. The tool associates the selected name with a hash of the public key of the site to display it every time the website is visited (Waterken Inc, 2004). The sudden absence of this corresponding name indicates that the visited page is a spoof. While this simplifies the recognition of websites by their Petnames, it is unclear if the usage of Petnames is worth increasing the memory load on users while the tool can simply identify a visited website with two values: trusted or not trusted. Since the secret is shared with the local system, the user is unprotected on other shared and public computers. Nevertheless, the tools become ineffective if the user is not motivated enough to customise these websites. Other examples where the local system is involved include Dynamic Security Skins (Dhamija and Tygar, 2005), Passpet (Yee and Sitaker, 2006) and SRD (Ye et al., 2005).

The second approach is not dependent on the client’s system and does not require installing any add-ons or browser extensions because the secret is shared between the user and the server exclusively. An example of this is PassMark (PassMark Security,
2010), also called SiteKey as in the implementation used in Bank of America (bankofamerica.com, 2010b), or simply the ‘image/phrase combination’ as in Alliance & Leicester (alliance-leicester.co.uk, 2010). In addition to the user ID and text password, a secret (PassMark) is shared with the server. The PassMark consists of: an image (chosen to be easily recognised by the user) and a title (any passphrase to accompany the image). A unique PassMark is shown to each user during authentication to verify the server’s identity.

A comprehensible advantage of the PassMark scheme over the anti-phishing tools is that users are not asked to install any software to their computers and there is not a false positive problem. Further it maintains server authenticity to users signing from new or shared computers. However, fraud vulnerabilities exist; more analysis and proposed solutions are presented in Section 7.5.

7.3. Why phishing still works

The demonstration of the anti-phishing schemes presented in section 7.2 clearly addresses the human factor as the weakest part in the resisting procedure. Even when the technical solution successfully defines a webpage as a definite spoof, reports suggest that security alerts are dismissed or ignored by the vast majority of users (Miller, 2005) (Dhamija et al., 2006) (Wu et al., 2006) (Downs et al., 2006). Reasons resolved from the examples discussed earlier dissect the human factor into the following elements:

1) Knowledge: users are not security specialists. Many of them are not acquainted with technical details and do not understand how the service or the anti-phishing scheme works. Figure 7.1 illustrated an example of a phishing attempt designed to exploit users vulnerable to this attack due to their lack of knowledge. Hence, sensitive details can be obtained through second channels such as online forms and emails.

2) Alertness: even if users are trained on a security scheme, they are not conscious at all times to respond respectively to security warnings. Security indicators might not be recognised. For example, the absence of the server image in the PassMark experiment (Stone, 2007) was not noticed as discussed earlier.

3) Motivation: users are not self-motivated to continuously examine security. For example, while the legitimacy of a site seal can be verified via a third-party
website, a spoofed seal could pass unchecked since it is inconvenient for users to examine it every time they use the service. Another example was the Petname tool; to work, it requires end users to manually customise e-commerce sites with identifiers, memorize the names given and recognise them in the future. Only highly motivated users might customise all websites.

4) Concern: lack of concern about warning messages. False-positive warnings can be a reason for that as genuine pages can be erroneously highlighted as a spoof by some security tools or correctly highlighted with other types of warning message. It is observed that when more alerts are generated, users start ignoring them at a higher rate.

These elements are grouped and referred to hereafter as the human factor. In addition to the human factor, phishing attacks have good chances to succeed since a large portion of end users do not have a good internet security tool to protect their online activities against identify spoof and malicious software. The (Gartner, 2007) survey shows that approximately 11% of online adults say they do not use any security software (e.g. an antivirus or anti-spyware products), and another 45% only use basic antivirus software available for free from the internet.

7.4. Towards a better anti-phishing approach

Further to the human factor, the majority of the proposed anti-phishing solutions require installation on the client machine, which implies that neither the user nor the online service provider can maintain phishing resistance on other shared computers. Some schemes employ tokens to generate one-time passwords. However, these are not widely adopted due to cost and usability concerns. In conclusion, the solution should be computer independent; therefore it would not require software installation or hardware tokens as a pre-requisite to use the service.

A good anti-phishing approach must consider users who can be vulnerable to any of the elements addressed in Section 7.3. The solutions are discussed bellow supported by examples from the work developed in this chapter.

1) The ‘Knowledge’ problem: basic understanding of how to use the service is essential. The attack in Figure 7.1 cannot be stopped if the user believes the message and decides to reply and share bank details and other sensitive information. To control this problem user training and anti-fraud alerts must be
provided, but in addition we argue that using a relatively hard to share password (hard to share outside the original system) is a mandatory part of the solution to overcome the knowledge problem. While the character-based secrets can be written and easily shared through forms, emails or phone, visual challenges such as the mouse clicks in the HybridPass system are hard for users to share through email and online forms.

2) The ‘Alertness’ problem: the unconscious condition of a knowledgeable user can be highly and effectively alerted if the system is designed to technically stop or fail before recovering the full user credentials to a spoof website as demonstrated in Figure 7.6 and then discussed further in 7.7. Table 7.2 shows how the enhanced version of HybridPass will stop the authentication process at least 4 or 5 times before retrieving the full user credentials.

3) The ‘Motivation’ problem: this can be solved through using a shared secret to authenticate the server (e.g. user portfolio required to obtain the VP) as part of the user authentication process. This way, server authentication is performed at every login attempt as a genuine part of the user authentication process and not as an optional step that can be neglected by unmotivated users.

4) The ‘Concern’ problem: the solution employs the server’s alphabet (shared secret) as a requirement to obtain a valid password (e.g. refer to Figure 7.6). This will have zero false-positives because it is designed as part of the user authentication process. The user will not be able to proceed at the absent of the server’s alphabet that is used to define the server as legitimate as well as obtaining the password. While a tool can suspect a legitimate server or display a warning message that can be ignored by the user.

A simple example of how to make user authentication fail on a spoof page can be seen in the development of HybridPass (presented in Chapter 4) where the VP is entered based on the alphabet (images) received from the server. A spoof server cannot initiate the correct alphabet and without them, users are unable to respond, hence stopped from entering their VPs as illustrated in Figure 7.6. The figure shows that to recover the full access credentials, the attacker should perform two successful phishing attacks; one to recover the text password and then obtain the relevant server’s alphabet offline and a second attack to obtain the VP (user clicks) if the text password is not changed in between. Users who are well informed about security, can change their text
passwords at events where the server fails to provide them with the correct alphabet, in this case, the attack must restart to recover the text password again.

Figure 7.6: In HybridPass, the design of the scheme forces user authentication to fail before recovering the VP during the first attack.

Analysing the solutions from the literature as demonstrated in Section 7.2 shows good potential for the PassMark scheme. However, it has security flaws and does not consider the human factor properly. The following section will discuss this in further detail and propose solutions to increase security in PassMark-based implementations without affecting their usability. This will also show how an existing scheme can be developed to consider the four main elements of the human factor in a good anti-phishing approach as defined earlier.

7.5. Developing the PassMark scheme

The scheme presents an image and a title (any passphrase to accompany the image) called PassMark. A unique PassMark is shown to each user during user authentication to verify that the server is genuine since a spoof page cannot display the correct PassMark.
The Bank of America’s version of the scheme is called SiteKey. During the sign-in session users go through one of the following two scenarios:

1) Client’s computer is recognised by the server: The server uses cookies or Flash objects to recognise and trust the computer from which the user has signed in before, in this case, the user is asked for his user ID and in response to it, the server will display the shared secret (PassMark) for the user to recognise and authenticate the server in his mind, then enter his text password with confidence. If the PassMark is not available or incorrect, the site should be considered a spoof and users are expected to terminate the login attempt.

2) Client’s computer is not recognised by the server: In this case, the server will ask one challenge question (e.g. in what city were you born?) to verify the user’s identity before displaying the shared secret.

Another bank using the scheme is Alliance & Leicester. During the sign-in session users enter their ID and in return the server responds with the relevant PassMark. After recognising the correct PassMark, the user is asked for a 5-digit PIN.

This section identifies two fundamental flaws in the PassMark scheme and present applicable solutions. More fraud vulnerabilities are reported and discussed in (Youll, 2006).

7.5.1. PassMark recognition

The PassMark scheme relies on the recognition of the server’s image, but will users terminate their authentication process at the absence of the shared secret? The answer comes from an experiment designed by a joint team of researchers from Harvard and M.I.T. where a number of Bank of America customers were gathered to conduct a routine online banking activity on a modified system where the images/titles are absent. The result shows that 58 out of 60 participants who proceeded into the study entered their passwords anyway (Stone, 2007).

This major flaw (illustrated in Figure 7.7) can be solved if the scheme is redesigned to involve the server’s image-recognition step as a genuine part of the user authentication procedure, so that if the image-recognition is ignored, user authentication consequently fails. This way, users are forced to be conscious about the server’s alphabet every time they use the system, this can be achieved through implementing one of the following approaches:
1) The correct image is displayed among other images in a grid and the user is asked to perform an action (e.g. a mouse click) to select it correctly. This way the users are forced to perform the recognition task every time they sign-in and should notice the absence of the PassMark on a spoof page. However, if the text password is submitted anyway, the attacker will be challenged to guess the correct image while signing in. If not guessed correctly, the account is locked after a minimum number of one or two failing attempts. The tolerance can be reduced to one or two attempts only since image recognition is the simplest memory task in comparison to pure-recall and cued-recall tasks. Implementing this idea in PassMark enhances the security of the system without affecting the usability since users are only asked to confirm a recognition task they were assumed to mentally perform in the original scheme.

2) The image is displayed alone as in the original scheme, but to proceed, the user must locate a specific click-point inside the image using his mouse. Unlike the previous approach, this requires users to memorise a click-point. However, this is a simple cued-recall task consisting of a single click. Since the password/challenge space is larger than the first approach, more tolerance can be given. The account can be locked after 2 or 3 incorrect clicks. It is more difficult for an attacker to correctly guess the click-point.

![Diagram](image)

Figure 7.7: In PassMark, users are asked to enter their text passwords only after authenticating the server by recognising a shared secret/image. However, most users are found to submit their passwords anyway (Stone, 2007) as illustrated.

Further it is possible to combine these two approaches into one. For example, instead of using a bigger grid in the first approach or a larger image in the second to resist
guessing, a few number of small images can be displayed (e.g. 3 images) where the user should recognise the PassMark and select a single click-point. Using any of the suggested approaches makes the server’s authentication through a shared secret a genuine part of the user authentication process. Nevertheless, it prevents an attacker who performs a successful phishing attack against the system to sign-in without a second successful attack to recover the server’s shared secret.

In addition, since the original scheme does not have a technical method to guide users toward recognising the shared secret, service providers such as Bank of America rely on advertising the concept to their customers to show them how important it is to recognise a PassMark/SiteKey. For instance, the following is part of a paragraph from their official website: ‘... when you see your SiteKey, you can be certain you’re at the valid Online Banking website at Bank of America, and not a fraudulent look-alike site.’ (bankofamerica.com, 2010b) however, this statement is not true at all times; the bank in this case is asking customers to trust a spoof page containing the SiteKey despite a possible security warning from the browser or an anti-phishing tool. Users might consider the warning as a false-positive since the SiteKey is displayed on the page. While in the case of combining server authentication with user authentication, advertising the SiteKey recognition is not required. Instead, once the authentication fails, the situation can be described to a staff member and the user may receive appropriate advice on security and phishing.

7.5.2. PassMark protection

The shared secret assures users that the server is legitimate so they could proceed with their authentication, while its absence indicates a problem with the server. A motivated attacker will search for a vulnerability to recover and exploit the shared secret before targeting the user with a phishing attack. Hence, another security concern is regarding the level of security designed to protect the PassMark. To display the server’s secret from an unknown computer (not recognised by the server); a user (or an attacker) should enter a user ID and answer a challenge question in the SiteKey scheme, while in Alliance and Leicester’s scheme the user ID is used alone to protect the shared secret.

User IDs do not add a good layer of protection because unlike passwords, they are exposed (written in the clear), easily learned and users do not periodically change them. Phishing attacks targeting the users of a particular service tend to have lists of user IDs and email addresses (to send mass phishing scams to a particular audience) prior to the attack. This can be achieved through exploiting an existing vulnerability in the service
provider or one of its associates. For example, a recent security hole in AT&T’s website exposed the owners of 114,000 iPad accounts including dozens of CEOs, politicians and military officials (Tate, 2010). The same source suggests that IDs and email addresses of every iPad 3G owner in the US has possibly been exposed due to the nature of the vulnerability and the fact that it has been shared with third-parties prior to closing the security hole. Other methods to collect user IDs include mass phishing and spyware, and unlike passwords, IDs can be searched and collected from computers and other documents such as bank and credit statements because they are stored in the clear. In addition, IDs are not frequently changed, so once they are recovered, they remain valid and can be exploited in attacks for a long period of time.

The Bank of America’s scheme has an additional layer of security to protect the PassMark. At the time of registration, users are asked to create and memorise the answer to three challenge questions. However, this kind of challenge can be guessed by someone close to the target or through an investigation tool (Danchev, 2008) especially with the increasing amount of personal detail shared on social networks such as Facebook, MySpace and many others. While there are no perfect challenge questions, there can be good and bad ones. Unfortunately, the questions presented at (bankofamerica.com, 2010a) include bad or easy to guess questions. Examples of these are:

- In what city were you born?
- What is the name of your first employer?
- What is your favourite hobby?
- In what year (YYYY) did you graduate from high school?

This kind of information can be known to friends and all contacts with access to one of the person’s social networking accounts. Some people publish these details on personal web pages, note that what is shared cannot be completely taken offline again with the existence of services like www.archive.org working on recording and archiving the internet on a regular basis. Nevertheless, some of these details can be collected from online CVs and other career services such as LinkedIn. Figure 7.8 shows more examples of the Bank of America’s SiteKey challenge questions from (bankofamerica.com, 2010a). The problem with challenge questions is that a question that is good for a user can be bad for another who has already exposed the answer to it in a way or another, so it is hard to select good security questions for all users.
What makes the system more vulnerable against guessing attacks is that each login attempt is challenged with one question selected from the three questions created by the user during registration, which implies that guessing any of the three security questions is sufficient to recover the shared secret if the user ID is known, since requesting the sign-in page will always give the attacker 1 out of 3 chances to get the same question repeated again. Recovering the server’s shared secret implies that a phishing attack has a better chance to succeed due to the false trust the spoof page will receive after displaying the correct image to the user.

![Bank of America SiteKey challenge questions](image)

**Figure 7.8:** Bank of America SiteKey challenge questions.

To reduce the risk of using security questions, they must be designed to resist investigation tools and be unknown to family and friends. In PassMark, a good approach to achieve that is through asking users about the shared secret itself since the answer is only known to the account owner who is expected to be familiar with the shared secret (image and accompanying title).

![An example of a shared secret, an image and its accompanying title](image)

**Figure 7.9:** An example of a shared secret, an image and its accompanying title.
Figure 7.9 displays a shared secret in the SiteKey implementation from (bankofamerica.com, 2010b). The possible inquiries on this example can be divided into two categories as demonstrated below:

1) Inquiries on the accompanying title, such as:
   - What are the first and last letters of the accompanying title?
     Answer: tr
   - What is the third and fourth letters of the accompanying title?
     Answer: dd

   In this scenario, the title should have a minimum length to allow a relatively large number of inquiries.

2) Inquiries on the image, examples are:
   - What does your SiteKey image remind you of?
     Possible answers: kids, bear, animal, gift, etc.
   - How do you relate your SiteKey image to your personal life?
     Possible answers: valentine, girlfriend, childhood, play, etc

   Users can be asked to respond with a single word and avoid using the ‘s’ or ‘es’ suffix at the end of their answers, because a user might recall the secret correctly, but forget if it was plural, singular or combined of more than one word. Such rules can be forced with online form validation methods such as JavaScript. While the inquiries on the image cannot be investigated, the number of inquiries is limited (e.g. to 3 question as in the original SiteKey scheme) hence it is not superior to the ‘inquiries on the accompanied title’ approach.

   The advantages of designing the challenge questions to be an inquiry about the shared secret include:

   - The answer is not based on the user’s personal life, choice or public information; hence it cannot be recovered using an investigation tool.
   - The answer is automatically updated each time the shared secret is changed.
   - The full answer is displayed during every successful sign-in attempt. Hence, maintained in the user’s memory.
The number of possible questions is larger in the case of inquiries on the title. Although redesigning the challenge question this way increases the level of security through resisting guessing attacks and investigation tools, the usability challenge is that the legitimate owner will also fail to recover the answer if they completely forget the PassMark. However, as mentioned earlier, the PassMark is maintained in memory with every successful sign-in attempt. More analyses and evaluation debate of the developed PassMark scheme is presented in Section 7.7.

The term ‘PassChallenge’ will be used in the following sections to refer to the newly developed scheme to differentiate it from the original PassMark scheme.

### 7.6. Developing the HybridPass/CCBS scheme

The proposed scheme of HybridPass/CCBS considers the roles of a good anti-phishing approach as defined in Section 7.4 except that it cannot resist multiple attacks in a raw. The user portfolio can be revealed on the second successful phishing attack. And since there is no guarantee that users who fall for a spoof page once will not be fooled again, this section will improve the design of the system to resist more successful attacks and provide an ID protection technique.

In HybridPass the text password is to protect the images used for the VP. The limitations of the scheme include:

- There is no mechanism to protect the user ID.
- If the attacker obtains the text password through phishing, the system assumes the user is going to change it before a second phishing attempt from the same attacker is performed. If the user fails to update the text password in time, the attacker will obtain the correct images linked to the targeted user account and be able to perform an advanced phishing attempt targeting the same user again to obtain his VP and recover the full password.

To resist multiple attacks and provide user ID protection, the following two possible scenarios are to replace the original scheme:

1) Client’s computer is recognised by the server: The server uses cookies or Flash objects (Secure Shared Object) to recognise and trust the computer from which the user has signed in before, in this case, one of the images will be displayed to prove the server’s authenticity before entering the user ID. This step is designed
to provide ID protection for users signing in from their personal computers without recovering all images. After entering the user ID, the rest of the images will be displayed for the user to enter his/her VP as illustrated in Figure 7.10.

![Flowchart](image)

**Figure 7.10**: User authentication when the client’s computer is recognized by the server in the anti-phishing implementation of HybridPass.

To load images and update the forms constantly without reloading the whole page, Ajax (Asynchronous JavaScript and XML) is used to maintain the server-client communication.

2) Client's computer is not recognised by the server: In this case, entering the correct user ID will retrieve one image, and then the server will prompt the user for two random letters from a memorable phrase. The rest of the images are retrieved from the server after entering the random letters correctly.

![Flowchart](image)

**Figure 7.11**: User authentication when the client’s computer is not recognized by the server in the anti-phishing implementation of HybridPass.
7.7. Evaluation and analysis

To evaluate the developed schemes of PassMark (PassChallenge) and HybridPass/CCBS, their characteristics are compared and examined against the original design and other user authentication schemes used in the following banks: HSBC (hsbc.co.uk, 2010), Lloyds TSB (lloydstsb.com, 2010b), Bank of America (bankofamerica.com, 2010a), Alliance & Leicester (alliance-leicester.co.uk, 2010) and Barclays (barclays.co.uk, 2010) for online banking services. Table 7.1 illustrates the login steps adopted by these banks.
Table 7.1: Authentication schemes used in different banks.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Login Screen 1</th>
<th>Login Screen 2</th>
<th>Login Screen 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bank of America (SiteKey)</strong></td>
<td>ID</td>
<td>1 random challenge question</td>
<td>Recognise the server’s alphabet &amp; Enter the text password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(out of 3 questions shared with the system)</td>
<td></td>
</tr>
<tr>
<td><strong>Pass Challenge</strong></td>
<td>ID</td>
<td>2 random letters of a memorable phrase</td>
<td>Server’s alphabet challenge &amp; Enter the text password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(selected using the mouse)</td>
<td></td>
</tr>
<tr>
<td><strong>HybridPass</strong></td>
<td>ID</td>
<td>VP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp; Text password</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HybridPass (developed proposal)</strong></td>
<td>ID</td>
<td>2 random letters of a memorable phrase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp; Text password</td>
<td>(entered with the keyboard)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp; VP</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lloyds TSB</strong></td>
<td>ID</td>
<td>3 random letters of a memorable phrase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Text password</td>
<td>(6-15 characters). (Selected using the mouse).</td>
<td></td>
</tr>
<tr>
<td><strong>HSBC</strong></td>
<td>ID</td>
<td>Date of Birth (ddmmyy).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(entered with the keyboard)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 random digits of a security number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(out of 6-10 digits). (entered with the keyboard)</td>
<td></td>
</tr>
<tr>
<td><strong>Alliance &amp; Leicester</strong></td>
<td>ID</td>
<td>Recognise the server’s alphabet and</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp; 5-digit pin</td>
<td></td>
</tr>
<tr>
<td><strong>Barclays</strong></td>
<td>Surname &amp; Membership number (12 digits)</td>
<td>5-digit passcode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp; 2 letters of a memorable phrase (out of 6 to 8 letters). (Selected using the mouse).</td>
<td></td>
</tr>
</tbody>
</table>
While all these schemes are designed to resist phishing attacks they vary in their effectiveness to protect user credentials from spoof pages. For instance, neither the Bank of America nor Lloyds TSB grant full-access to a single successful phishing attack. However, if one of their customers proceeds on a spoof page; the attacker can recover Lloyds TSB’s account ID, text password and 3 letters of the memorable phrase (if the user’s memorable information is the minimum, i.e. 6 letters, that is half of the memorable phrase); while in the case of Bank of America, the attack will recover the user ID and stop since the spoof page is unaware of the correct challenge question to display.

Table 7.2 provides a comparison between these schemes by demonstrating the number of basic phishing attacks required to recover the full user credentials. While the first expression suggests that they can resist phishing through protecting end users against multiple attacks. In reality, phishing is not the only technique used by attackers to penetrate the system. Hence, numerous factors must be considered to evaluate the schemes such as their resistant to phishing if the user ID is known. As discussed earlier, user IDs of a particular service can possibly be recovered prior to the phishing attack, therefore, the PassMark implementation of Bank of America protects the shared secret with a challenge question, whilst in the case of Alliance & Leicester, the scheme recovers the shared secret if the ID is known, which implies that an attacker with access to the user ID can target the user with a spoof page that can display the correct PassMark and recover the PIN in a single successful phishing attack.

Another technique to recover more information within a single phishing attack is the ‘error-feedback-trick’. This term is used in this research to refer to the situation where a spoof page gathers the first part of a memorable phrase (2 or 3 letters based on the scheme), then display an error asking the user to enter 3 other letters. A user can easily fall for this trick assuming he/she did not enter the correct letters on the first attempt. Implementing the error-feedback-trick gives a deceiver the ability to reduce the number of phishing attacks required to recover the minimum length of a secret memorable phrase to a single attack. Nevertheless, even if the phrase is larger than the minimum length, retrieving 6 letters of memorable information gives the attacker an easy guessing task to recover the remaining letters. Table 7.3 provides a new comparison between the schemes assuming the user ID is known and the error-feedback-trick is implemented. In addition it considers the case where users tend to neglect the PassMark recognition task due to the human factor.
Table 7.2: The number of basic phishing attacks required to recover the full user credentials in different authentication schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Phishing attacks to recover user credentials</th>
</tr>
</thead>
</table>
| **Bank of America (SiteKey)**   | 1<sup>st</sup> attack: recovers the ID and stops since the spoof page is unaware of the correct challenge question to ask.  
2<sup>nd</sup> attack: recovers a challenge question hence recovers the PassMark.  
3<sup>rd</sup> attack: recovers the text password.                                                                                           |
| **Pass Challenge**              | 1<sup>st</sup> attack: recovers the ID and 2 letters of the memorable phrase.  
2<sup>nd</sup> attack: recovers 2 more letters.  
3<sup>rd</sup> attack: recovers the minimum length of the phrase (6 letters).  
If the length is more than 6 letters, the 4<sup>th</sup> & 5<sup>th</sup> attacks will continue recovering the memorable phrase  
Or  
4<sup>th</sup> attack: recovers the PassMark challenge and the text password.                                                                 |
| **HybridPass**                  | 1<sup>st</sup> attack: recovers the text password and stop since the spoof page is unaware of the server’s alphabet (images).  
2<sup>nd</sup> attack: recovers the VP.                                                                                                             |
| **HybridPass (developed proposal)** | 1<sup>st</sup> attack: recovers the ID.  
2<sup>nd</sup> attack: recovers 2 letters of the memorable phrase.  
3<sup>rd</sup> attack: recovers 2 more letters.  
4<sup>th</sup> attack: recovers the minimum length of the phrase (6 letters).  
If the length is more than 6 letters, the 5<sup>th</sup> & 6<sup>th</sup> attacks will continue recovering the memorable phrase  
Or  
5<sup>th</sup> attack: recovers the VP.                                                                                                             |
| **Lloyds TSB**                  | 1<sup>st</sup> attack: recovers the ID, text password and 3 letters of the memorable phrase.  
2<sup>nd</sup> attack: recovers the minimum length of the memorable phrase (6 letters).                                                                 |
| **HSBC**                        | 1<sup>st</sup> attack: recovers the ID, date of birth and 3 digits of the secret number.  
2<sup>nd</sup> attack: recovers the minimum length of the secret number (6 digits).                                                                 |
| **Alliance & Leicester**        | 1<sup>st</sup> attack: recovers the ID and the relevant PassMark  
2<sup>nd</sup> attack: recovers the PIN                                                                                                             |
| **Barclays**                    | 1<sup>st</sup> attack: recovers the surname, membership number, five-digit passcode and 2 letters of a memorable phrase.  
2<sup>nd</sup> attack: recovers 2 more letters.  
3<sup>rd</sup> attack: recovers the minimum length of the phrase (6 letters).                                                                 |

157
Table 7.3: The number of phishing attacks required to recover the full user credentials in different authentication schemes, assuming the user ID is known and the error-feedback-trick is implemented.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Phishing attacks to recover user credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of America (SiteKey)</td>
<td>ID is known, hence the challenge question is known, if the PassMark is ignored due to the human factor (or if it was recovered after guessing the answer to the challenge question):</td>
</tr>
<tr>
<td></td>
<td>1st attack: recovers the answer to a challenge question and the text password.</td>
</tr>
<tr>
<td>Pass Challenge</td>
<td>1st attack: recovers 2 x 2 letters of the memorable phrase.</td>
</tr>
<tr>
<td></td>
<td>2nd attack: recovers more than the minimum length of the phrase (6-8 letters).</td>
</tr>
<tr>
<td></td>
<td>3rd attack: recovers the PassMark challenge and the text password.</td>
</tr>
<tr>
<td>HybridPass</td>
<td>1st attack: recovers the text password and stops since the spoof page is unaware of the server’s alphabet (images).</td>
</tr>
<tr>
<td></td>
<td>2nd attack: recovers the VP.</td>
</tr>
<tr>
<td>HybridPass (developed proposal)</td>
<td>1st attack: recovers the ID and 2 x 2 letters of the memorable phrase.</td>
</tr>
<tr>
<td></td>
<td>2nd attack: recovers the minimum length of the phrase (6-8 letters).</td>
</tr>
<tr>
<td></td>
<td>3rd attack: recovers the VP.</td>
</tr>
<tr>
<td>Lloyds TSB</td>
<td>1st attack: recovers the ID, text password and the minimum length of the memorable phrase (6 letters).</td>
</tr>
<tr>
<td>HSBC</td>
<td>1st attack: recovers the ID, date of birth and the minimum length of the secret number (6 digits).</td>
</tr>
<tr>
<td>Alliance &amp; Leicester</td>
<td>ID is known, hence the PassMark is known and can be used to gain the trust of the user (or if the PassMark is ignored due to the human factor):</td>
</tr>
<tr>
<td></td>
<td>1st attack: recovers the PIN.</td>
</tr>
<tr>
<td>Barclays</td>
<td>1st attack: recovers the surname, membership number, passcode and 2 x 2 letters of a memorable phrase.</td>
</tr>
<tr>
<td></td>
<td>2nd attack: recovers more than the minimum length of the phrase (6-8 letters).</td>
</tr>
</tbody>
</table>
Table 7.3 shows that the majority of schemes used for internet banking despite being designed to resist phishing attacks can still be spoofed to recover user credentials via a single advanced phishing attack. However, it can be observed that the developed proposals of PassChallenge and HybridPass can successfully resist multiple phishing attacks. The reasons are:

- Both schemes include a secret shared with the server (server alphabet). And it is designed as a challenge so that it cannot be neglected due to the human factor.
- Both schemes require 2 letters of the memorable phrase, that is one third of the phrase only, hence the attacker needs to fool a user to re-enter the letters 3 times to recover the minimum length of a password.

If we assume that a particular user responded to a spoof page performing the error-feedback-trick more than 3 or 4 times, the memorable phrase will definitely be recovered. But even in this case, the developed proposals of PassChallenge and HybridPass will still require two successful phishing attacks at least before revealing the full user credentials.

Authentication schemes used for internet banking require two passwords; One is hashed in the database using a one way algorithm to resist database attacks and the other password is stored in the clear for the system to inquire and compare random letters of it to resist replay and phishing attacks. An interesting exception is found in the HSBC scheme since it requires the date of birth in addition to 3 random digits of a security number which implies that their system either does not add encryption to the passwords stored in their database or it uses a reversible encryption method which is essentially similar to storing the passwords as plaintext (Microsoft TechNet, 2005). The same reference from Microsoft suggests if reversible encryption is used for authentication, the system requirement must outweigh the need to protect the passwords. Hence, unlike other banks the HSBC scheme is designed to allow more usability for the price of security. Database attacks can be very dangerous because many of them are not acknowledged and hard to monitor since a typical database may have 15,000 to 20,000 connections per second. Some of the famous database attacks include: exploiting privileged usernames, targeting unpatched database vulnerabilities, SQL injection and stolen backups (Higgins, 2008).

In addition to the database attack, other risks to consider include malware, shoulder-surfing and guessing. Table 7.4 evaluates the effectiveness of the schemes discussed against these attacks.
Table 7.4: The authentication schemes resistant to different password attacks.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Schemes resistant to password attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of America (SiteKey)</td>
<td>Database attack: the text password is hashed in the backend.</td>
</tr>
<tr>
<td></td>
<td>Shoulder surfing: single attack can succeed through observing the keyboard. The monitor can expose the ID, challenge question/answer and the shared secret (image/title). The keyboard can expose all user credentials including the text password.</td>
</tr>
<tr>
<td></td>
<td>Guessing: the challenge questions are vulnerable to investigation tools.</td>
</tr>
<tr>
<td></td>
<td>Malware: keystroke logging recovers the full user credentials.</td>
</tr>
<tr>
<td>Pass Challenge</td>
<td>Database attack: the text password is hashed in the backend.</td>
</tr>
<tr>
<td></td>
<td>Shoulder surfing: multiple attacks since the system inquires 2 letters of the memorable phrase at a time. The monitor can expose the ID, the memorable phrase and the server’s alphabet’s challenge. The keyboard can expose the ID only since the mouse is used to select the memorable phrase and the server’s alphabet’s challenge.</td>
</tr>
<tr>
<td></td>
<td>Malware: mouse clicks and screen must be recorded to recover the full user credentials.</td>
</tr>
<tr>
<td>HybridPass</td>
<td>Database attack: both text and VPs are hashed in the backend.</td>
</tr>
<tr>
<td></td>
<td>Shoulder surfing: single attack can succeed, but both keyboard and monitor must be observed. The monitor can expose the ID and the VP. The keyboard can expose the ID and the text password.</td>
</tr>
<tr>
<td></td>
<td>Malware: Keystrokes, mouse clicks and screen must be recorded to recover the full user credentials.</td>
</tr>
<tr>
<td>HybridPass (developed proposal)</td>
<td>Database attack: the VP is hashed in the backend.</td>
</tr>
<tr>
<td></td>
<td>Shoulder surfing: multiple attacks since the system inquires 2 letters of the memorable phrase at a time. In addition, both keyboard and monitor must be observed. The monitor can expose the ID and the VP. The keyboard can expose the ID and the memorable phrase.</td>
</tr>
<tr>
<td></td>
<td>Malware: Keystrokes, mouse clicks and screen must be recorded to recover the full user credentials.</td>
</tr>
<tr>
<td><strong>Bank</strong></td>
<td><strong>Database attack</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Lloyds TSB</strong></td>
<td>the text password is hashed in the backend.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HSBC</strong></td>
<td>Database attack: no user credentials is hashed, hence a successful database attack can reveal the credentials of all registered accounts.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Alliance &amp; Leicester</strong></td>
<td>Database attack: the PIN is hashed in the backend. However, the password space is small therefore it can be recovered in less time with a brute force attack compared to text and VP.</td>
</tr>
<tr>
<td><strong>Barclays</strong></td>
<td>Database attack: the passcode is hashed in the backend. However, the password space is small therefore it can be recovered in less time with a brute force attack compared to text passwords and VP.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.8. Man-in-The-Middle (MiTM) phishing attack

This is an advanced attack combining phishing with MiTM techniques. As such, the phishing server acts as a proxy between the user who was forwarded from a phishing email and the original system as illustrated in Figure 7.12. Upon request, the phishing server imports and forwards a live copy of the system’s layout and client-side functionality to the victim and communicates their inputs and requests back to authenticate successfully. The attacker eavesdrops on the connection to recover user credentials in real-time or remains logged in after the user decides to log off his account, in this case the system’s clone is logged out in front of the victim, while the session between the attacker and original system remains open (RSA, 2007).

![Figure 7.12: In a MiTM-phishing, the attacker creates two separate connections pretending to be the bank’s website in the user’s browser and then interacts with the bank’s website using the user’s data.]

7.8.1. Sample attacks

Citibank’s customers are provided with tokens to generate one-time passwords (OTPs) to authenticate. Static phishing attacks are not effective since the password changes over time. But a MiTM-phishing enables a deceiver to bypass this defence mechanism because the attack was designed to make instant use of the token-generated password to login (Krebs, 2006). To illustrate such attacks (Soghoian and Jakobsson, 2007) video recorded a demonstration of how to interact with the website of BoA and retrieve the SiteKey on a spoof server. Figure 7.13 demonstrate a MiTM-phishing attack against BoA showing the first screen of the sign-in procedure secured with SSL (check the lock icon next to the URL). However, the address bar shows the domain name to be: http://sitekey.evil-phisher.com/ (This domain doesn't exist. It was created on a university
computer with a copy of the apache web server running on the 'localhost'). Figure 7.14 shows how a spoof server can recover the SiteKey. After feeding the forms with correct user details, the spoof site contacts the original site and imports the SiteKey. The spoof page is displaying the correct SiteKey using a phishing domain, while BoA note reads: ‘If you recognize your SiteKey, you’ll know for sure that you are at the valid Bank of America site’. Further, Soghoian and Jakobsson posted a sample of the code to query BoA’s SiteKey as a factual proof (Soghoian, 2007)

![Bank of America phishing attack](image)

**Figure 7.13:** A demonstration of a MiTM-phishing attack against BoA showing the first screen of the sign-in procedure secured with SSL.
164

Figure 7.14: A demonstration of a MiTM-phishing attack against BoA. After feeding the forms with correct user details.

7.8.2. Analysis of solutions

Extending user authentication with OTPs, shared secrets or authentication codes delivered via second channels such as phones and emails (2-factor authentication) are all vulnerable to MiTM attacks (Krebs, 2006) since they can be passed through from the user’s computer to the original service provider. However, the attack can be stopped using one of the anti-phishing tools discussed in Section 7.2.2, if the tool succeeds in recognising the domain name as a spoof, but relying exclusively on an anti-phishing tool has many constraints as analysed earlier. Hence, more techniques are developed to provide a level of resistance to MiTM-phishing such as:

Secure cookies: For example, these are used in the SiteKey implementation to link a user’s computer to the Bank of America website. A secure cookie can only be read by the bank’s server to recognise the computer and display the relevant SiteKey. A server with access to the cookie should be trusted. But what happens if the cookie is deleted or
a new computer is used? The server will ask a challenge question to rebuild the trust and display the SiteKey. Hence, an attacker can exploit this to recover the SiteKey and the end user will assume the server is re-authenticating his/her computer. Yahoo uses a similar approach employing secure cookies called: Personalised Sign-in Seals (Yahoo!, 2010).

**IP Address Geolocation:** The service provider compares the geographical location of the connected IP address with the user’s location. If the registered account is in the UK while the IP is connected from another country then a fraud is suspected as this can be the location of a MiTM phisher. Some banks request users to manually select their location while signing in to give them flexibility using their accounts from different places. While this anti-fraud mechanism can help to stop many fraudulent attacks, it can still be fooled by routing the MiTM server to a local proxy/botnet located in the same geographical region as the victim. Further, the MiTM server can change the geographical details of the user before passing it to the server to match those of the server.

**Content Verification:** It binds the domain name of a website with the content that needs to be protected (e.g. a logo or a sign-in form) into a single entity and users get a plug-in installed to verify the integrity of the protected entities. When a user encounters protected content, the real site would display a green border indicator (contentverification.com, 2010). While this can be used to stop attackers from copying the content, it does not stop them from creating their own version of the content that looks identical to the original. Nevertheless, it cannot stop a pharming attack where the real domain name of the service provider is used. Further, it requires installation on the user machine, so it may not exist on other shared computers.

### 7.8.3. Proposed scheme to resist MiTM-phishing

Technology employed to design online user authentication methods essentially includes HTML tags in addition to JavaScript code to apply form validation. We argue that MiTM-phishing attacks occur because the code is not protected and therefore can easily be edited and retransferred. For instance, a ‘universal MiTM kit’ (RSA, 2007) was developed to create system clones and make MiTM phishing easy to perform even by an attacker who lacks the necessary programming skills. Figure 7.15 illustrates how the original HTML code can easily be altered to perform the attack. The first block of code is taken from the login page of Lloyds TSB Bank (https://online.lloydstsb.co.uk/customer.ibc). A MiTM proxy could alters the Action attribute of the HTML login form to a URL belonging
to the spoof server to receive the submitted data and then resubmit these to the original system to gain access.

```html
...<form method="post" id="form"
action="https://online.lloydstsb.co.uk/logon.html"
name="theform" autocomplete="off">
  <div id="formDividerTitleUnderscore" class="title">Log on details</div> ...

...<form method="post" id="form"
action="https://online.lloydstsb.com/logon.html"
name="theform" autocomplete="off">
  <div id="formDividerTitleUnderscore" class="title">Log on details</div> ...
```

**Figure 7.15:** Altering HTML code to run Phishing attacks.

To protect the important block of code used for authentication, a programming language such as Java Applets must be used. Java class files are not impossible to decompile by default, but they can be obfuscated to prohibit unwanted access to the original source code using obfuscator software such as (yGuard, 2010). This way, the form will always submit to the original server because the form’s target address is hardcoded. Nevertheless, all attempts to decipher this kind of protected code require a tool to perform analysis tasks, hence a MiTM attack will still be stopped since the attack is time-limited. Assuming that such an advanced tool is developed, the analysis process requires time in addition to a programmer to read the result and alter the code accordingly to rebuild the class file, therefore the MiTM proxy will fail to forward the login form to the user on time.

A possible technique to bypass this defence mechanism is to analyse the class files prior to receiving a login request from a user. This can be stopped through implementing an expiry time for each authentication form sent from the original server. Each code sent out includes a unique hash that will be validated when the user credentials are submitted. The form is rejected by the server if its creation time exceeds a reasonable amount of time. For instance, an authentication form requesting a username and a password can have a life span of 60 seconds.

If deciphering the code is not effective, the deceiver will design a similar authentication form for a certain service provider to gain user credentials similar to a static phishing attack and then write a script to input the user details into the legitimate form. To resist these scripts, the authentication form must include human validation...
technology. For example, CAPTCHAs can be used but these are not fool-proof hence the form can be designed to accept all or part of the user credentials via mouse clicks to stop the script from filling in the form’s fields automatically. However, the script can be designed to cause the original form to submit and alter its data after it leaves the form and before it is submitted to the internet. The link inside the spoof server between the form and the point where the data is sent to the internet can be protected through encryption, as Java Applets can be used to encrypt the data inside the original form before sending it out to the internet where the browser adds the default SSL encryption used to secure the connection on the internet.

7.9. Conclusion

The phishing problem continues to cause financial loses of billions of dollars (Gartner, 2007) to major online service providers including banks despite user training, anti-fraud announcements and the development of many security tools. Authentication systems used in financial services such as PayPal and banks websites are the most popular target for phishers as reported by Avira security group (Raywood, 2009b). Phishing resistance is hard due to a number of factors, essentially the human factor accompanied by the sophistication of recent attacks. An attacker who is skilled and motivated enough can design a dynamic server which can respond accordingly to the user and the genuine server as a MiTM; further the legitimate domain name can be exploited through DNS poisoning, also known as Pharming.

Although user education is mandatory to control the problem, service providers must research new methods to provide their customers with better protection against online fraud. Phishing is a web authentication problem hence two schemes were developed as a factual proof that a better design of the authentication system can resist more phishing attempts and provide better protection. Nevertheless, implementing online user authentication methods with a powerful and secure yet usable technology (programming languages such as Java Applets) instead of current HTML forms provide better resistance against MiTM-phishing attacks and creates a better environment to develop secure solutions.

However, user training is a vital requirement. Security groups and online service providers must develop methods to educate users about the recent tricks used by phishers. A good example is a game called ‘Phish/No Phish’ developed by VeriSign (hosted at https://www.phish-no-phish.com) for users to experiment their ability to spot
phishing sites (SC Staff, 2009). To remain secure on the internet, end users can be taught the following good practices:

1) If you buy online, make sure to check that both the website’s address and name are being authenticated in your browser. This means the website has an EV SSL Certificate as explained earlier. If this is the case, the address bar will change its colour to green and to see it make sure your browser is not outdated.

2) If you have to use a trusted service provider that has no EV Certificate, make you that the check-out or authentication page has been secured with SSL. Look for the ‘lock’ icon on the right hand side of the address bar.

3) Do not click links in emails claiming to be from your bank or online financial services such as PayPal. Always type in the address yourself in a new window if you decide to login to any website. If they ask you to go to a specific page with a long address and you have your doubts, contact their customer service to check the validity of that email or message using the contact details published on their official documents and not from the email. Another way is to search parts of the email using a search engine (e.g. Google); if the email is a phishing scam you could find it posted in phishing forums and other online databases. Similarly, do not give any sensitive information to a form attached to an email address.

4) If you notice spelling mistakes in the email or the bank’s website, this is a sign of a spoof. Investigate the matter.

5) If you receive a sudden warning to update your details in the email or a website you are visiting, investigate the matter as this is a common phishing scenario.

6) Install good internet security software instead of a basic anti-virus. These usually protect you from a variety of attacks including phishing scams and malware.

7) Use the ‘Remember me’ feature to store your ID. Expect the website to recognise you every time you login. If you did not delete your cookies and the website failed to fill in your ID, be conscious and do further investigation before signing in with your password. Do not use the ‘Remember me’ feature on shared computers as this will compromise your ID.

8) It is a good practice to bookmark your important links to make sure you reach the legitimate address every time you visit them.
Chapter 8  Conclusion and Future Work

8.1. Conclusion

Novel contributions of this research started with the investigation of public attitudes and perceptions towards different forms of visual passwords and towards the implementation of continuous monitoring in online authentication systems. The research also provided evidence that the number of frequently used password-protected systems is continually increasing.

A flexible scheme (HybridPass) was designed with advantages over older click-based system such as Passpoints and CCP in terms of flexibility and providing better resistance to security threats such as shoulder-surfing and phishing. In addition, it was designed to eliminate denial of service attach since it replaces the traditional technique of controlling the number of incorrect login attempt to resist guessing with an alternative novel technique. Further, the prototype was used to identify criteria and address challenges applicable to click-based systems such as image types, tolerances, accuracy and dictionary attacks (the hotspots problem).

Another scheme (CCBS) was then developed with a novel design to resist dictionary attacks while maintaining the password space, accuracy of password submissions and user memorability for the short and long term.

True Discretization was proposed, a method to accept clicks within an area of tolerance with zero false accepts/rejects in click-based systems. Older methods from the literature are proved to allow false accepts and/or false rejects if applied with no further modification.

Further, an anti-phishing approach is developed and presented following an extensive security and usability discussions of various proposals from the literature. Existing schemes such as PassMark (currently used for internet banking) and HybridPass/CCBS (previously proposed in earlier chapters) were developed. Their superior resistant to phishing attacks was then analysed and evaluated against schemes currently used in
various banks including HSBC, Lloyds TSB, Bank of America and others. Finally, a new scheme to resist the new generation of phishing attacks (Man-in-The-Middle phishing) is proposed.

At this stage revisiting the aims and objectives is mandatory to evaluate this research. The first aim of the study was to:

*Develop a new VP authentication mechanism for web applications. The authentication scheme must be user-friendly, cost-effective and provide novel techniques to resist some of the security threats text passwords are vulnerable to, such as phishing, brute force and dictionary attacks.*

Nevertheless, and because text passwords remain the de facto approach to user authentication and are still used in the vast majority of systems (DTI, 2006), the study aimed secondly to:

*Investigate users’ awareness of, and attitudes toward, text password security and usability and provide up-to-date information.*

**Objective 1:** Investigate users’ attitudes toward text passwords, their perceptions of other knowledge/secret-based mechanisms and the concept of continuous monitoring.

This was addressed (Chapter 2) by investigating the text password problem, possible attacks and countermeasures and extended with a comprehensive survey \((N = 157)\) of user perception to study:

- Public awareness of, and attitudes toward, password security and usability.
- Public attitudes towards different forms of initial user authentication for web applications.
- Public attitudes towards the concept of continuous monitoring.

The survey concluded with a number of findings, amongst which:

- People are dealing with an increasing number of access-controlled accounts.
- Text passwords remain the most adopted technique to authenticate end users.
- A limited portion of end users pay attention to secure practices. For instance, only 5.10% change the passwords of their critical accounts on a monthly basis; only 21% always use a combination of letters, numbers and symbols; and
sharing passwords is a significant phenomenon with 74.54% of them being given a password of their friends or colleagues.

**Objective 2:** Identify the best VP approach based on a security and usability evaluation to develop a substantial authentication solution suitable for web application.

An up-to-date survey of VP schemes available in the literature was conducted (Chapter 2). Consequently, the most appropriate approach was selected based on a security and usability evaluation. This was found to be a cued-recall click-based system in which the password is a sequence of click-points on one image or more.

**Objective 3:** Build the new authentication scheme to be more resistant to brute force and dictionary attacks.

A click-based scheme was developed (Chapters 4, 5, 6 and 7) in which the password is cued-recall. This approach was adopted based on a security and usability analysis of a survey of existing VP schemes. The scheme’s first prototype is called HybridPass. The security and usability of the system was first discussed theoretically and then analysed based on a laboratory study. The advantages of the new scheme include design flexibility; it requires only a small layout and supports image-navigation compared to the other click-based systems from the literature. Nevertheless, its password space is relatively large to resist brute force attacks while maintaining memorability. The system is easy to learn by new users and has high success rates (95%). Further, it has a built-in technique to authenticate the server which is critical to resisting phishing attacks. However, analysis of the laboratory study of HybridPass and the related work from the literature revealed a number of security and usability challenges in click-based systems.

An example of the usability problems is the accuracy of password submissions; there was no method for the user to confirm submission of the intended click-point to the system. In the HybridPass experiment 70% of the rejected clicks were one to four pixels away from the accepted tolerance which implies that users recognised the correct click-point but failed to select it correctly. Increasing tolerance to accept these clicks is problematic because that would reduce the number of available click-points in the system which affects the password space. The VP retention for the long term was another concern. A related study (Wiedenbeck et al., 2005c) found no significant difference in password retention after one week and then after a month between text passwords and the VP in Passpoints. This implies that click-based systems could not be the solution to maintain better memorability in the long term. Further, in click-based systems, users are expected to sign-in with a VP string that consists of coordinates.
within a specific tolerance distance. That is necessary, because people’s ability to recall and click the all exact pixels is very rare. Implementing a system with a tolerance distance implies that the system should accept all string hashes within that tolerance area. A 6-pixel tolerance from the actual click-point results in 13x13 possible hashes for every click individually. The number of hashes grows exponentially with additional tolerance, which are, in any case, too many to store. Thus, discretization is used to solve this problem. Existing, discretization methods such as Robust Discretization (Birget et al., 2003) are proven to be vulnerable to false accepts and false rejects. Thereafter, Centred Discretization (Chiasson et al., 2008c) was presented to provide better solution. Nevertheless, both methods produced grid square tolerance in the 2-dimensional space. This can result in an effective rate of false accepts.

The HybridPass experiment revealed a number of security problems. For instance, VP analysis shows that most clicks are contained within hotspots; and that every image has a hotspot map. This can be used to run dictionary attacks against the system. Shoulder-surfing is another concern since the VP is entered using the mouse and the screen is more exposed in front of other individuals than a keyboard.

In response to the discretization problem, True Discretization is developed as a method to maintain the accuracy of click-based systems by allowing discretization with zero false accepts rate since all clicks outside the accepted distance are rejected.

A second prototype of the system called Cued Click-Based System (CCBS) was developed to maintain memorability in the long term, accuracy of password submissions and a large password space immune to dictionary attacks. These were major challenges to implement a usable and secure click-based system as discussed earlier. System-generated VPs were used to force a uniform distribution of click-points to resist hotspots and hence dictionary attacks. CCBS was designed with two cues to maintain memorability, a textual cue in addition to the visual one. The results of a laboratory study gave empirical evidence to support the usability of adopting system-generated passwords. It was found that using system-generated passwords in CCBS did not have negative impact on password retention in comparison with Passpoints and text passwords. In addition, users accurately selected the intended click-points to input their VPs, hence the number of incorrect password submissions was reduced significantly.

Success rates of VP retention in CCBS were significantly higher than Passpoints and text passwords. This implies that click-based systems can be developed to maintain better memorability in the long term than traditional text passwords. Further, each click-
point can be entered using the keyboard in an attempt to resist cursor vulnerability to shoulder-surfing. Users found the scheme easy-to-use but password submission was significantly slower than text passwords, especially if the retention is after a week or more.

**Objective 4:** Develop an effective anti-phishing technique.

Phishing attacks are investigates (Chapter 7). Existing proposals are analyses to understand why phishing still works. In response to phishing attacks, the research first developed an approach towards better anti-phishing solutions. Secondly, the research addressed the problem as a web authentication problem and developed technical solutions for two schemes, PassMark (an existing scheme used by the Bank of America) and HybridPass/CCBS. Security analysis of the proposals are factual proof that a better design of authentication systems can resist more phishing attempts (including MiTM attacks) hence provide better protection.

Finally, user training is essential. As such, good practices were discussed and suggested for end users.

**Objective 5:** Evaluate the scheme against different attacks analysing its strength and weakness to achieve better security and usability.

Password attacks against the system were addressed (Chapter 4) and consequently investigated (Chapters 5, 6 and 7). For example, after presenting the HybridPass scheme, vulnerabilities such as the hotspots problem (which can be exploited to run dictionary attacks against the system) were addressed. Solutions were developed to resist these security threats. The evaluation covered other type of attacks such as shoulder-surfing, guessing and key loggers. The final prototype of HybridPass/CCBS can resist database attacks with a hashed VP in the backend database. Multiple shoulder-surfing or key logging attacks are required to recover the password, and both, keyboard and monitor must be observed.

**8.2. Future work**

This research developed a novel authentication scheme. Laboratory studies found it usable with password retention rates better than text passwords. In addition, it provides better resistance to brute force, dictionary and phishing attacks. However, further research is still required to address additional security and usability issues. For instance, an important concern to the usability of visual password is interference. Further work
must investigate the effect of having multiple click-based VPs since psychological research such as (Wixted, 2004) indicates that interference causes significant memory problems. It may be expected that generating multiple VP using the same portfolio will cause memory problems due to interference. Hence, using a different portfolio to generate click-based VPs may be critical. In addition, click-point selection is performed using the mouse or keyboard in CCBS. Since the size of each click-point is small (to maintain a large password space), testing is required to ascertain if that will cause problems on personal digital assistants (PDAs) and mobile phones.

Further, the results of the experiments developed throughout the research are based on laboratory studies and theoretical discussions. Countermeasures were taken to maintain an environment of real-world characteristics; however, a significant field-study will give more accurate data to investigate how the system would operate online in practice. An appropriate field-study can last from weeks to a couple of months allowing a large number of users to authenticate to a web-based service. In addition, testing the system online is vital to study the scheme’s behaviour under different type of security attacks. Moreover, user comments based on their experience in real environments will assist developments of the design or discover new security threats.
Appendices

Appendix A - Survey of user perceptions (snapshots)

Figure A.1: Snapshot 1/9 showing the welcome page of the online application survey presented in Chapter 3.
Figure A.2: Snapshot 2/9 of the online application survey presented in Chapter 3.
### User perceptions of online authentication mechanisms

#### Part Two: Level of computer usage

- **Do you have a computer?**
  - Choose one of the following answers
    - At home & work

- **Do you have internet access?**
  - Choose one of the following answers
    - At home & work

- **How long have you been using computers?**
  - Choose one of the following answers
    - 1 Year
    - 2-5 Years
    - Over 5 Years

- **How long have you been using the internet?**
  - Choose one of the following answers
    - 1 Year
    - 2-5 Years
    - Over 5 Years

- **How many systems do you use that require passwords?**
  - (Each and every website you have an account with is counted as a system)
  - Choose one of the following answers
    - More than 30

- **Do you use any Single Sign-On software?**
  - Yes
  - No

**What is Single Sign-On (SSO) software?**

This software enables users to log in with a single password to their different email accounts, websites and Windows applications without being prompted to enter each system’s password individually.

The SSO software stores all users’ passwords in its encrypted database and uses it for the related accounts only when the user enters the software password.

Please note that I am not referring here to limited solutions such as Windows Live ID, which logs you in to Microsoft different services.

---

**Figure A.3:** Snapshot 3/9 of the online application survey presented in Chapter 3.
Figure A.4: Snapshot 4/9 of the online application survey presented in Chapter 3.
Figure A.5: Snapshot 5/9 of the online application survey presented in Chapter 3.
Appendices

Part Four: Alternative techniques for initial login

**What is an initial login technique?**

It is the technique used to check your identity when you first log in to your account. An example is when you first enter your username and password to log in to your email; in this case the technique used is "password".

Please rate the acceptability of the following initial login techniques to you on a 5-point scale where 1 is "Very Unacceptable" and 5 is "Very Acceptable".

<table>
<thead>
<tr>
<th>Password</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate the acceptability:</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question and Answer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate the acceptability:</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

*What is Question and Answer?*

When you set up your account, you were asked some personal "security questions." When you log in you are asked to answer some or all of the questions.

Examples:

- What is the name of your favourite place to visit?
- Where did you spend your childhood?
- Who was your favourite teacher in high school?

---

**Figure A.6**: Snapshot 6/9 of the online application survey presented in Chapter 3.
Figure A.7: Snapshot 7/9 of the online application survey presented in Chapter 3.
**User perceptions of online authentication mechanisms**

**Part Five: Continuous monitoring**

**What is continuous monitoring?**

It is the process of tracking users’ behaviour continuously during a logged-in session to verify their identity. So if you leave your email account open and another person tries to use it, the system will log him/her out automatically.

Please rate the following on a 5-point scale where 1 is "Very Unacceptable" and 5 is "Very Acceptable".

<table>
<thead>
<tr>
<th>How comfortable you are with the concept of continuous monitoring:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate the acceptability:</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rate the acceptability of using Mouse Dynamics as a continuous monitoring technique:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate the acceptability:</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

What is mouse dynamics?

This method is based on the way users move their mouse. The system can automatically analyze the way you move your mouse to know if it is you who is still using the account, or someone else.

If it is someone else, the system logs out your account automatically.

<table>
<thead>
<tr>
<th>Rate the acceptability of using Keystroke Dynamics as a continuous monitoring technique:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate the acceptability:</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

What is keystroke dynamics?

This method identifies users based on the way they use their keyboard. The system is able to analyze the habits of every person while typing on the keyboard and know if the same person is still using the account, or someone else.

If it is someone else, the system logs out your account automatically.

---

**Figure A.8:** Snapshot 8/9 of the online application survey presented in Chapter 3.
Figure A.9: Snapshot 9/9 of the online application survey presented in Chapter 3.
Appendix B - Experimental data (Chapter 4)

Table B.1: Prototype 1 registration data. (*) is used to highlight data for successful registrations only.

<table>
<thead>
<tr>
<th>#</th>
<th>Password creation time (first trial)</th>
<th>Password creation time (second trial)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Registration time</td>
<td>Confirmation time</td>
</tr>
<tr>
<td>1</td>
<td>149</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>65</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>156</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>346</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>87</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>267</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>236</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>132</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>262</td>
<td>27</td>
</tr>
<tr>
<td>11</td>
<td>92</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>117</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>442</td>
<td>46</td>
</tr>
<tr>
<td>14</td>
<td>155</td>
<td>23</td>
</tr>
<tr>
<td>15</td>
<td>132</td>
<td>22</td>
</tr>
<tr>
<td>16</td>
<td>140</td>
<td>31</td>
</tr>
<tr>
<td>17</td>
<td>147</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>166</td>
<td>36</td>
</tr>
<tr>
<td>19</td>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>20</td>
<td>369</td>
<td>34</td>
</tr>
<tr>
<td>21</td>
<td>160</td>
<td>68</td>
</tr>
<tr>
<td>22</td>
<td>224</td>
<td>18</td>
</tr>
<tr>
<td>23</td>
<td>142</td>
<td>30</td>
</tr>
<tr>
<td>24</td>
<td>198</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>80</td>
<td>43</td>
</tr>
<tr>
<td>26</td>
<td>129</td>
<td>30</td>
</tr>
<tr>
<td>27</td>
<td>50</td>
<td>23</td>
</tr>
<tr>
<td>28</td>
<td>50</td>
<td>124</td>
</tr>
<tr>
<td>29</td>
<td>136</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>54</td>
<td>11</td>
</tr>
<tr>
<td>31</td>
<td>80</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>99</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>33</td>
<td>101</td>
<td>21</td>
</tr>
<tr>
<td>34</td>
<td>204</td>
<td>35</td>
</tr>
<tr>
<td>35</td>
<td>51*</td>
<td>15*</td>
</tr>
<tr>
<td>36</td>
<td>88*</td>
<td>28*</td>
</tr>
<tr>
<td>37</td>
<td>123*</td>
<td>29*</td>
</tr>
<tr>
<td>38</td>
<td>131*</td>
<td>36*</td>
</tr>
<tr>
<td>39</td>
<td>108*</td>
<td>28*</td>
</tr>
<tr>
<td>40</td>
<td>148*</td>
<td>37*</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean time</td>
<td>153.125</td>
<td>30.5</td>
<td>57.79</td>
<td>24.47</td>
</tr>
<tr>
<td>SD</td>
<td>89.37</td>
<td>18.34</td>
<td>47</td>
<td>20</td>
</tr>
<tr>
<td>Min</td>
<td>50</td>
<td>11</td>
<td>31</td>
<td>12</td>
</tr>
</tbody>
</table>

**Mean of the number of incorrect submissions (SD)**

1.42 (0.74)

N=40

**Successfully registered**

6/40 (15%) | 11/40 (27.5%) | 42.5%
### Table B.2: Prototype 1 sign-in times and calculation details. Data are from participants who successfully sign-in. (*) is used to highlight data for successful registrations only.

<table>
<thead>
<tr>
<th>#</th>
<th>Text password entry time</th>
<th>VP entry time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>31</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Text password entry</th>
<th>VP entry time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>14.69</td>
<td>19.15</td>
</tr>
<tr>
<td>SD</td>
<td>4.47</td>
<td>7.17</td>
</tr>
<tr>
<td>Min</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>
### B.3: Prototype 2 and Prototype 3 registration times, success rates and other calculated information.

<table>
<thead>
<tr>
<th>#</th>
<th>Password creation time</th>
<th>Confirmation time</th>
<th>Password creation time</th>
<th>Confirmation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49</td>
<td>21</td>
<td>36</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>82</td>
<td>28</td>
<td>59</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>15</td>
<td>24</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>58</td>
<td>24</td>
<td>86</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>53</td>
<td>25</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>13</td>
<td>28</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>20</td>
<td>48</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>67</td>
<td>26</td>
<td>63</td>
<td>38</td>
</tr>
<tr>
<td>9</td>
<td>36</td>
<td>22</td>
<td>35</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>39</td>
<td>22</td>
<td>54</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>28</td>
<td>16</td>
<td>47</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>69</td>
<td>23</td>
<td>42</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>68</td>
<td>23</td>
<td>39</td>
<td>22</td>
</tr>
<tr>
<td>14</td>
<td>53</td>
<td>51</td>
<td>43</td>
<td>33</td>
</tr>
<tr>
<td>15</td>
<td>46</td>
<td>28</td>
<td>52</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>145</td>
<td>31</td>
<td>78</td>
<td>30</td>
</tr>
<tr>
<td>17</td>
<td>43</td>
<td>24</td>
<td>47</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>46</td>
<td>21</td>
<td>33</td>
<td>19</td>
</tr>
<tr>
<td>19</td>
<td>57</td>
<td>25</td>
<td>80</td>
<td>23</td>
</tr>
<tr>
<td>20</td>
<td>98</td>
<td>42</td>
<td>39</td>
<td>16</td>
</tr>
<tr>
<td>21</td>
<td>42</td>
<td>13</td>
<td>74</td>
<td>35</td>
</tr>
<tr>
<td>22</td>
<td>115</td>
<td>33</td>
<td>44</td>
<td>43</td>
</tr>
<tr>
<td>23</td>
<td>67</td>
<td>26</td>
<td>99</td>
<td>28</td>
</tr>
<tr>
<td>24</td>
<td>58</td>
<td>26</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td>25</td>
<td>52</td>
<td>36</td>
<td>46</td>
<td>31</td>
</tr>
<tr>
<td>26</td>
<td>119</td>
<td>34</td>
<td>36</td>
<td>17</td>
</tr>
<tr>
<td>27</td>
<td>41</td>
<td>24</td>
<td>87</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>30</td>
<td>15</td>
<td>25</td>
<td>16</td>
</tr>
<tr>
<td>29</td>
<td>36</td>
<td>20</td>
<td>52</td>
<td>24</td>
</tr>
<tr>
<td>30</td>
<td>37</td>
<td>15</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>22</td>
<td>54</td>
<td>24</td>
</tr>
<tr>
<td>32</td>
<td>38</td>
<td>20</td>
<td>60</td>
<td>31</td>
</tr>
<tr>
<td>33</td>
<td>71</td>
<td>16</td>
<td>31</td>
<td>15</td>
</tr>
<tr>
<td>34</td>
<td>42</td>
<td>22</td>
<td>61</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>63</td>
<td>42</td>
<td>29</td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>36</td>
<td></td>
<td>48</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td>30</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>54.51</td>
<td>25.37</td>
<td>50.11</td>
<td>23.51</td>
</tr>
<tr>
<td>SD</td>
<td>27.62</td>
<td>9.58</td>
<td>19.08768</td>
<td>7.53</td>
</tr>
<tr>
<td>Min</td>
<td>21</td>
<td>13</td>
<td>24</td>
<td>11</td>
</tr>
<tr>
<td>Successfully registered</td>
<td>37 (92.5%)</td>
<td>35 (87.5%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of incorrect submissions</td>
<td>0.325 (0.61)</td>
<td>0.475 (0.75)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table B.4: Prototype 4 and Prototype 5 registration times, success rates calculated details.

<table>
<thead>
<tr>
<th>#</th>
<th>Password creation time</th>
<th>Confirmation time</th>
<th>Password creation time</th>
<th>Confirmation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>43</td>
<td>22</td>
<td>34</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>46</td>
<td>32</td>
<td>61</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>25</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>14</td>
<td>45</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>41</td>
<td>23</td>
<td>35</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>34</td>
<td>13</td>
<td>27</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td>17</td>
<td>58</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>66</td>
<td>21</td>
<td>58</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>51</td>
<td>25</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>39</td>
<td>46</td>
<td>35</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>43</td>
<td>15</td>
<td>22</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>38</td>
<td>17</td>
<td>43</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>38</td>
<td>20</td>
<td>47</td>
<td>22</td>
</tr>
<tr>
<td>14</td>
<td>34</td>
<td>21</td>
<td>46</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>85</td>
<td>32</td>
<td>43</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>39</td>
<td>11</td>
<td>38</td>
<td>29</td>
</tr>
<tr>
<td>17</td>
<td>84</td>
<td>38</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>18</td>
<td>39</td>
<td>21</td>
<td>38</td>
<td>42</td>
</tr>
<tr>
<td>19</td>
<td>58</td>
<td>26</td>
<td>37</td>
<td>29</td>
</tr>
<tr>
<td>20</td>
<td>101</td>
<td>34</td>
<td>57</td>
<td>42</td>
</tr>
<tr>
<td>21</td>
<td>58</td>
<td>55</td>
<td>35</td>
<td>13</td>
</tr>
<tr>
<td>22</td>
<td>56</td>
<td>22</td>
<td>86</td>
<td>29</td>
</tr>
<tr>
<td>23</td>
<td>84</td>
<td>40</td>
<td>59</td>
<td>24</td>
</tr>
<tr>
<td>24</td>
<td>57</td>
<td>20</td>
<td>162</td>
<td>29</td>
</tr>
<tr>
<td>25</td>
<td>71</td>
<td>36</td>
<td>46</td>
<td>20</td>
</tr>
<tr>
<td>26</td>
<td>54</td>
<td>23</td>
<td>38</td>
<td>17</td>
</tr>
<tr>
<td>27</td>
<td>36</td>
<td>54</td>
<td>34</td>
<td>24</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>16</td>
<td>72</td>
<td>82</td>
</tr>
<tr>
<td>29</td>
<td>35</td>
<td>16</td>
<td>39</td>
<td>22</td>
</tr>
<tr>
<td>30</td>
<td>63</td>
<td>31</td>
<td>61</td>
<td>22</td>
</tr>
<tr>
<td>31</td>
<td>40</td>
<td>13</td>
<td>33</td>
<td>23</td>
</tr>
<tr>
<td>32</td>
<td>55</td>
<td>18</td>
<td>57</td>
<td>20</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>20</td>
<td>39</td>
<td>17</td>
</tr>
<tr>
<td>34</td>
<td>46</td>
<td>15</td>
<td>47</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>42</td>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>36</td>
<td>57</td>
<td>28</td>
<td>40</td>
<td>19</td>
</tr>
<tr>
<td>37</td>
<td>28</td>
<td>17</td>
<td>70</td>
<td>20</td>
</tr>
<tr>
<td>38</td>
<td>37</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>34</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>SD</th>
<th>Mean</th>
<th>SD</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>48.46</td>
<td>24.25</td>
<td>47.45</td>
<td>24.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>17.93</td>
<td>10.86</td>
<td>24.08</td>
<td>11.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>19</td>
<td>11</td>
<td>21</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

 Successfully registered: 39 (97.5%) 37 (92.5%)

 Number of incorrect submissions: 0.1 (0.37) 0.25 (0.58)
### Table B.5: Sign-in times, success rates and calculation details.

<table>
<thead>
<tr>
<th>#</th>
<th>Prototype 2</th>
<th>Prototype 3</th>
<th>Prototype 4</th>
<th>Prototype 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17</td>
<td>15</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>14</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>14</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>24</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>24</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>19</td>
<td>18</td>
<td>28</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>12</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>17</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>12</td>
<td>23</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>17</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>24</td>
<td>13</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>19</td>
<td>52</td>
<td>39</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>17</td>
<td>12</td>
<td>41</td>
</tr>
<tr>
<td>14</td>
<td>19</td>
<td>52</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>17</td>
<td>22</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>32</td>
<td>26</td>
<td>15</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
<td>18</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>18</td>
<td>30</td>
<td>18</td>
<td>11</td>
<td>25</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>12</td>
<td>18</td>
<td>25</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>15</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td>13</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>23</td>
<td>27</td>
<td>29</td>
</tr>
<tr>
<td>23</td>
<td>29</td>
<td>19</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>14</td>
<td>28</td>
<td>18</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>17</td>
<td>24</td>
<td>22</td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>13</td>
<td>40</td>
<td>21</td>
</tr>
<tr>
<td>27</td>
<td>17</td>
<td>9</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>28</td>
<td>11</td>
<td>16</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>29</td>
<td>14</td>
<td></td>
<td>25</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>27</td>
<td></td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td>46</td>
<td>29</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td>21</td>
<td>13</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>12</td>
<td>36</td>
<td>12</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Mean</td>
<td>19.06</td>
<td>19.14</td>
<td>18.86</td>
<td>18.03</td>
</tr>
<tr>
<td>SD</td>
<td>6.74</td>
<td>10.16</td>
<td>8.71</td>
<td>7.04</td>
</tr>
<tr>
<td>Min</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Success rates</td>
<td>30/37 (81%)</td>
<td>28/35 (80%)</td>
<td>37/39 (94.8%)</td>
<td>32/37 (86.4%)</td>
</tr>
<tr>
<td>Number of incorrect submissions</td>
<td>0.18 (0.39) N=37</td>
<td>0.2 (0.40) N=35</td>
<td>0.05 (0.22) N=39</td>
<td>0.13 (0.34) N=37</td>
</tr>
</tbody>
</table>
Table B.6: The images used in HybridPass experiments to create user portfolios and their reference codes.

<table>
<thead>
<tr>
<th>p01x</th>
<th>p02x</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>p03x</td>
<td>p04x</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>p05x</td>
<td>p06x</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>p07x</td>
<td>p08x</td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>p09x</td>
<td>p10x</td>
</tr>
<tr>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Appendices
Table B.7: The VPs (user clicks) collected throughout the experiments from the 5 prototypes.

These are the VPs of the successful registrations attempts only. Each click is represented by two numbers (image coordinates) and an image reference. Each element is separated by a semicolon.

<table>
<thead>
<tr>
<th>#</th>
<th>Prototype 1</th>
<th>Prototype 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2;5;p02x;1;95;p02x;224;95;p02x;224;2;p02x;</td>
<td>161;85;p06x;210;38;p06x;102;33;p06x;16;43;p06x;</td>
</tr>
<tr>
<td>2</td>
<td>137;43;p10x;169;44;p10x;219;75;p10x;157;75;p10x;</td>
<td>154;84;p01x;155;38;p01x;202;37;p01x;203;83;p01x;</td>
</tr>
<tr>
<td>3</td>
<td>5;3;p03x;225;3;p03x;225;92;p03x;6;94;p03x;</td>
<td>2;4;p03x;222;94;p03x;225;5;p03x;4;95;p03x;</td>
</tr>
<tr>
<td>4</td>
<td>80;8;p11x;26;25;p11x;55;19;p11x;43;59;p11x;</td>
<td>115;8;p05x;15;39;p06x;78;82;p04x;73;33;p03x;</td>
</tr>
<tr>
<td>5</td>
<td>101;27;p03x;102;14;p03x;102;35;p03x;101;47;p03x;</td>
<td>72;15;p03x;72;31;p03x;73;70;p03x;73;84;p03x;</td>
</tr>
<tr>
<td>6</td>
<td>89;15;p08x;135;13;p08x;160;12;p08x;109;78;p08x;</td>
<td>27;15;p01x;43;17;p01x;44;31;p01x;31;33;p01x;</td>
</tr>
<tr>
<td>7</td>
<td>196;28;p02x;78;50;p02x;61;55;p02x;48;55;p02x;</td>
<td>27;13;p01x;43;28;p01x;88;43;p01x;109;45;p01x;</td>
</tr>
<tr>
<td>8</td>
<td>80;82;p04x;127;36;p04x;134;85;p04x;170;59;p04x;</td>
<td>48;59;p04x;12;78;p01x;114;94;p01x;145;86;p01x;</td>
</tr>
<tr>
<td>9</td>
<td>115;9;p05x;112;81;p05x;114;8;p05x;72;33;p03x;</td>
<td>36;21;p01x;179;53;p01x;61;68;p01x;219;46;p01x;</td>
</tr>
<tr>
<td>10</td>
<td>5;75;p10x;222;91;p10x;156;76;p10x;134;82;p10x;</td>
<td>79;59;p02x;47;55;p02x;60;29;p02x;23;92;p02x;</td>
</tr>
<tr>
<td>11</td>
<td>192;86;p11x;191;63;p11x;191;40;p11x;3;97;p11x;</td>
<td>12;78;p01x;78;16;p01x;114;94;p01x;145;86;p01x;</td>
</tr>
<tr>
<td>12</td>
<td>76;33;p03x;73;84;p03x;161;37;p03x;176;72;33;p03x;</td>
<td>36;21;p01x;179;53;p01x;61;68;p01x;219;46;p01x;</td>
</tr>
<tr>
<td>13</td>
<td>208;66;p11x;170;54;p11x;58;45;p11x;26;44;p11x;</td>
<td>11;8;p10x;12;95;p10x;88;94;p10x;88;7;p10x;</td>
</tr>
<tr>
<td>14</td>
<td>5;75;p10x;222;91;p10x;156;76;p10x;134;82;p10x;</td>
<td>46;10;p04x;74;9;p04x;42;50;p04x;42;61;p04x;</td>
</tr>
<tr>
<td>15</td>
<td>18;68;p09x;75;65;p09x;23;34;p09x;41;32;p09x;</td>
<td>18;68;p09x;75;65;p09x;23;34;p09x;41;32;p09x;</td>
</tr>
<tr>
<td>16</td>
<td>48;9;p04x;72;7;p04x;45;49;p04x;45;60;p04x;</td>
<td>48;9;p04x;72;7;p04x;45;49;p04x;45;60;p04x;</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>184;55;p01x;175;55;p01x;219;46;p01x;219;83;p01x;</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>79;51;p02x;46;50;p02x;26;89;p02x;169;2;p02x;</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>216;59;p05x;151;74;p05x;180;92;p05x;138;91;p05x;</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>76;34;p03x;80;72;p03x;73;84;p03x;41;58;p03x;</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>73;32;p03x;71;72;p03x;74;84;p03x;157;24;p03x;</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>15;41;p06x;162;46;p06x;26;27;p06x;36;26;p06x;</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>36;20;p01x;53;20;p01x;35;5;p01x;18;21;p01x;</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>212;81;p02x;75;34;p03x;38;80;p04x;219;87;p05x;</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>36;22;p01x;34;67;p01x;183;54;p01x;218;47;p01x;</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>73;84;p03x;161;47;p06x;80;82;p04x;48;55;p02x;</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>161;47;p06x;15;41;p06x;25;29;p06x;36;29;p06x;</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>15;40;p06x;25;27;p06x;36;27;p06x;127;23;p06x;</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>36;19;p01x;13;81;p01x;219;64;p01x;125;65;p01x;</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>41;44;p05x;103;8;p05x;115;7;p05x;158;34;p03x;</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>79;51;p02x;46;50;p02x;196;29;p02x;153;15;p02x;</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>94;10;p04x;106;13;p04x;114;20;p04x;89;56;p04x;</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>16;43;p06x;26;29;p06x;37;29;p06x;86;9;p06x;</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>101;42;p01x;35;73;p06x;171;47;p04x;74;32;p03x;</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>44;46;p04x;45;60;p04x;64;53;p04x;88;55;p04x;</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>15;42;p06x;77;17;p06x;162;46;p06x;213;17;p06x;</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>49;53;p02x;158;41;p03x;16;78;p01x;207;36;p06x;</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>171;59;p04x;47;31;p01x;74;33;p03x;16;41;p06x;</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>86;54;p04x;65;54;p04x;47;51;p04x;46;62;p04x;</td>
<td></td>
</tr>
</tbody>
</table>

# Prototype 3

| 1 | 63;30;p08x;91;29;p08x;113;30;p08x;135;30;p08x; |
| 2 | 18;68;p09x;74;63;p09x;36;27;p08x;86;25;p08x; |
| 3 | 119;43;p09x;164;38;p09x;21;34;p09x;41;31;p09x; |
| 4 | 150;39;p10x;101;45;p11x;171;83;p09x;213;69;p13x; |
| 5 | 31;25;p08x;62;27;p08x;88;29;p08x;113;27;p08x; |
| 6 | 135;93;p08x;164;92;p08x;182;94;p08x;47;92;p08x; |
| 7 | 215;36;p09x;164;38;p09x;118;41;p09x;137;48;p11x; |
| 8 | 165;91;p08x;139;92;p08x;181;59;p08x;182;40;p08x; |
| 9 | 21;66;p09x;78;61;p09x;121;66;p09x;172;85;p09x; |
| 10 | 86;44;p12x;62;27;p08x;113;19;p08x;10;44;p13x; |
| 11 | 62;33;p08x;94;33;p08x;134;32;p08x;157;33;p08x; |
| 12 | 19;69;p09x;76;62;p09x;118;66;p09x;171;84;p09x; |
| 13 | 206;36;p11x;126;74;p11x;102;47;p11x;25;26;p11x; |
| 14 | 171;84;p09x;119;66;p09x;74;63;p09x;16;69;p09x; |
| 15 | 164;41;p09x;122;42;p09x;77;61;p09x;18;68;p09x; |
| 16 | 47;85;p08x;124;98;p08x;166;98;p08x;179;28;p08x; |
| 17 | 222;63;p12x;191;47;p12x;135;31;p12x;47;19;p12x; |
# Prototype 4

1 161;50;p06x;214;16;p06x;105;33;p06x;15;41;p06x;
2 112;53;p01x;217;86;p01x;218;11;p01x;11;11;p01x;
3 80;50;p02x;73;86;p03x;24;85;p05x;79;80;p04x;
4 37;80;p04x;80;83;p04x;2;96;p04x;226;96;p04x;
5 59;77;p04x;218;86;p05x;73;86;p03x;161;48;p06x;
6 28;12;p01x;41;12;p01x;27;32;p01x;42;31;p01x;
7 28;13;p01x;44;11;p01x;45;30;p01x;29;29;p01x;
8 87;57;p04x;72;33;p03x;72;71;p03x;72;83;p03x;
9 81;82;p04x;99;90;p04x;46;8;p04x;71;8;p04x;
10 12;81;p01x;79;96;p01x;113;94;p01x;145;83;p01x;
11 28;45;p06x;73;59;p06x;102;45;p06x;69;90;p06x;
12 175;54;p01x;219;52;p01x;218;68;p01x;219;81;p01x;
13 48;10;p04x;71;7;p04x;45;51;p04x;87;55;p04x;
14 169;44;p04x;101;89;p04x;79;82;p04x;38;81;p04x;
15 116;9;p05x;103;9;p05x;106;69;p05x;113;84;p05x;
16 83;59;p02x;45;54;p02x;24;91;p02x;38;21;p02x;
17 36;21;p01x;22;75;p01x;137;39;p01x;219;46;p01x;
18 185;9;p06x;223;8;p06x;162;49;p06x;15;43;p06x;
19 73;85;p03x;77;33;p03x;67;71;p03x;197;24;p03x;
20 36;21;p01x;179;54;p01x;109;47;p01x;88;43;p01x;
21 80;51;p02x;170;3;p02x;196;29;p02x;206;30;p02x;
22 37;21;p01x;52;21;p01x;37;5;p01x;20;22;p01x;
Appendices

23 79;52;p02x;156;24;p03x;80;82;p04x;24;86;p05x;
24 49;45;p05x;107;17;p05x;208;2;p05x;220;88;p05x;
25 157;23;p03x;73;84;p03x;71;71;p03x;75;33;p03x;
26 80;51;p02x;169;3;p02x;209;37;p02x;161;69;p02x;
27 43;43;p05x;110;8;p05x;114;76;p05x;217;60;p05x;
28 31;72;p06x;16;40;p06x;163;50;p06x;214;15;p06x;
29 184;55;p01x;174;53;p01x;36;81;p04x;84;81;p04x;
30 38;82;p04x;80;83;p04x;170;59;p04x;185;10;p04x;
31 50;51;p02x;64;48;p02x;79;63;p02x;23;91;p02x;
32 27;13;p01x;32;67;p01x;107;46;p01x;218;66;p01x;
33 73;33;p03x;98;42;p01x;40;48;p05x;33;70;p06x;
34 45;51;p04x;44;63;p04x;65;54;p04x;89;57;p04x;
35 23;86;p05x;115;7;p05x;180;61;p05x;219;84;p05x;
36 12;80;p01x;47;53;p02x;73;32;p03x;41;44;p05x;
37 28;13;p01x;43;11;p01x;28;29;p01x;45;30;p01x;
38 17;40;p06x;160;48;p06x;186;13;p06x;227;94;p06x;
39 90;56;p04x;62;51;p04x;45;47;p04x;44;62;p04x;

# Prototype 5

1 151;18;p10x;188;26;p10x;220;91;p10x;158;75;p10x;
2 123;72;p11x;138;46;p11x;167;28;p11x;206;36;p11x;
3 164;39;p09x;119;42;p09x;40;31;p09x;22;34;p09x;
4 157;76;p10x;119;21;p13x;207;35;p11x;75;63;p09x;
5 37;27;p08x;64;26;p08x;90;28;p08x;113;29;p08x;
6 20;54;p13x;43;49;p13x;30;80;p13x;46;72;p13x;
7 206;42;p11x;8;90;p13x;29;88;p13x;49;80;p13x;
8 23;56;p13x;42;52;p13x;18;19;p13x;165;18;p13x;
9 20;23;p10x;110;23;p10x;151;23;p10x;192;25;p10x;
10 9;45;p13x;33;39;p13x;44;93;p13x;59;85;p13x;
11 20;65;p09x;74;62;p09x;118;41;p09x;167;40;p09x;
12 55;22;p11x;139;49;p11x;207;44;p11x;123;76;p11x;
13 172;85;p09x;119;65;p09x;39;3;p09x;21;7;p09x;
14 172;84;p09x;121;71;p09x;75;63;p09x;19;68;p09x;
15 159;34;p08x;135;35;p08x;112;33;p08x;93;35;p08x;
16 18;68;p09x;75;62;p09x;164;39;p09x;171;84;p09x;
17 200;78;p12x;143;65;p12x;128;75;p12x;46;55;p12x;
18 117;43;p09x;164;39;p09x;171;84;p09x;120;66;p09x;
19 19;68;p09x;75;62;p09x;118;43;p09x;164;40;p09x;
20 18;68;p09x;75;62;p09x;22;34;p09x;41;31;p09x;
21 19;49;p10x;43;50;p10x;70;50;p10x;106;50;p10x;
22 75;62;p09x;20;42;p10x;130;19;p11x;8;63;p12x;
23 1;2;p12x;3;2;p08x;3;3;p13x;1;1;p11x;
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>111;77;p08x;217;60;p08x;206;6;p08x;182;20;p08x;</td>
</tr>
<tr>
<td>25</td>
<td>18;67;p09x;75;61;p09x;118;42;p09x;164;38;p09x;</td>
</tr>
<tr>
<td>26</td>
<td>43;57;p11x;57;26;p11x;28;25;p11x;125;76;p11x;</td>
</tr>
<tr>
<td>27</td>
<td>201;73;p13x;37;65;p12x;38;28;p12x;155;77;p10x;</td>
</tr>
<tr>
<td>28</td>
<td>51;28;p08x;121;14;p08x;58;55;p08x;118;58;p08x;</td>
</tr>
<tr>
<td>29</td>
<td>26;26;p11x;42;58;p11x;58;19;p11x;127;74;p11x;</td>
</tr>
<tr>
<td>30</td>
<td>84;76;p08x;178;40;p08x;135;92;p08x;162;92;p08x;</td>
</tr>
<tr>
<td>31</td>
<td>217;27;p13x;3;75;p10x;5;7;p12x;11;34;p08x;</td>
</tr>
<tr>
<td>32</td>
<td>76;85;p12x;91;85;p12x;108;88;p12x;121;88;p12x;</td>
</tr>
<tr>
<td>33</td>
<td>63;34;p08x;93;34;p08x;114;32;p08x;134;33;p08x;</td>
</tr>
<tr>
<td>34</td>
<td>214;68;p13x;20;65;p09x;157;76;p10x;112;77;p08x;</td>
</tr>
<tr>
<td>35</td>
<td>18;68;p09x;75;62;p09x;120;65;p09x;170;85;p09x;</td>
</tr>
<tr>
<td>36</td>
<td>217;69;p13x;103;49;p11x;131;56;p12x;171;84;p09x;</td>
</tr>
<tr>
<td>37</td>
<td>125;74;p11x;138;50;p11x;168;37;p11x;205;44;p11x;</td>
</tr>
</tbody>
</table>
Appendix C – Experimental data (Chapter 6)

**Table C.1**: The images used in the CCBS experiment to create users portfolios accompanied by their reference codes.

<table>
<thead>
<tr>
<th>p01x</th>
<th>p02x</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="image2.png" alt="Image 2" /></td>
</tr>
<tr>
<td>p03x</td>
<td>p04x</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image 3" /></td>
<td><img src="image4.png" alt="Image 4" /></td>
</tr>
<tr>
<td>p05x</td>
<td>p09x</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image 5" /></td>
<td><img src="image6.png" alt="Image 6" /></td>
</tr>
</tbody>
</table>
Table C.2: The VPs (user clicks) from the CCBS experiment. Each click is represented by two numbers (image coordinates) and an image reference code. Each element is separated by a semicolon.

<table>
<thead>
<tr>
<th>#</th>
<th>Visual passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4;0;p01x;19;0;p02x;23;4;p03x;5;7;p05x;21;4;p09x;</td>
</tr>
<tr>
<td>2</td>
<td>16;1;p09x;22;5;p05x;18;10;p04x;10;3;p02x;21;1;p01x;</td>
</tr>
<tr>
<td>3</td>
<td>3;3;p01x;14;5;p02x;5;5;p04x;17;1;p05x;15;10;p09x;</td>
</tr>
<tr>
<td>4</td>
<td>4;9;p02x;8;9;p05x;12;8;p01x;18;7;p04x;8;9;p09x;</td>
</tr>
<tr>
<td>5</td>
<td>24;6;p03x;9;8;p01x;17;7;p05x;8;6;p09x;16;6;p02x;</td>
</tr>
<tr>
<td>6</td>
<td>18;7;p02x;12;0;p03x;17;8;p09x;19;5;p01x;24;0;p05x;</td>
</tr>
<tr>
<td>7</td>
<td>3;3;p02x;22;3;p05x;16;7;p09x;25;3;p01x;5;2;p03x;</td>
</tr>
<tr>
<td>8</td>
<td>19;8;p05x;25;10;p01x;12;8;p03x;17;9;p04x;10;2;p02x;</td>
</tr>
<tr>
<td>9</td>
<td>7;5;p02x;2;6;p04x;14;4;p01x;20;6;p09x;20;1;p03x;</td>
</tr>
<tr>
<td>10</td>
<td>18;1;p05x;1;3;p01x;22;6;p02x;16;4;p03x;3;9;p09x;</td>
</tr>
<tr>
<td>11</td>
<td>14;3;p03x;4;6;p02x;10;8;p01x;17;7;p04x;24;1;p05x;</td>
</tr>
<tr>
<td>12</td>
<td>19;8;p04x;6;5;p01x;14;6;p05x;4;3;p09x;16;10;p02x;</td>
</tr>
<tr>
<td>13</td>
<td>8;2;p05x;2;6;p04x;4;7;p02x;20;8;p01x;23;3;p09x;</td>
</tr>
<tr>
<td>14</td>
<td>11;7;p03x;14;1;p05x;8;10;p09x;21;1;p02x;14;4;p05x;</td>
</tr>
<tr>
<td>15</td>
<td>14;8;p04x;2;2;p03x;17;2;p09x;3;1;p01x;11;10;p02x;</td>
</tr>
<tr>
<td>16</td>
<td>3;9;p03x;21;5;p01x;3;4;p05x;23;10;p02x;24;8;p09x;</td>
</tr>
<tr>
<td>17</td>
<td>16;8;p04x;10;3;p03x;14;5;p02x;18;9;p09x;8;1;p05x;</td>
</tr>
<tr>
<td>18</td>
<td>19;6;p01x;22;5;p05x;21;3;p09x;11;4;p02x;8;10;p04x;</td>
</tr>
<tr>
<td>19</td>
<td>25;5;p09x;1;9;p01x;9;4;p04x;11;1;p05x;12;2;p03x;</td>
</tr>
<tr>
<td>20</td>
<td>12;1;p02x;18;9;p03x;13;4;p09x;11;2;p01x;18;2;p05x;</td>
</tr>
</tbody>
</table>
References


CALKINS, M. W. 1898. Short studies in memory and in association from the Wellesly College Psychological Laboratory. 5(5), 451-462.


GARFINKEL, S. L. 2005. Design principles and patterns for computer systems that are simultaneously secure and usable. PhD, Massachusetts Institute of Technology.


GOLDBERG, J., HAGMAN, J. & SAZAWAL, V. 2002. Doodling our way to better authentication. CHI ’02 extended abstracts on Human factors in computing systems. Minneapolis, Minnesota, USA: ACM.


ZIMMERMANN, P. 1995. The official PGP user's guide, MIT Press, Boston, Massachusetts, U.S.A.