METHODS AND TECHNIQUES TO PROTECT AGAINST SHOULDER SURFING AND PHISHING ATTACKS

PEI PEI SHI

A THESIS

IN

THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN INFORMATION SYSTEMS SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

AUGUST 2010

Your file Votre référence
ISBN: 978-0-494-71049-4
Our file Notre référence
ISBN: 978-0-494-71049-4

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Pei Pei Shi**

Entitled: **Methods and Techniques to Protect Against Shoulder Surfing and Phishing Attacks**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Information Systems Security

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

_____ Examiner

_____ Examiner

_____ Examiner

_____ Supervisor

_____ Co-supervisor

Approved _____

Chair of Department or Graduate Program Director

_____ 20 _____ _____

Dr. Nabil Esmail, Dean

Faculty of Engineering and Computer Science

ABSTRACT

Methods and Techniques to Protect Against Shoulder Surfing and Phishing Attacks

Pei Pei Shi

Identity theft refers to the preparatory stage of acquiring and collecting someone else's personal information for criminal purposes. During the past few years, a very large number of people suffered adverse consequences of identity theft crimes.

In this thesis, we investigate different methods and techniques that can be used to provide better protection against identity theft techniques that have some hi-tech relevance such as shoulder surfing of user's passwords and personal identification numbers (PINs), phishing and keylogging attacks.

To address the shoulder surfing threat to traditional PIN entry schemes, two new PIN entry schemes are proposed. Both schemes achieve a good balance between security and usability. In addition, our analysis shows that these two schemes are resilient to shoulder surfing, given that the attacker has a limited capability in recording the login process.

We also propose a click-based graphical password authentication scheme. This scheme aims at improving the resistance to shoulder surfing attacks while maintaining the merits of

the click-based authentication solutions. It is also resilient to shoulder surfing attacks even if the attacker can record the entire login process for one time with a video device.

Finally, in order to defend against online phishing attacks, we present a framework to strengthen password authentication using mobile devices and browser extensions. The proposed authentication framework produces a different password depending on the domain name of the login site. Besides defending against phishing attacks, this solution does not require any modifications at the server side.

Acknowledgments

First of all, I would like to thank my supervisors, Dr. Bo Zhu and Dr. Amr Youssef. I gratefully acknowledge their prompt, insightful, valuable, and frank feedback to my research. It is their enthusiasm and knowledge that inspired me throughout my graduate studies.

I would like to thank all my colleagues at the Concordia Institute for Information Systems Engineering, especially Benwen, Hongying, Shuai, Thomas, and Rabeah.

I also appreciate the help, companionship, understanding, and support of all my friends.

Last, but certainly not least, special thanks to my parents and younger sister for supporting me throughout these two years. Their unconditional love is what really helped me to get to where I am now.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

According to the Royal Canadian Mounted Police (RCMP), identity theft refers to the preparatory stage of acquiring and collecting someone else's personal information for criminal purposes [44].

During the past few years, online identity theft became a very serious problem that made millions of people all over the world suffer many kinds of adverse consequences. Unfortunately, there are diverse techniques that can be adopted by online identity thieves to obtain and exploit personal information. Consequently, it is not an easy task to propose a single solution to defend against all of these attacks.

In this thesis, we investigate different methods and techniques that can be used to provide better protection against shoulder surfing and phishing attacks. In particular, we focus on protecting user's passwords and personal identification numbers (PINs). Despite the fact that many security issues have been reported against password-based authentication protocols and many authentication protocols have been proposed to replace traditional passwords authentication, the widespread application of these protocols still has a long way to go. This is partially due to the fact that unlike other complex authentication schemes,

such as zero-knowledge proofs [37], that may require high complexity to deploy and use, password-based authentication is relatively very cheap and easy to use and deploy. Nowadays, most sites such as banks, e-mail service providers, online stores and social networking sites, are still using password-based authentication techniques and this is very unlikely to change in the near future. Thus passwords and PINs will remain the hot targets of identity thieves.

1.2 Motivation and Objectives

A recent report [4] indicated that 11.1 million people in US were victims of identity fraud in 2009. The associated cost of these reported cases were estimated at about $54 billion. Similarly, based on the report released by United Kingdom fraud prevention service, there were 85,000 victims of impersonation and 24,000 victims of bank account takeovers in 2009. Compared to 2008, these two numbers increased by about 35% and 15%, respectively. Based on the report by McMaster eBusiness Research Center (MeRC) [14], almost 1.7 million Canadians were the victims of some kind of identity fraud in 2007. Over 20 million hours and more than $150 million were spent in order to resolve these frauds. All these startling numbers reveal the importance and urgency of the battle against identity theft.

The security of most payment services at current terminal applications, such as Automatic Teller Machine (ATM) and Point of Sale (PoS), are usually protected by a combination of certain unique information stored on a physical device, typically a magnetic stripe card, and certain secret knowledge (e.g., PIN). A common risk of this type of protection is that magnetic stripe cards can be stolen or skimmed by fake card readers. Once it happens, the security of the authentication scheme relies only on the protection of the PIN. Unfortunately, recent reports [3, 6, 56] show that shoulder surfing attacks are already used by criminals to obtain the PINs, and thus break into the accounts. This class of shoulder

surfing attacks has become a serious threat to many security applications that rely on the combination of magnetic stripe cards and PINs to perform the authentication.

Unfortunately, terminal applications are not the only targets of attackers; current online authentication systems suffer even more security risks. Phishing scam is one of the most popular and effective methods adopted by attackers to collect sensitive information/credentials from legitimate users. Sometimes, a forged website and/or a forged e-mail claiming to originate from a given financial organization is usually enough to trick a novice, and thus the cleartext password for the legitimate site is acquired by the attacker.

Over the last few years, various graphical authentication schemes were proposed as alternatives to current text-based password authentication systems. This trend draws a lot of attention from both academia and industry. The fast development of graphical authentication is based on psychological studies, which indicate that the human brain is better at recognizing and recalling visual information than verbal or textual information [40,51,71]. Researchers hope that this cognitive capability of people would allow the development of more usable and secure authentication solutions.

Throughout this work, we focus on proposing and analyzing solutions to prevent or mitigate shoulder surfing threats to terminal applications (e.g., ATM and PoS), shoulder surfing issues of click-based graphical passwords and phishing attacks to online authentication systems.

The objectives of this work can be summarized as follows:

- Develop and evaluate new techniques to mitigate shoulder surfing attacks against ATM and PoS terminal applications.
- Design, implement and evaluate a click-based graphical authentication scheme resilient to shoulder surfing attacks.

- Design and develop an online authentication solution to protect against phishing attacks.

1.3 Contributions

In this section, we briefly outline our contributions in the following three aspects: (1) shoulder surfing threats to terminal applications; (2) shoulder surfing issues of click-based graphical passwords; and (3) phishing attacks against online authentication systems. In particular:

- We propose two new PIN entry schemes (namely, *Position-based PIN Entry Scheme* and *Rotary PIN Entry Scheme*) to address the shoulder surfing threat. Both schemes achieve a good balance between security and usability. In addition, our analysis shows that these two schemes are resilient to shoulder surfing, given that the attacker has a limited capability in recording the login process.

- We propose a click-based graphical authentication scheme inspired by Cued Click Points (CCP) [9]. The scheme aims at improving the resistance to shoulder surfing attacks while maintaining the merits of the CCP style click-based authentication solutions. The proposed solution is resilient to shoulder surfing attacks even if the attacker can record the entire PIN entry process one time with a video device.

- We design and implement a framework aiming at combating phishing and shoulder surfing attacks while requiring no modification to the server side.

Some of the results presented in this thesis were partially published in [52, 53].

1.4 Thesis Organization

The rest of this thesis is organized as follows. Shoulder surfing, phishing and keylogging attacks are reviewed in the next chapter. In chapter 3, we present two PIN entry schemes which are resistant to shoulder surfing attacks. Then, a graphical authentication scheme is presented in chapter 4, followed by a solution to strengthen password authentication using mobile devices and browser extensions in chapter 5. Finally, in chapter 6, we present a summary of our research and some possible future research directions.

Chapter 2

Identity Theft Techniques

2.1 Introduction

There are several techniques for obtaining and exploiting personal information for identity theft. Examples for these techniques include rummaging through rubbish for personal information (dumpster diving), retrieving personal data from redundant IT equipments and disposed or carelessly placed storage media, using public records, stealing identification cards, passports, or bank cards, typically by pickpocketing.

In this thesis we only focus on identity theft techniques with some hi-tech relevance. In particular, we focus on shoulder surfing, phishing and keylogging. Countermeasures for shoulder surfing are presented in chapters 3 and 4, while a countermeasure against online phishing and keylogging is given in chapter 5. Other threats such as viruses and spywares are out of the scope of this work.

The rest of this chapter is organized as follows. Shoulder surfing is reviewed in section 2.2. In section 2.3, we introduce techniques used in phishing attacks as well as damages caused by phishing scams. Keylogging techniques and its damages are discussed in section 2.4.

2.2 Shoulder Surfing

As the name implies, shoulder surfing is used to obtain passwords, PINs or other sensitive information through observation, such as by looking over someone's shoulder. It is a simple, yet effective, method to get information especially in crowded places such as coffee-shops where it is relatively easy to stand next to or behind someone without being noticed.

Protection against shoulder surfing has received a lot of attention by researchers since the beginning of 21^{st} century, especially with the widespread use of ATMs and PoSs, where many solutions are proposed in order to mitigate shoulder surfing issues in different scenarios.

Obviously, the effectiveness of shoulder surfing attacks is highly dependent on the shoulder surfer's capability to record the victim's login process. In this thesis, we distinguish two types of shoulder surfing attacks based on the recording capabilities of the adversaries. Shoulder surfing are classified as *cognitive shoulder surfing* and *recording-based shoulder surfing*. In the former, the shoulder surfer depends on the shoulder surfer's cognitive capability to retain PINs or other sensitive information. This type of attacks is relatively easy to defend against due to the limited cognitive capabilities of human beings [38, 60]. In contrast, in the latter, the accuracy of recording sensitive information has been significantly improved with the help of concealed miniature cameras or video mobile phones. It should be noted that recording-based shoulder surfing is not a pure theoretical possibility. In fact, recent reports [3, 6, 56] show that recording-based shoulder surfing attacks have already been used by criminals, which present a serious threat to many security systems.

2.3 Phishing

The Anti-phishing working group (APWG) [1] defined phishing as follows [2]:

> Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. Social-engineering schemes use spoofed e-mails purporting to be from legitimate businesses and agencies to lead consumers to counterfeit websites designed to trick recipients into divulging financial data such as usernames and passwords. Technical-subterfuge schemes plant crimeware onto PCs to steal credentials directly, often using systems to intercept consumers online account user names and passwords - and to corrupt local navigational infrastructures to misdirect consumers to counterfeit websites (or authentic websites through phisher-controlled proxies used to monitor and intercept consumers' keystrokes).

The phishing technique was firstly described in detail in a paper and presentation delivered to the International HP Users Group, Interex [12] in 1987. The first recorded mention of the term "phishing" is on the *alt.online-service.America-online* (Usenet newsgroup) [43] on January 2, 1996. From that time, researchers have started the war with this attack. However, nowadays, phishing is still one of the most prevalent attacks on the Internet. For example, APWG reported 49084 unique phishing websites detected in June 2009, which is the second-highest number recorded since APWG began reporting.

2.3.1 Phishing Techniques

Phishers are able to "Phish" for users' personal information in various ways, of which fraudulent e-mails claiming to be from users' banks or some financial institutions are the most common. Usually, a user is informed that the account has a problem which needs

to be resolved immediately. A malicious link is usually included in the e-mail. When the user clicks on it, the victim is directed to a spoofed website which is genuinely looking the same as the legitimate one. When the user enters account information (e.g., username and password), these credentials are relayed to the phisher.

Another way to perform phishing attacks is through telephones which does not require a fake or spoofed website at all. In this attack scenario, short messages claimed to be from banks or financial institutions are sent to users in which users are informed that they should call a number regarding problems with their accounts. When users dial the number, they are asked to enter their credentials. Some phishers even use fake caller-ID through VoIP to deceive users that calls are from a trusted organization [54]. Instant Messages is also an option for phishing.

Besides these traditional phishing approaches, phishing attacks are continuously evolving into more sophisticated forms. A recent tendency is that attacks have been directed specifically at senior executives and other high value individuals within financial institutions. This kind of attacks, termed *spear-phishing* [35,46], usually tries to deceive victims into installing malware or disclosing valuable credentials about their organizations. In order to convince the victims more easily, advanced phishers include some exploited information of victims into their phishing scams [27]. As a result, this so called *context-aware phishing* attack, dramatically enhances the probability of success, from 3% percent for a common attack to 48-96% for a deliberately spoofed context-aware attack [27]. Social networks present another hot target for phishers where they can easily obtain a large amount of information about thousands of individuals. This information can be harvested later on in order to craft e-mails claiming to come from someone the victim knows or even trusts.

Since phishing e-mails and phishing websites are most common among these phishing techniques, we will address these two methods used by these attacks in the following two subsections.

2.3.2 Phishing E-mail Methods

The objective of a phishing e-mail is to entice the receiver to read the e-mail and click on the link inside it. In order to achieve this goal, some methods are adopted by phishers, such as forging the address of the sender, enticing subject lines, using genuine looking content and disguised links [25]. In what follows, we briefly summarize these methods.

- **Spoofed senders' address.** To forge the address of a sender is quite easy nowadays. A forged e-mail address, which looks like the one it claims to be, is usually used by phishing e-mails. For example, the forged address *service@rbc.ca* can be used in order to convince the recipient that this e-mail comes from the Royal Bank of Canada.

- **Beguiling subject lines.** Phishing e-mails usually try to have an attractive subject line which seems to originate from the related financial institutions. Apparently, the intention is to urge the user to open this e-mail. For instance, a beguiling subject line may look like this: *You have one new message from Royal Bank of Canada.* Besides, some phishing e-mails use misspelled key words to bypass spam filters, which some users may not identify when they quickly scan the subject line.

- **Genuine looking content.** Normally, phishers copy the images, logos and text styles from the legitimate website in order to make their phishing e-mails look genuinely. Moreover, some of the phishing e-mails may even have genuine links to the institution's term of use, privacy policy and other pages on the legitimate website, which deceives some unsuspecting users. Some trusts and authentication marks are also copied to convince receivers the authenticity of the e-mail.

- **Disguised links.** A deliberately disguised link, which is very significant, is usually provided in the phishing e-mail aiming at deceiving the recipient. If HTML

is enabled in the e-mail, this link is displayed as a genuine Uniform Resource Locator (URL). However, when the user clicks on it, it redirects the user to a different website. For example, a hyperlink in a phishing e-mail that I have recently received is *RBC Royal Bank ONLINE*. When I clicked on it, I was redirected to http://feltd.gotadsl.co.uk/royal.html. In text only e-mails, spoofed links contain the "@" symbol. For example, a spoofed URL may look like http://www.genui nesite.com@phishing.com. This URL takes the victim to http://www.phishing.com instead of http://www.genuinesite.com.

- **E-mail form.** Instead of providing a disguised link, some phishing e-mails contain a form for recipients to enter their personal information or account information and click the "submit", "send" or "update" button. Forms within e-mails are used together with script language located on a remote server to receive the submitted information and either forward the information to the phishers, or save the information in a database for the future use.

2.3.3 Phishing Website Methods

The objective of phishing attacks is to acquire personal information, especially financial credentials. In order to achieve this goal, phishers have to convince users that they are dealing with genuine sites since users will not disclose their credentials to untrusted sites or organizations. Consequently, phishing sites which support the phishing e-mails usually mirror the claimed legitimate sites. Several methods are utilized by phishers to achieve this goal, such as disguising the URL in the address bar and using genuine looking images and text [26].

- **Genuine-looking URLs.** Some phishing websites simply register a domain name similar to the one of the targeted organization or institution. In this case, these two URLs look like each other. As a result, some users may not be able to identify the

spoofed one from the genuine one. For instance, phishers may use the domain name of http://www.g**ooo**gle.com, where the actual domain is http://www.google.com.

- **Fake address bar.** Some phishers utilize a more technical way to deceive users. This method involves the removal of the address bar and building a fake address bar using images and text with the help of script languages. When a user clicks on the URL in the phishing e-mail, a new browser window is opened without the original address bar, and sometimes without the status bar. A false address bar constructed by HTML, XML User Interface Language (XUL) and JavaScript is displayed in place of the original one. The straightforward way to defend against this class of phishing attacks is to disable scripts, from browser settings, which is usually not a very practical solution since most sites need these scripts.

- **Genuine looking content.** Phishing websites usually utilize copied images, logos and text styles of the legitimate site to make the phishing site looks the same as the genuine one. Moreover, some of the phishing pages even have genuine links to the target institution's term of use, privacy policy, products, and other pages from the legitimate website. Furthermore, some trusts and authentication marks (e.g., Secure Sockets Layer (SSL) locks on the status bar) are also copied to convince the victims the authenticity of the phishing site.

- **Pop-up windows.** When this method is utilized by the phisher, a genuine webpage is opened in the background by script. Meanwhile, a small bare pop-up window (without address bar, status bar and scrollbars) is also opened by script on top of the genuine webpage. Through this, phishers attempt to convince users that these two windows are associated with each other, i.e., they both belong to the same organization. As mentioned above, it is hard to defend against this technique by disabling scripts.

2.3.4 Damages Caused by Phishing Scams

Generally, the ultimate objective of phishing scams is financial benefit. Several ways are used by phishers to achieve this financial goal using the collected personal information. In what follows, we briefly review some of the possible damages caused by phishing attacks [24].

- **Hijacking user accounts.** In here, accounts do not necessarily mean bank accounts. E-mail accounts and social network site accounts can also be the valuable targets to attackers. Phishers handle information gathered by different accounts in diverse ways. If the information of a given bank account is provided by the victim, the fraudsters may empty the victim's bank account, e.g., by electronically transferring money to a temporary account created by them using someone else's personal information. And then, withdraw the money from that account before being noticed by the victim. Cashing fake cheques on the victim's account is another option for the phisher.

- **ATM card duplication.** Some phishing websites require users to provide information about their ATM cards. With this information, it is possible for phishers to duplicate ATM cards which are linked to victims' debit accounts. When this is done, the victims' account can be emptied by ATM withdrawals.

- **Unauthorized use of credit cards.** If the victim provides detailed information about the credit card to the phisher, it is very easy for the phisher to make unauthorized purchases through the Internet.

- **Identity theft.** Even if users provide their non-financial accounts information or just personal information to phishers, it is still dangerous because phishers can apply for credit cards using this stolen information, sell information to criminal organization or launch spear-phishing attacks with the help of this information.

2.4 Keylogging

Keylogging, sometimes called keystroke logging, is the practice of tracking or monitoring each keystroke, or mouse movement, of users on a given computer. Usually, this tracking is done in a covert manner so that the victim is unaware of it. Keylogging can be done through dedicated hardware devices or through pure softwares.

2.4.1 Hardware-based Keyloggers

As a hardware device, a keylogger is a small sized plug which works as a connector between the keyboard and the computer. This kind of device usually looks like an ordinary keyboard plug, thus users may not pay much attention to it. All hardware keylogger devices have the following two components [67]:

- **Microcontroller.** This component is intend to interpret every keystroke typed by the user, process it if necessary and then save it to the non-volatile memory.
- **Non-volatile memory.** This part is intend to save data passed by microcontroller, and retain it even when power is off. Flash memory is a good choice for this module.

Compared to software-based keyloggers, hardware-based ones can track any user input from the time a computer is turned on. Consequently, even the password for the basic input/output system (BIOS) or disk encryption software can also be interpreted. Usually, a pre-defined character string is needed to access the logged data which is saved in internal memory of the hardware by the keylogger. Typically this memory may range from a few kilobytes to several megabytes, and usually each keystroke recorded cost a byte of memory. In here, we describe few kinds of hardware keyloggers [67].

- **Keyboard hardware.** This kind of keylogger is usually a hardware circuit that is attached somewhere between the computer keyboard and the computer (e.g., keyboard's cable connector). Some of these implementations are designed to have an

innocuous appearance, while others even can be installed or built into standard keyboards and consequently will not be visible externally. An internal memory is equipped into these types of keyloggers aiming at recording all keystrokes. The saved keystrokes can be accessed subsequently, for instance, by inputting a pre-defined secret password or key sequences [29].

- **Firmware-based.** In this case, the computer's BIOS, which is typically responsible for handling keyboard events, is modified to record these events when they are processed.

- **Wireless Keylogger sniffers.** This type of keylogger is designed to collect packets of data being transferred from a wireless keyboard to the computer. If the communication channel between a wireless keyboard and a computer is secured by encryption, the deployed encryption needs to be cracked in order to acquire the collected information.

- **Keyboard overlays.** A recent report [31] has shown that criminals have utilized keyboard overlays on ATM machines to record users' PINs. Each keypress is registered by this malicious device as well as the legitimate one that the user is using.

2.4.2 Software-based Keyloggers

Software-based keyloggers are programs which can either be downloaded to a particular computer by someone who is intend to monitor the users of this computer, or downloaded as a spyware and executed as part of a rootkit or trojan horse while the legitimate user is unaware of its existence. A keylogger program typically has two parts: a dynamic link library (DLL) and an executable file (.EXE) [50]. All the recording workload is done by the DLL while the executable file is designed to install the DLL and trigger it to work. Similar to the hardware-based keyloggers, each keystroke typed by the user is recorded by this program

and uploaded to the owner of this keylogger through a covert channel over the Internet. This covert communication can be achieved in different ways, for example, recorded data can be uploaded to a website, database or emailed to a pre-defined e-mail address. From a technical perspective, we can classify keylogger programs into five categories [68]:

- **Hypervisor-based.** In this case, keylogger program can theoretically reside in a malware hypervisor which is running underneath the operating system. Blue Pill is a conceptual example [65].

- **Kernel-based.** This type of keyloggers resides at the kernel level and, thus, is difficult to be detected, especially for user-mode applications. These keylogger programs are usually implemented in the form of rootkits which are very powerful since they can subvert the operating system kernel and gain unauthorized access.

- **Hook-based.** The idea of this type of keylogger is to hook the keyboard using the functionality provided by the operating system for some applications subscribed to keyboard events. When this hook action is successfully done, each keypress is notified by the operating system to the keylogger, and the latter simply records it.

- **Passive methods.** Some operating system APIs (e.g., GetAsyncKeyState()) are used in this kind of keyloggers to poll the state of the keyboard or to subscribe to keyboard events. These keylogger programs are relatively easy to write, but easy to be detected since they can cause a noticeable increase in Central Processing Unit (CPU) usage as a result of the constant polling of each keypress.

- **Web form Grabbers.** Instead of recording each keystroke of the keyboard, web form grabbers keyloggers record the web browsing "onSubmit" event functions in order to get the web form submissions. This method records data before it is passed over the Internet.

Besides the methods mentioned above, some keyloggers can be augmented with additional capabilities such as:

- **Clipboard logging.** These keylogger programs have the ability to record anything that has been copied to the clipboard.

- **Screen grabbers.** In this case, keylogger programs can take screenshots to capture graphics-based information. Some of them may take the whole screen while others just take partial screen around the mouse clicks. Conceptually, these keylogger programs are similar to the record-based shoulder surfing.

The ultimate goal of keylogging is not merely collecting personal information. Similar to phishing attacks, stealing personal information is just the first step. Keyloggers are mainly financially-motivated. There are even some keyloggers that are identified as phishing-based keyloggers, and a significant increase of this class of key loggers has been reported by APWG [20].

Most of the ways mentioned in section 2.3.4 can be adopted by attackers using keyloggers in order to achieve financial benefits from the stolen personal information.

Chapter 3

PIN Entry Schemes Resistant to Shoulder Surfing

3.1 Introduction

Authentication mechanisms play a central role in the protection of systems that offer personalized financial services such as ATMs and POSs. Currently, the security of ATMs and POSs are protected by a combination of a physical token, typically a magnetic card, and certain secret knowledge (e.g., PIN). Both of these factors are needed for successful authentication to such systems. A common risk of this type of protection is that magnetic stripe cards can be stolen or skimmed by fake card readers. Once it happens, the security of the authentication scheme relies only on the protection of PINs, which is subjected to the different types of attacks discussed in chapter 2. In this chapter, we investigate the design of PIN entry schemes that are resistant to shoulder surfing attacks.

To address the shoulder surfing issues, several challenge-response-based approaches are proposed [48, 61, 64]. In order to protect users' PINs, these schemes usually require users to interact with systems multiple rounds and do not provide direct feedback. As a result, the login procedure can be very long [61, 64], which makes these schemes unsuitable for

many practical scenarios. Hence, it is desirable to design solutions applicable in scenarios where both security and reasonably fast response time are important.

Further security protection beyond the combination of a magnetic stripe card and PIN includes biometric authentication [33] and one-time keypad [49,58] which are quite widely used. However, there also exist specific types of attacks aiming at these solutions. For instance, many commercial fingerprint scanners can be reliably fooled using a combination of cheap kitchen supplies and a digital camera [34]. Detailed discussions about biometric authentication and one-time keypad are out of the scope of this thesis.

Throughout the rest of this chapter, we present two new PIN entry schemes that are designed in order to mitigate shoulder surfing attacks.

- **Position-based PIN Entry Scheme.** In this scheme, the PIN is not a sequence of 4 digits but a sequence of 4 locations chosen in a specific table displayed to the user. Compared to previous work, our scheme achieves a good balance between security and usability. Our analysis shows that this design can offer a strong security protection against adversaries with limited surfing capability, e.g, when the adversary is limited to acquiring only up to two records of the whole login process. The usability issue is addressed from two aspects: the time required by the authentication process and the interface design. Our scheme has a short response time and yet still provides a strong protection against shoulder surfing attacks.

- **Rotary PIN Entry Scheme.** This rotary PIN entry scheme has a few variants. Each variant provides the option of balancing security conditions against two types of potential risks, namely, *random guessing* and *shoulder surfing assisted guessing*. More importantly, a thorough security analysis is conducted in order to give a quantitative guidance on choosing an appropriate option according to a system's specific security requirements. Unlike previous work, instead of emphasizing the robustness of a specific scheme against a particular type of attack, we indicate that there is a trade-off

between protections against different types of risks, and thus the selection of an appropriate scheme should be application-dependent, i.e., by considering the types and frequencies of risks faced in the application. A usability study is also included.

The rest of this chapter is organized as follows. Related work is reviewed in section 3.2. The details of our schemes are presented in section 3.3, followed by security analysis in section 3.4 and section 3.5. In section 3.6 and section 3.7, we discuss the usability issues of the proposed schemes.

3.2 Related Work

Roth *et al.* [48] proposed a PIN entry method which offers limited resilience against shoulder surfing. In their method, the keys on the keypad are randomly partitioned into two sets, one set is white while the other is black. Depending on the set to which the digit of the PIN under verification belongs, the user enters the key with the same color as the set. Multiple rounds are needed to complete the input of a single digit of the PIN. The same process is repeated until the whole 4-digit PIN is entered. Their method has three variants: *Immediate Oracle Choices* (IOC), *Delayed Oracle Choices* (DOC) and *probabilistic cognitive trapdoor game*. The first two are secure against cognitive shoulder surfing, but are vulnerable to recording-based shoulder surfing. The *probabilistic cognitive trapdoor game* variant offers limited resilience to recording-based shoulder surfing. It aims at the case where a shoulder surfer may acquire one record of the whole login process.

Weinshall [61] proposed two challenge response protocols, one with a high complexity query and the other with a low complexity query. In both protocols, users need to answer a sequence of challenges posed by the login terminal. The challenges are based on a shared secret between the login terminal and the user, which is a fixed set of pictures divided into two sub-groups. As indicated by the authors, this scheme has two weaknesses: users

need to be trained in order to familiarize them with their associated secrets, e.g., a set of pictures; and the authentication process may take longer time than alternate methods. A later work by Golle and Wagner [18] shows that Weinshall's protocols are vulnerable to a certain attack that leverages a SAT solver. This attack can recover the user's PIN in few seconds after recording only a small number of successful logins.

The Convex Hull Click Scheme (CHC) [64] is another multiple-round challenge-response authentication scheme proposed to defend against shoulder surfing. In this scheme, users needs to select a set of icons, out of a larger number of icons, as their password icons. During the login process, to respond to the challenge, the user must virtually find three or more of her password icons, mentally create a convex hull formed by these icons, and then click inside this convex hull. The user must respond to the multiple challenges correctly in order to be authenticated to the system. As a result, the login time can be very long. According to their simulation results, the mean time for correct password inputs is around 72 seconds.

PassFaces [42] relies on the user's capability of recognizing faces. The user first needs to choose a set of faces as the secret, and have a few minutes of training to be familiar with the chosen faces. The login may take several rounds. For each round, among all the faces displayed only one belongs to the secret set. The user needs to identify those faces in the secret set correctly in order to pass the authentication. This method is designed to be resilient against cognitive shoulder surfing.

3.3 Proposed PIN Schemes

3.3.1 System and Adversary Models

In this chapter, we assume that the verifier (e.g., the ATM) is trusted to perform the authentication process correctly, and the alphabet of PIN digits is {0, 1,..., A-1}, e.g., {0, 1,..., 9}, although it can be readily extended to generic characters instead of digits. Let L denote the

number of digits in the secret PINs. Throughout the rest of this chapter, the default value of L is 4. As to the capability of adversaries, we assume that they can obtain the information stored on the magnetic stripe card (e.g., by pickpocketing or by fake card readers) and thus the protection of the authentication mechanism relies solely on the security of PINs.

Below, we consider three adversary models. The first type is called a *Zero-knowledge adversary*, in which the adversary only has the card (e.g., picking up a card by chance), but does not have any knowledge about the PIN. Thus the only option available for the adversary in this case is to launch a random guessing attack. The other two considered adversary models are cognitive shoulder surfing and recording-based shoulder surfing. In particular, in recording-based shoulder surfing, we assume that the adversaries can acquire only one record of the entire login process. We argue that such an assumption is reasonable in most applications that are protected by the combination of a magnetic stripe card and a PIN. For example, even if the adversary can install a concealed miniature camera over the input screen of an ATM, the probability that the camera can record two or three login processes of the same user on the same ATM within a reasonable time frame is practically slim.

We assume that the verifier keeps a record on the number of successive failed inputs. After three failed attempts, the verifier will retain the card and terminate the usage of the related account on any ATM, until the PIN is reset through a secure channel, e.g., at a bank branch. In addition, for convenience, this counter is automatically reset upon a successful login. We are aware of the fact that this rule may be misused to launch discontinuous attacks. More specifically, a smart attacker may first make two guesses. If the attacker fails, she will wait until the counter is reset, before trying another two. Hence, in our design, the legitimate user will be notified by previous failed login attempts (if any) upon a successful login. As a result, the maximum number of retries that the adversary can make without being detected is two.

Throughout the rest of this chapter, we focus our discussion on the ATM environment, although our scheme is also be applicable to other scenarios such as PoS terminals and portable digital assistants (PDAs).

3.3.2 Position-based PIN Entry Scheme

For each login, after the user inserts her bank card into the ATM, the machine displays an $A \times A$ table, each cell of which contains a number from the set $\{0,\ 1, \ldots, A-1\}$. The assignments of numbers to cells are executed in such a way that A copies of every number in the set are randomly assigned to the cells in the table. Since the PIN in our scheme is a sequence of 4 positions, in the first step, the user needs to input the number displayed within the cell corresponding to the first position. Afterwards, a new $A \times A$ table is generated, and the second number is keyed in according to the second position. The same process is repeated until the user has input 4 numbers. At the end of the login process, if the user inputs all the numbers correctly according to the secret, i.e., the sequence of 4 positions, she is authorized to access the account.

The assignments of numbers to the cells in any two displays are completely independent. Note that, although the whole login process consists of four steps, unlike previous work [48, 61, 64], there is only one-round selection per PIN digit in our scheme.

Besides the random assignments of numbers, during the generation of tables, we also consider introducing a colorful display to improve the usability of our scheme, which is discussed in section 3.6. Since the distribution of colors are determined independently from the assignments of numbers, security analysis presented in section 3.4 is applicable to both the general scheme without colors and the colored designs presented in section 3.6.

3.3.3 Rotary PIN Entry Scheme

Here, the intension is to design a secure yet user friendly PIN entry scheme. In addition, in order to provide different options for balancing security levels against zero-knowledge adversary and shoulder surfers, our scheme contains two parameterized variants: Variant One ($VO(N)$) and Variant Two ($VT(N)$).

Variant One

Suppose that the user has inserted the bank card into an ATM, and her personal PIN of the card is $X_1X_2X_3X_4$, ($X_i \in \{0, 1,\ldots, \text{A-1}\}$). In this variant, after the welcome message, the verifier displays a set of L co-centered rotating wheels each of which is equally divided into A sectors as shown in Figure 1. Each sector is marked with a digit randomly chosen from the alphabet, i.e., 0 to $A-1$, and each digit in the alphabet is displayed exactly once within each circle. Circles can be rotated either clockwise or anticlockwise. We denote the four circles from the innermost to the outermost as the 1^{st}, 2^{nd}, 3^{rd} and 4^{th} circle, respectively. To input the PIN, the user needs to align all PIN digits along one sector in the correct sequence, i.e., align X_i in the i^{th} circle. A control button is pressed to indicate the completion of the PIN alignment process.

Upon the completion of the PIN entry process, there are A aligned inputs contained in the wheel. For example, assume that Figure 1 shows the final input of a PIN entry process, the aligned inputs include {8121,9243,7774,5838,3985,6607,2496,1012,4559,0360}. The verifier checks all these potential PINs. If any of them matches the genuine PIN corresponding to the bank card inserted, the verifier will give the user access to the linked account. Otherwise, an error message is displayed, and the counter of the number of successive PIN entry failures is increased by 1.

The above scheme, called $VO(1)$, can be generalized into $VO(N)$ by dividing each circle into $N \times A$ sectors, instead of A sectors. In this case, each digit in the alphabet is

displayed exactly N times, instead of once, within each circle.

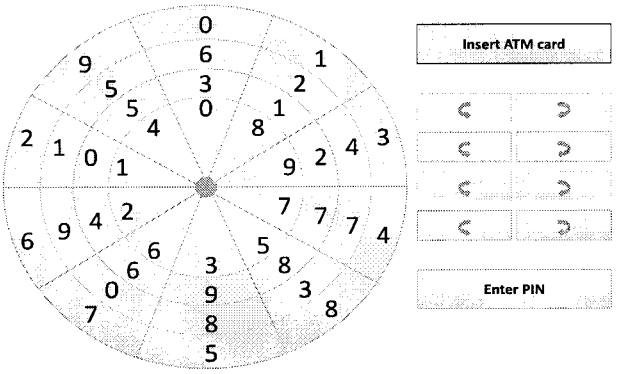


Figure 1: Example for $VO(1)$

Figure 2 shows an alternate graphical user interface (GUI) for $VO(1)$ that might be appropriate in some applications.

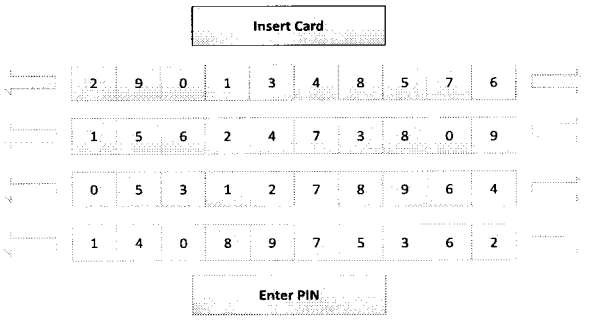


Figure 2: An alternate GUI for $VO(1)$

Variant Two

Similar to VO, each circle is equally divided into A sectors, and each digit in the alphabet is displayed exactly once within a circle. However, the wheel in VT contains only two circles,

as shown in Figure 3. As a result, in order to input a 4-digit PIN, VT needs two rounds to complete the PIN entry process. More specifically, X_1X_2 and X_3X_4 are entered in round one and round two, respectively. Note that, the position of the aligned input X_1X_2 is not necessary to be the same as that of the aligned input X_3X_4.

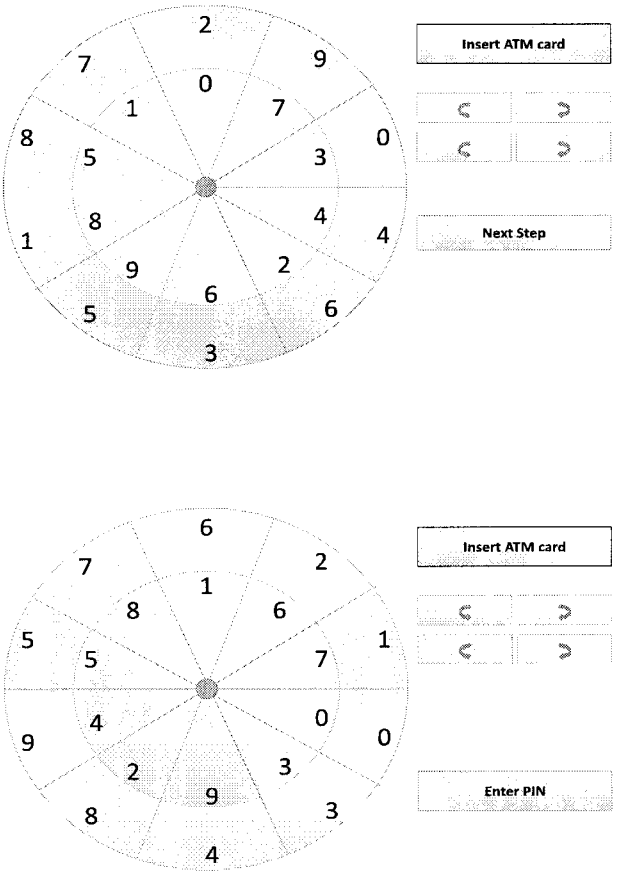


Figure 3: Example for VT(1)

After the two rounds, the verifier checks all A potential PINs and verifies whether there exists an input equal to the genuine PIN. Based on the result of the verification, the verifier will grant or deny the access to the corresponding account.

Again, $VT(1)$ can be generalized in the same as $VO(N)$ by dividing each circle of the

wheel into $N \times A$ sectors, instead of A sectors. Then each digit in the alphabet is displayed exactly N times, instead of once, within a circle. Due to the limitation on the display area of an ATM machine, the typical setting of N is 1, 2 or 3.

3.4 Security Analysis of the Position-based PIN Entry Scheme

3.4.1 Cognitive Shoulder Surfing

There has been some interesting research on the cognitive capabilities of human beings. In 1956, Miller [38] noted that the limitation on short term memory (STM) is 7 plus or minus 2 symbols. A more recent work by Vogel *et al.* [60] shows that STM of normal people is limited to three to four symbols. Few subjects were able to remember five symbols in their STM throughout the experiments conducted in [60]. This discovery is based on the neurophysiological evidence.

For the position-based PIN entry scheme, there are A×A numbers on the ATM screen per display/step and only one number is input by the user at a time. If we define the combination of the number that a user inputs at one step and the corresponding A positions as a *knowledge set*, then the whole login process generates 4 knowledge sets. According to previous research in human's cognitive capabilities [38, 60], remembering all the knowledge sets are far beyond the cognitive capabilities of a typical human being adversary.

3.4.2 Recording-based Shoulder Surfing

Compared to cognitive shoulder surfing, the adversary is much more powerful in recording-based shoulder surfing. Some schemes aiming at the former [42] can be trivially broken by a recording-based shoulder surfing even when the adversary has a relatively limited recording capability. In this section, we concentrate on security analysis of our scheme for scenarios in which the shoulder surfer can obtain up to two records of the whole login

process.

One Login Record

If one record of the whole login process is available, a shoulder surfer can limit the guess within the knowledge sets. More specifically, the shoulder surfer randomly chooses a position from the set of positions assigned with the same value as the input. For each position of the PIN, the probability of making a correct guess is $1/A$. Thus, the probability of identifying the whole PIN is $1/A^4$, which ranges from 0.16% to 0.01% when the value of A varies from 5 to 10.

Two Login Records

We first evaluate the probability that a shoulder surfer can successfully guess the first position of the PIN, given that she can obtain two records of the whole login process. Let us denote this probability as P_1.

In our design, each number in the set (i.e., $\{0,\ 1, \ldots, A-1\}$) is repeated A times in a table. Given that the user entered a number x for the first position during a login process, the shoulder surfer can deduce that the actual position is among the set of A cells assigned with the number x by checking the corresponding record. Let $L_1 = \{l_1^0, l_1^1, \ldots, l_1^{A-1}\}$ and $L_2 = \{l_2^0, l_2^1, \ldots, l_2^{A-1}\}$ denote the sets of locations corresponding to the first input from the user in the first record and the second record, respectively. Therefore, there is at least one element that belong to both L_1 and L_2, since the actual first position of the PIN must be included in both sets.

Let E denote the set of positions that belong to both L_1 and L_2. Thus, $1 \leq |E| \leq A$. Obviously, when $|E| = 1$, the shoulder surfer can uniquely identify the first position of the PIN, i.e., the only element in set E. In a more general case, the best strategy of a shoulder surfer is to pick a position from set E, and the success rate of the guess is equal to $1/|E|$.

As a result, P_1 can be calculated as:

$$P_1 = \sum_{|E|=1}^{A} \frac{1}{|E|} \cdot P_{1,\,|E|} \tag{1}$$

where $P_{1,\,|E|}$ denotes the probability that there are $|E|$ elements in common between the two sets of possible positions of the first inputs in the two login processes.

Since the actual first position of the PIN must be in both L_1 and L_2, $P_{1,\,|E|}$ is equivalent to the possibility that the number of elements in common between two sets, each of which is formed by randomly picking $A-1$ cells from the remained A^2-1 cells, is $|E|$. Thus, we have:

$$P_{1,\,|E|} = \frac{\mathrm{C}_{A-1}^{|E|-1} \cdot \mathrm{C}_{A^2-1-(A-1)}^{A-1-(|E|-1)}}{\mathrm{C}_{A^2-1}^{A-1}} = \frac{\mathrm{C}_{A-1}^{|E|-1} \cdot \mathrm{C}_{A^2-A}^{A-|E|}}{\mathrm{C}_{A^2-1}^{A-1}} \tag{2}$$

By combining Equation (1) and Equation (2), we can calculate P_1 as follows:

$$P_1 = \sum_{|E|=1}^{A} \frac{1}{|E|} \cdot \frac{\mathrm{C}_{A-1}^{|E|-1} \cdot \mathrm{C}_{A^2-A}^{A-|E|}}{\mathrm{C}_{A^2-1}^{A-1}} \tag{3}$$

Let P_2, P_3, and P_4 denote the probability that the shoulder surfer correctly guesses the second, third and fourth digits of the PIN after two records, respectively, given that she can obtain two records of the whole login process. Let P_{PIN} denote the probability that the shoulder surfer correctly guesses the whole PIN given two records. Thus, we have:

$$P_{PIN} = P_1 \cdot P_2 \cdot P_3 \cdot P_4 \tag{4}$$

Since the table is reset before each input from the user, if the 4 positions of the PIN are totally independent from each other, P_i's ($i \in \{1,\ 2,\ 3,\ 4\}$) have the same value. Thus, we have:

$$P_{PIN} = P_1^4 \tag{5}$$

According to Equation (5), the success rate of a shoulder surfer correctly guesses the whole PIN ranges from 25% to 20%, when the size of the table (i.e., A) is set from 5 to 10. Since the maximum number of retries that the adversary can make without being detected is 2, our scheme can provide sufficient protection given that the shoulder surfer can acquire two login records.

3.5 Security Analysis of the Rotary PIN Entry Scheme

In this section, we analyze the security of the two variants of this scheme under three types of adversary models, i.e., zero-knowledge (ZK) adversary, recording-based shoulder surfing (RSS) and cognitive shoulder surfing (CSS). Let P_{Ad-r}^{Var} denote the probability that an adversary successfully gets the secret PIN within r attempts in a specific variant of this scheme under one of the three adversary models, where Ad and Var denote an adversary model and a variant of the scheme, respectively. In the following, we first analyze each adversary model when $r = 1$, and then discuss the case when $r = M$, where M is the maximum number of successive PIN entry failures allowed, before the verifier retains the card and suspends the usage of the relevant account.

3.5.1 Zero-knowledge Adversary

Under the zero-knowledge adversary model, the adversary knows nothing about the secret PIN, and thus can only launch a random guessing attack.

In $VO(N)$, whatever choice the adversary makes, $N \times A$ distinct inputs are submitted for the verification. Thus, we have:

$$P_{ZK-1}^{VO(N)} \approx \frac{N \times A}{A^4} = \frac{N}{A^3} \tag{6}$$

Note that, as shown in Equation (6), $\frac{N}{A^3}$ is an approximation of $P_{ZK}^{VO(N)}$. This is due to the

fact that, except for the case $N = 1$ where the above expression corresponds to the exact value of $P_{ZK}^{VO(1)}$, for $N > 1$, there may exist repeated values among all the $N \times A$ inputs submitted. Since in our design N is a small number (e.g., 2 or 3), such an approximation is acceptable.

Similarly, for $VT(N)$ where the two rounds of the PIN entry process are executed independently from each other, we have

$$\begin{aligned} P_{ZK-1}^{VT(N)} &= P_1^{VT(N)} \cdot P_2^{VT(N)} \\ &\approx \frac{N \times A}{A^2} \cdot \frac{N \times A}{A^2} = \frac{N^2}{A^2} \end{aligned} \tag{7}$$

where $P_1^{VT(N)}$ and $P_2^{VT(N)}$ denote the probability that in $VT(N)$ an adversary inputs the secret X_1X_2 and X_3X_4 in the first and second rounds, respectively.

3.5.2 Recording-based Shoulder Surfing

In the following analysis, under the recording-based shoulder surfing model, we assume that the adversary acquires one record of the whole PIN entry process. All variants of this scheme fail, if the adversary can obtain more than one record of the whole PIN entry process.

In $VO(1)$, an adversary can easily deduce that the secret PIN is among the A distinct aligned inputs. Apparently, $P_{RSS-1}^{VO(1)}$ is $1/A$. Similarly, for $N > 1$, $N \times A$ aligned inputs are checked in $VO(N)$, although it is possible that some of these inputs are repeated. Since in our design N is a small number (e.g., 2 or 3), $P_{RSS-1}^{VO(N)}$ is approximately $1/(N \times A)$.

Similarly, in each round of $VT(N)$ there are $N \times A$ inputs subjected to the verification. Therefore, the probability that an adversary successfully guesses the correct digits can be computed in the same way as the calculation of $P_{RSS-1}^{VO(N)}$ in $VO(N)$. Since there are two rounds in $VT(N)$, we have:

$$P_{RSS-1}^{VT(N)} = (P_{RSS-1}^{VO(N)})^2 \approx (\frac{1}{N \times A})^2 = \frac{1}{N^2 \times A^2} \quad (8)$$

3.5.3 Cognitive Shoulder Surfing

Let L denote the maximum number of characters that a human being can memorize during cognitive shoulder surfing. In $VO(N)$, the best strategy for an adversary is to memorize $\lfloor L/4 \rfloor$ aligned inputs. According to previous research results [38, 60] about the cognitive capabilities of human beings, we know that $\lfloor L/4 \rfloor \geq 1$. Since each input has the same possibility of being the secret PIN and in each login attempt the adversary can test only one of inputs, we thus have $P_{CSS-1}^{VO(N)} = P_{RSS-1}^{VO(N)}$.

In $VT(N)$, an adversary has to make a decision about the distribution of her cognitive capability. Let L_1 and L_2 denote the number of characters that the adversary plan to memorize in the first round and the second round, respectively. Thus, we have $P_{CSS-1}^{VT(N)} = P_{RSS-1}^{VT(N)}$, as long as the following conditions is satisfied: $\lfloor L_i/2 \rfloor \geq 1$, where $i = 1,\ 2$.

3.5.4 Multiple Attempts by the Adversary

Let M denote the maximum number of successive PIN entry failures allowed, before the verifier retains the card and terminates the usage of the relevant account.

In the zero-knowledge adversary model, each random chosen input has the same probability of being the secret PIN. Thus, we have $P_{ZK-M}^{Var} = M \cdot P_{ZK-1}^{Var}$. Similarly, in the recording-based shoulder surfing model, each recorded input has the same probability of being the secret PIN. Thus, we have $P_{RSS-M}^{Var} = M \cdot P_{RSS-1}^{Var}$.

As to the cognitive shoulder surfing model, in $VO(1)$ and $VO(N)$, if $\lfloor L/4 \rfloor \geq M$, we have $P_{CSS-M}^{Var} = M \cdot P_{CSS-1}^{Var}$. Otherwise, we have:

$$P_{CSS-M}^{Var} = \lfloor \frac{L}{4} \rfloor \cdot P_{CSS-1}^{Var} + (M - \lfloor \frac{L}{4} \rfloor) \cdot P_{ZK-1}^{Var} \quad (9)$$

Similarly, in $VT(N)$, if $\lfloor L_i/2 \rfloor \geq M$ for $i = 1,\ 2$, we have $P^{Var}_{CSS-M} = M \cdot P^{Var}_{CSS-1}$. Otherwise, we have:

$$P^{Var}_{CSS-M} = A \cdot P^{Var}_{CSS-1} + (M - A) \cdot P^{Var}_{ZK-1} \tag{10}$$

where $A = min\{\lfloor L_1/2 \rfloor, \lfloor L_2/2 \rfloor\}$. Based on Equation (10), the best strategy of an adversary in the cognitive shoulder surfing model and $VT(N)$ is to equally spread her cognitive capability to the two rounds.

3.5.5 Summary of Security Analysis of the Rotary PIN Entry Scheme

In Table 1, we summarize the security levels of the two variants proposed in terms of the probability that an adversary successfully guesses the secret PIN in one attempt, i.e., P^{Var}_{Ad-1}.

Adversary Type	VO(N)	VT(N)
Zero-knowledge	$\frac{N}{A^3}$	$\frac{N^2}{A^2}$
Recording-based shoulder surfing	$\frac{1}{N \times A}$	$\frac{1}{N^2 \times A^2}$
Cognitive shoulder surfing	$\frac{1}{N \times A}$	$\frac{1}{N^2 \times A^2}$

Table 1: P^{Var}_{Ad-1} for $VO(N)$ and $VT(N)$

From Table 1, we observe that there is a trade-off between the security level against zero-knowledge adversary and that against recording-based/cognitive-based shoulder surfing. Let N_{max} denote the maximum value allowed for N, taking into account the usability concerns. Then, on one hand, $VO(1)$ and $VT(N_{max})$ are the most secure variants against zero-knowledge adversary and recording-based/cognitive-based shoulder surfing, respectively. On the other hand, $VO(1)$ and $VT(N_{max})$ are the least secure variants against recording-based/cognitive-based shoulder surfing and zero-knowledge adversary, respectively.

In practice, in the design of a secure system, one needs to consider all the possible attacks and then find an appropriate balance between different security requirements. For example, assume that an ATM system is designed to allow only one person to enter the transaction area at a time. Thus, only zero-knowledge adversary and recording-based shoulder surfing are concerned. Assume that the security requirements against zero-knowledge adversary and recording-based shoulder surfing are P_{ZK}^{max} and P_{RSS}^{max}. In addition, assume that there exist estimated probabilities that the conditions for launching a given type of attacks are satisfied according to previous statistical data, e.g., reported events of card loss or installation of hidden cameras on ATM machines. Let denote such probabilities for zero-knowledge adversary and recording-based shoulder surfing as P_g and P_r, respectively. During the design, we need to minimize the value of $P = P_{ZK}^{Var} \cdot P_g + P_{RSS}^{Var} \cdot P_r$, while at the same time ensure that $P_{ZK}^{Var} \leq P_{ZK}^{max}$ and $P_{RSS}^{Var} \leq P_{RSS}^{max}$.

Apparently, the same trade-off exists when we consider the scenarios where multiple attempts are possible and the idea of balancing security conditions based on application-dependent requirements and risks can be readily extended to such scenarios.

3.6 Usability of the Position-based PIN Entry Scheme

Like many other security applications, there is always a trade-off between security and usability. Meanwhile, usability is one of the most important factors that affect the acceptance of a new security solution in reality.

Usability in this scheme is mainly relevant to two issues: (i) the difficulty of either searching/identifying a specific position in the table; (ii) the difficulty of memorizing all the positions of a PIN.

As indicated in section 3.4, the security of our design is not very sensitive to the size of the table. Thus, a natural choice is to select a small table size, and it is helpful in addressing both issues relevant to usability. In the following examples of design patterns, we set the

table size to 6.

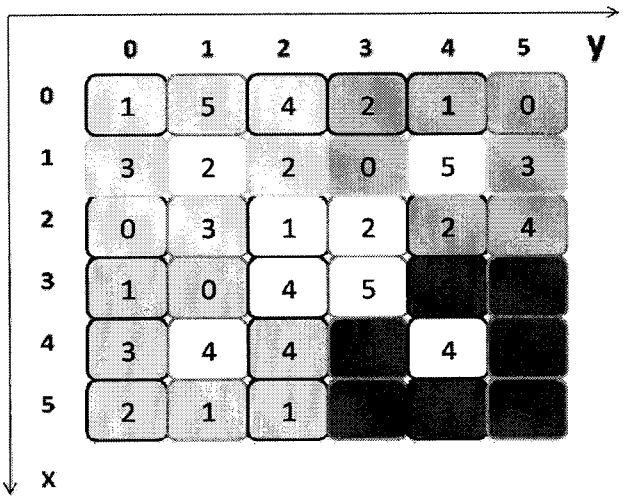


Figure 4: An example of colored patterns in displayed tables

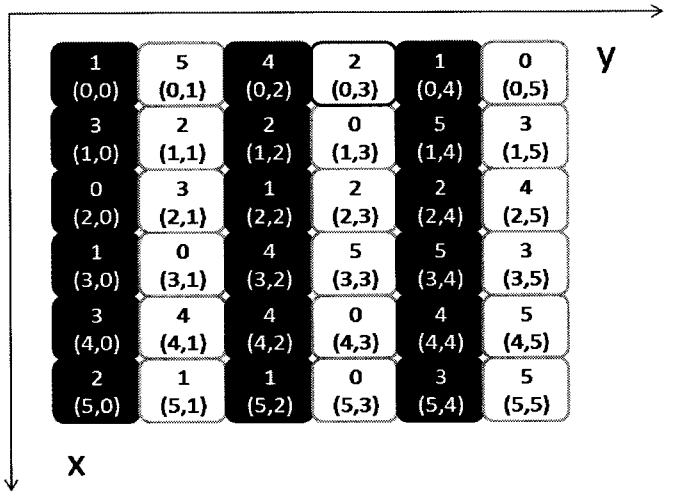


Figure 5: Another example of patterns in displayed tables

Another method that may improve the usability of our scheme is to use a colorful display. The basic idea is to use the contrast between different colors to enhance the identification and memorization of the positions in a table. Thus, the following rule is proposed to guide the design of the login interface: for any cell, at most two of its *directly-connected* cells have the same color as itself. *directly-connected* cells refers to cells that share one

and only one edge with each other. According to this rule, we can generate many different displayed patterns. Two examples are shown in Figure 4 and Figure 5. Furthermore, in Figure 5, besides following the above rule for coloring the display, we also include the pair of coordinates as part of information displayed on each cell so as to easily identify a specific position. These coordinates might be of more help to users who find it easier to remember numbers compared to remembering a set of locations or patterns.

Some users, e.g., color-blinded people, may have difficulty in color recognition but are more sensitive to geometric shapes or paths. Thus, they may choose certain geometric shape (or path) as the PIN, where each input/position is a vertex (or an endpoint of a subpath). Such a way of choosing a PIN has certain impacts on security, because the positions of a PIN are distinct and thus the choices of positions are not totally independent. However, we can still use P_1^4 as an approximate estimation of P_{PIN}. For example, our simulation results show that there is only around 1% difference between P_1^4 and the actual rate of guessing the PIN, when the table size is 6.

3.7 Usability Evaluation of the Rotary PIN Entry Scheme

In order to test the usability of this rotary scheme, different variants were evaluated by a group of twenty (7 female and 13 male) graduate students with engineering or computer science background. The average age of participants is about 27 years old (StdDev=4.285).

At the beginning of the test session, all participants were asked to provide their opinions, on a five point scale, for the following two statements: (Q1) There is a need to improve the security of current PIN entry method (Q2) I am willing to spend more time to enter my ATM PIN if the new scheme provides more security against shoulder surfing.

After answering the above two questions, the participants were given a short introduction to the scheme in order to help them understand its design objectives. During this introduction session, participants were also explicitly warned not to point to their chosen

PIN digits, e.g., by using their fingers, when they are trying to align them because this action may help reveal their PINs to shoulder surfers. Then each participant was asked to choose a PIN number that she is familiar with and register it through our evaluation software. Each subject was allowed to try the system for five times before the beginning of the actual testing session. Both the short introduction above and the trial session, combined, required about 5 minutes to complete.

During the test session, each user was asked to enter their PIN for 10 times using different variants. The system was designed to record the PIN entry time, and count the number of incorrect logins. Finally, after the experiment, participants were asked to (Q3) express their satisfaction about the new system, on a five point scale. Additionally, in order to have a more insight into the security of our scheme, participants were asked, (Q4), if they had a specific preferences in choosing the direction of the sector in which they align their PINs.

Unlike some other graphical PIN entry scheme where complex PIN entry procedure or the images chosen by the users might be forgotten, no follow up sessions were needed in our evaluation process since our system require users to only remember their PINs which is the same requirement for current PIN entry systems.

The participants' response for the first two questions are shown in Figures 3.6(a) and 3.6(b) (SA: strongly agree, A: agree, N: neutral, D: disagree, and SD: strongly disagree). Figures 3.6(c) and 3.6(d) show the users response to the statement "I am totally satisfied with this new PIN entry scheme" for $VO(1)$ and $VT(1)$.

Table 2 shows both mean PIN entry time and standard deviations for different variants of this rotary scheme. It is interesting to note that $VT(1)$, despite requiring two rounds to enter the PIN, has slightly shorter mean entry time compared to $VO(1)$. This can be explained by the fact that many participants felt more comfortable dealing with user interfaces having less number of digits. It should be noted that, throughout our experiments we used the *mouse* to rotate the wheels during the PIN entry process. In practical application

where touch screens or push bottoms are available, a considerable decrease in the mean entry time should be expected.

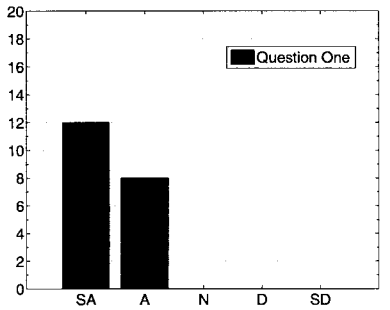
	Mean(seconds)	StdDev(seconds)
VO(1)	11.70	1.551
VO(2)	12.74	2.553
VO(3)	16.32	3.802
VT(1)	10.57	2.983

Table 2: Mean and standard deviations of PIN entry time

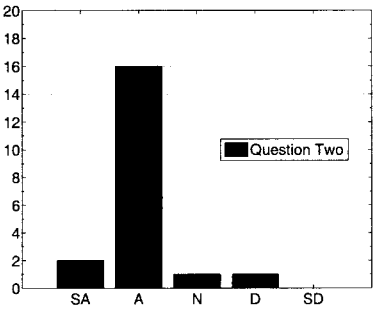
None of the participants made any mistake while entering their PINs after the practice session. Based on the feedback from participants, they felt that the system was easy to use and does not require them to remember any new information besides their regular PINs.

As for question 4, 80% of participants said that they did not give any preference to the position of the sector in which they align their PINs. Usually, they just tried to find the position of one digit of their PIN, then aligned the remaining digits in that position. However, 20% of participants had some preferred positions into which they aligned their PINs. For example, one participant always tried to align her PIN towards the 12 o'clock direction, while another participant preferred the 3 o'clock direction. Another two participants gave preference to a random (but fixed) area of the wheel.

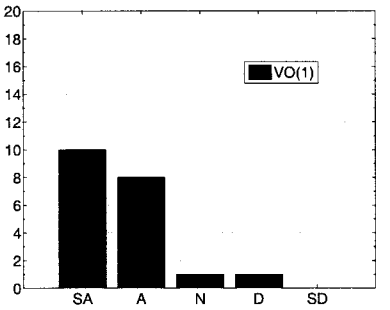
It should also be noted that despite the fact that the average login time using $VT(1)$ was less than the corresponding time for $VO(1)$, a larger number of participants were strongly satisfied with VO because it requires only one round for successful authentication.



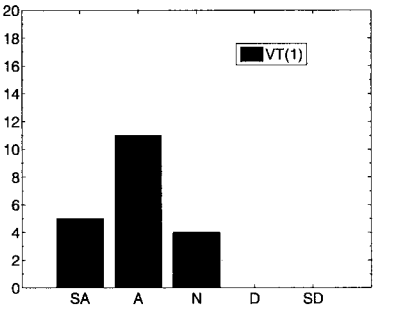
(a) Distribution of Question 1



(b) Distribution of Question 2



(c) Distribution of Question 3 for $VO(1)$



(d) Distribution of Question 3 for $VT(1)$

Figure 6: Distribution of answers to questions 1-3

Chapter 4

Click-based Graphical Authentication Scheme

4.1 Introduction

Improving the usability of security features in end user applications has been the focus of many recent works. No matter how secure the system is, if it is too difficult to use, users will simply drop it or try to circumvent security features. Encouraged by several psychological studies which reveal that people are better at recognizing and recalling visual information, as compared to verbal or textual information [40,51,71], graphical authentication schemes (also called graphical passwords schemes) have been proposed as an alternative to text-based passwords.

In this work, we focus on a subcategory of graphical passwords, namely, click-based graphical passwords. While existing click-based graphical passwords (e.g., [9,42,62,63]) achieve a high level of usability, many [9,62] of these schemes still suffer from several security concerns such as the hotspots issue and shoulder surfing attacks. In this chapter, we propose a click-based graphical authentication scheme inspired by Cued Click Points

(CCP) [9]. The proposed scheme aims at maintaining the merits of the CCP style click-based authentication solutions while improving the resistance to shoulder surfing attacks. Different strategies are adopted in order to address the shoulder surfing issues. Moreover, our security analysis provides a quantitative guidance on how to select the system parameters based on the specific security requirements. Furthermore, a preliminary usability study is also conducted on the proposed solution.

The rest of this chapter is organized as follows. Related work is reviewed in section 4.2. The details of our proposed solution are presented in section 4.3, followed by security analysis in section 4.4. Prototype implementation and usability evaluation are presented in section 4.5.

4.2 Related Work

In this section, we review some of the recent works related to click-based password schemes. The reader is referred to [5, 57] for further details about other graphical authentication schemes.

Wiedenbeck *et al.* [62, 63] proposed a click-based graphical authentication system, PassPoints, in which the user's password consists of a sequence of five click-points on a given image. In the password initialization phase, a user is allowed to pick any five pixels in an image as a password. During the login session, the image is displayed and the user has to click on the sequence of password click-points in the correct order. Since it is hard to require the user to click on the exact pixel, each click within a system-defined tolerance region of the original click-point is acceptable. The security and usability of PassPoints was evaluated by the authors [62, 63] as well as other researchers [7, 19, 59]. It was reported that, although PassPoints is user friendly, it has few security concerns. The primary security concern is the hotspots problem. This problem refers to the situation where some portions of image are more likely to be chosen by different users. This presents a serious

threat to the system since it narrows down the scope of guessing attacks. Another security concern is that this PassPoints scheme is not suitable for deployment in circumstances where shoulder surfing attacks are possible; an attacker will be able to break into the user's account if she records the login process for one time.

Chiasson *et al.* [9] presented another click-based graphical password scheme named Cued Click Points (CCP). CCP can be considered as a combination of PassPoints [62, 63], Passfaces [42] and Story [11]. In CCP, a password is one click-point per image for a sequence of five images. The displayed image is based on the previous click-point which provides an immediate implicit feedback to the user whether she is on the correct path of login. The experimental results provided by the authors show that this scheme is good in terms of speed, accuracy and number of errors. They also argued that CCP achieves a better security level compared to PassPoints [62, 63]. On the other hand, the CCP scheme still suffers from both the hotspots problem and vulnerability to shoulder surfing attacks.

In a subsequent paper [8], Chiasson *et al.* introduced a "persuasion" technique to their previous CCP scheme in order to reduce the hotspots problem. Their new proposal, named Persuasive Cued Click-Points (PCCP), utilizes persuasion to encourage users to choose more random, and hence more secure click-points. Based on their analysis, PCCP significantly reduces hotspots without significantly sacrificing the usability of the scheme. However, PCCP scheme is still vulnerable to shoulder surfing attacks.

4.3 Proposed Scheme

In this section, we present the system and adversary models of our solution. Then, we describe our proposed graphical authentication scheme.

4.3.1 System and Adversary Models

The proposed solution can be deployed in systems where the user has a client device which is able to display the graphical interface during the login process. We assume that the verifier is trusted to perform the authentication process correctly. During the authentication process, the verifier keeps a record of the number of successive failed logins. After a pre-specified number of failed attempts, the verifier may terminate the usage of the corresponding accounts until they are reset through a secure channel. In addition, for convenience, this counter is automatically reset upon a successful login.

In case of remote server logins, we further assume that all images used by the system are stored on the server side and all communication between the client and server are protected (e.g., through Secure Sockets Layer (SSL) or Transport Layer Security (TSL)). In this way, the selected click-points and their corresponding images are protected against simple networking sniffing attacks.

4.3.2 Proposed Scheme

As mentioned above, our design is inspired by both Cued Click Points [9] and PassPoints [62, 63]. Let I denote the set of images in the system database. Let $U_u \subset I$ denote the set of password images corresponding to user u. In this scheme, the password of user, u, corresponds to one click-point per image for each image in U_u. In the initialization session, user u is required to choose her password, i.e., choose one click-point per image for each image in U_u. Let N and P denote the size of U_u and I, respectively. A successful login response consists of one click-point per image for a sequence of S distinct images from U_u. In other words, after the initialization session, when a user, u, wants to login to the system, after entering the username, an image from the password set U_u will be randomly chosen and displayed. If the user clicks on the correct point (password click-point) in the picture, another image, chosen randomly from U_u, is displayed. Otherwise, an image from $I - U_u$

is displayed. As mentioned above, this gives an implicit feedback to the user whenever she makes a mistake. When the user clicks on the right point for a sequence of S images, the login process will succeed. If not, an error message will be displayed, at the end of the login process, to indicate a login failure.

4.4 Security Analysis

4.4.1 Random Guessing

To simplify our analysis, we assume that all the images used by our system have the same dimensions and we used tolerance squares of 19×19 pixels (same values adopted by PassPoints [62, 63], CCP [9] and PCCP [8]). Following the same analysis in [9], each picture would have about 400 valid tolerance squares.

Let P_{img} denote the probability of clicking within the right tolerance square based on random guess. Then we have

$$P_{img} \approx \frac{1}{400}. \tag{11}$$

Note that the above estimate does not take into account the hotspots problem. Careful analysis of the used images may allow the attackers to improve the chance of clicking within the right tolerance square.

4.4.2 Cognitive Shoulder Surfing

Based on the research results related to people's short term memory, which are mentioned in section 3.4, chapter 3, it is supposed to be difficult for a typical attacker to remember these S click-points and their corresponding images by simply observing the login process. Moreover, in each login session, these S images are selected from a larger set of images corresponding to the password images set of size $N > S$. Thus, this sequence of S images

has a relatively high probability to be different in each login session.

4.4.3 Recording-based Shoulder Surfing

Compared to cognitive shoulder surfing, having access to a recording device gives more power to the adversary. Some graphical authentication schemes designed to resist cognitive shoulder surfing, especially click-based ones, are vulnerable to recording-based shoulder surfing. As will be shown by our analysis, our scheme can be deployed in scenarios where the login process might be incidentally exposed to a recording-based shoulder surfer for up to two times.

One Login Record

As described above, in order to successfully login to a server using the proposed scheme, the user u is supposed to click on the right click-point for each image in a sequence of S images which are included in the password images set U_u. If the shoulder surfer is able to record the whole login process one time, she is able to obtain all the correct S click-points during this recorded session. All of these S images and their corresponding click-points can be seen as a knowledge set K_{OO} where $K_{OO} = \{img_1, img_2, \cdots, img_s\}$. Based on the system design, these S images are parts of the a user's password images set, U_u, which consists of N images and their corresponding click-points. In order to reduce the probability that the attacker, who tries to impersonate the legitimate user, is challenged by the same S images which are recorded during a previous login session, the size of password images set N is chosen such that $N \geq 2S$. In the case where $S \leq N < 2S$, at least $2S - N$ images from the knowledge set K_{OO} will show up for the attacker. If we ignore the influence of hotspots, for each challenge which is not in the knowledge set K_{OO}, the attacker has to make a guess. As discussed in section 4.4.1, the probability of making the right guess on each unknown image is P_{img}. Thus, the probability of breaking into the

system, which is denoted by P_{OO}, is given by:

$$P_{OO} = \sum_{i=i_0}^{S} \frac{C_S^i \cdot C_{N-S}^{S-i}}{C_N^S} \cdot (P_{img})^{S-i} \tag{12}$$

where i_0 is determined based on the size of the password images set U_u as follows:

$$i_0 = \begin{cases} 0, & N \geq 2S \\ 2S - N, & S \leq N < 2S \end{cases}$$

Figure 7 shows, for $S = 5$, how the probability of successful login by an attacker who records one login session, is reduced by increasing N.

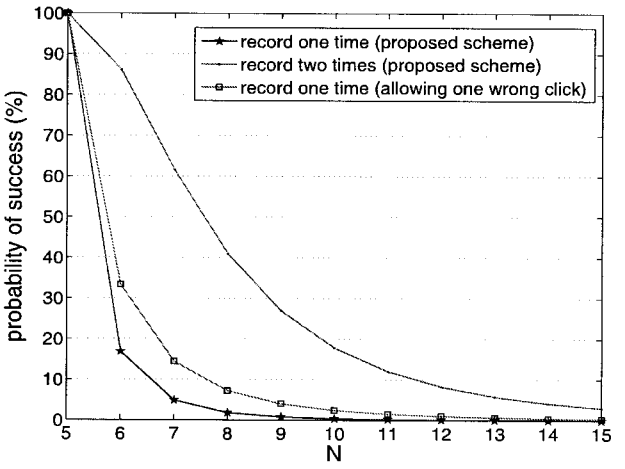


Figure 7: Probability of successful attack (S=5)

Two Login Records

In this section, we assume that the attacker is able to record two login sessions. Let $K_{OT} = \{img_1, img_2, \cdots, img_K\}$ denote the knowledge set of size $S \leq K \leq 2S$ gained by the attacker. Similar to the analysis of the one login record, the attacker is assumed to refer to the knowledge set K_{OT} whenever possible and make a random guess when the challenge

images are not included in the knowledge set. Let P_{OT} denote the probability of breaking into the system. Then we have Figure 7 shows, for $S = 5$, how the probability of successful login by an attacker who records two login sessions, is reduced by increasing N.

$$P_{OT} = \sum_{i=i_0}^{S} \sum_{j=j_0}^{S} \frac{C_S^i \cdot C_{N-S}^{S-i} \cdot C_{2S-i}^{j} \cdot C_{N-(2S-i)}^{S-j}}{C_N^S \cdot C_N^S} \cdot (P_{img})^{S-j} \tag{13}$$

where

$$i_0 = \begin{cases} 0, & N \geq 3S \\ 0, & 2S \leq N < 3S \\ 2S - N, & S \leq N < 2S \end{cases}$$

$$j_0 = \begin{cases} 0, & N \geq 3S \\ max(3S - N - i, 0), & 2S \leq N < 3S \\ 3S - N - i, & S \leq N < 2S \end{cases}$$

It should be noted that, while increasing the value of N clearly increases the system resistance to shoulder surfing attacks, the usable value of N is limited by the cognitive capabilities of the systems' users.

4.4.4 Increasing the Uncertainty of Observed User Input

Intuitively, one may think that providing some wrong information to the shoulder surfer during the login session may increase the security level. In this section we analyze the security of our scheme if the user is required to correctly click on exactly $S - 1$ out of S images during a login session. In other words, we insist that the user makes one wrong click during the login session. We also assume that the attacker is able to record the user clicks during one login session. In this case, unlike the original proposed scheme, if the shoulder surfer is able to record one successful login process, one of the recorded click-points will be incorrect.

Assume that the user decides to choose the wrong click-point when challenged by image img_{wrong}, $1 \leq wrong \leq S$. Let K_{set} denote the set obtained from the the knowledge set $K_{TO} = \{img_1, img_2, \cdots, img_S\}$ after excluding img_{wrong}. Again, in order to reduce the probability that the attacker, who tries to impersonate the legitimate user, is challenged by the same S images recorded during a previous session, we can set $N \geq 2S$. In the case where $S \leq N < 2S$, at least $2S - N$ images from the knowledge set K_{TO} will show up for the attacker.

Since the attacker can not identify img_{wrong}, when img_{wrong} shows up, the attacker just follows the observed user's click-point, which is not correct. If we ignore the influence of hotspots, for each challenge which is not in the knowledge set K_{TO}, the attacker has to make a guess. Let P_{img} denote the probability of making the right guess on each unknown image. Then, one can show that the probability of breaking into the system is given by:

$$P_{TO} = \sum_{i=i_0}^{S-1} \frac{C_{S-1}^{i} \cdot \left(C_1^1 \cdot C_{N-S}^{S-1-i} + C_1^0 \cdot C_{N-S}^{S-i}\right)}{C_N^S} \cdot (P_{img})^{S-1-i} \tag{14}$$

where

$$i_0 = \begin{cases} 0, & N \geq 2S \\ 2S - N, & S \leq N < 2S \end{cases}$$

From the results depicted in Figure 7, it is clear that, for the chosen system parameters, introducing this ambiguity in the user input reduces the security of the system because it also tolerates one wrong click from the attacker.

4.5 Prototype Implementation and Usability Evaluation

In order to be consistent with previous PassPoints [62, 63], CCP [9] and PCCP [8] studies, we set the dimensions of all the images used in the prototype system to 451×331 pixels and the tolerance squares to 19×19 pixels. The proposed scheme is implemented in JAVA and

the experiments are conducted using a Windows-based PC which has a screen resolution of 1440×900. A set of 200 images is compiled from personal collections and online websites which provide free-for-use images. Since secure storage of passwords is outside the scope of this work, for simplicity, our prototype system does not hash the passwords, however, this should be done in real systems. The exact pixel coordinates are stored as metrics to determine whether the users' click-points are tolerated or not. A snapshot of the prototype system is showed in Figure 8.

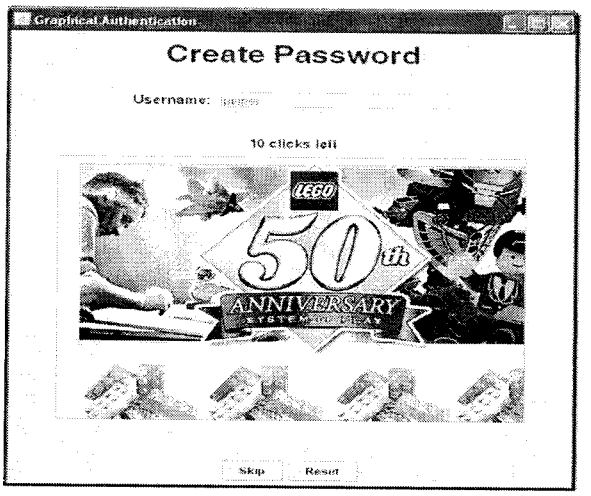


Figure 8: Snapshot of create password phase

In order to evaluate the usability of our proposed solution, the original proposed scheme with settings $S = 5$ and $N = 10$ has been evaluated by a group of 20 graduate students with engineering or computer science background (12 female and 8 male).

All participants completed an individual half an hour lab session. At the beginning, the participants were given a short introduction to our scheme in order to help them understand its design objectives. During this session, several life demos were shown to the participants. We also explained that the next image in the sequence depended on whether they clicked on the correct click-point on the current image. They were told that if they see an image

that they do not recognize as their password image during the Confirm Password or Login phases, then they should recognize that they have made a mistake. Participants had a chance to practice with the scheme, for 5 minutes, before starting to collect the statistics. Users were then asked to perform the following:

- Create Password: Create a password by clicking on one point for each 10 images displayed in sequence. In our implementation, users can skip an image if they find it difficult to remember the click-points in that image.

- Confirm Password: Confirm password by re-clicking it correctly. If users make mistakes during this step, they could reset the system to return to the Create Password step.

- Login: Log in with already chosen passwords.

Participants were allowed to use the reset button during the Create Password and Confirm Password phases when they saw an wrong image or whenever they felt they can not remember the current password.

Table 3 shows the success rate and completion time (average and standard deviation) for the Create Password, Confirm Password and Login phases. From Table 3, it is clear that, compared to the Login phase, a relatively longer time is required by the users in the Create Password phase. This can be explained by noting that users have to select images they feel easy to remember and choose click-points on them. The results in Table 3 also indicate that, on average, about 12 seconds are enough for a user to login to the system, which while longer than the time required to input a text based password, is still acceptable in practical scenarios.

The last 5 minutes of each session were devoted to filling up a questionnaire about the user's opinion on this graphical authentication scheme (see Table 4). Five-point Likert scales [69] were adopted, where 5 indicates strong agreement and 1 represents strong

	Create	Confirm	Login
Success Rate	102/103(99%)	93/102(91%)	86/92(93%)
Mean Time (StdDev)(in seconds)	47.3(22.5)	22.9(8.1)	12.2(4.2)
Median Time(in seconds)	41.5	20.6	11.1

Table 3: Success rate and completion time

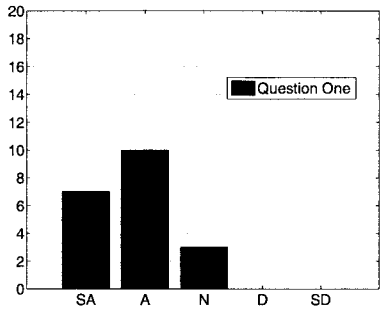
disagreement with the given statement (SA: strongly agree, A: agree, N: neutral, D: disagree, and SD: strongly disagree). Table 4 shows the mean and median scores for all the answers. The detailed distributions for each question are shown in Figures 4.9(a), 4.9(b), 4.9(c), 4.9(d), 4.10(a), 4.10(b), 4.10(c), 4.10(d), 4.10(e) and 4.10(f), respectively.

Questions	Mean	Median
1.I can easily create a graphical password	4.2	4
2.Logging on using a graphical password is easy	4.05	4
3.I can easily remember a graphical password	3.9	4
4.A stranger would be worse at guessing my graphical password than a friend	3.55	3
5.I prefer graphical passwords to text passwords	3.55	4
6.I think graphical passwords are more secure than text passwords	4.25	4
7.I think other people would choose different points than me for a graphical password	3.7	4
8.With practice, I can quickly enter my graphical password	4.5	5
9.I think I have preference to certain points when create graphical password	4.35	4
10.I think the features of images used affect my choice when I choose click-points	4.7	5

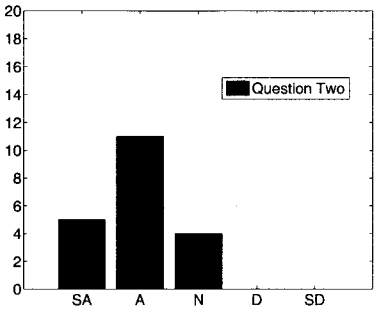
Table 4: Questionnaire results. Scores are out of 5.

4.5.1 Hotspots and User Choice

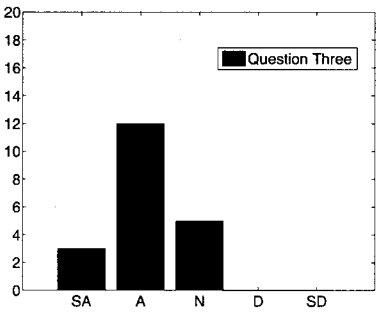
Based on the response for question 9 as well as post-test feedback, it is clear that participants have certain preference to certain click-points for creating their graphical passwords. Several participants tried to select symmetric points of the images while some tried to choose people's or animals' (e.g., cats, dogs and birds) noses or ears in the images. One participant claimed he had trouble to remember these images, so he always chose top left



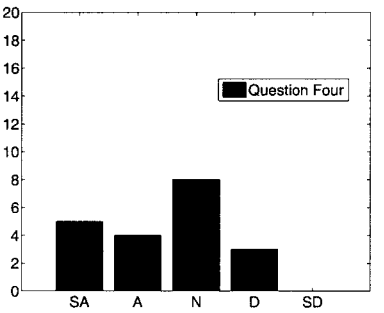
(a) Distribution of Question One



(b) Distribution of Question Two



(c) Distribution of Question Three



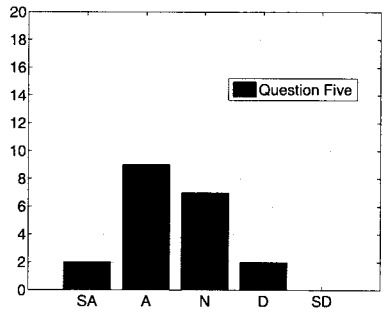
(d) Distribution of Question Four

Figure 9: Distribution of answers to questions 1-4

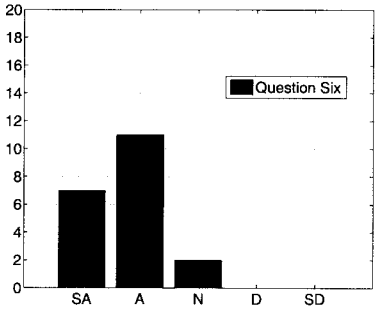
corner of each image as his graphical password, even though he understood that this behavior was unsecure. In general, small and "clickable" points (e.g., circles, squares or letters) were most likely to be selected by participants.

Some participants reported that some images were too difficult to be adopted as their password images. These images were either cluttered or empty. This was expected by us before the experiments and that was why we provided a skip button in our prototype implementation.

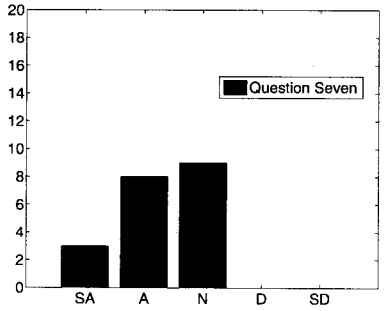
One participant pointed out that with this scheme it was very difficult for him to share password with his friends in case he had to do that. Indeed, his argument is true and it indicates an advantage of our scheme since it discourages the unsecure behavior of password sharing.



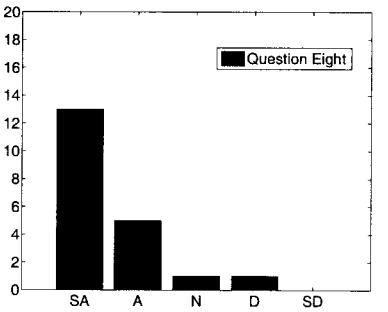
(a) Distribution of Question Five



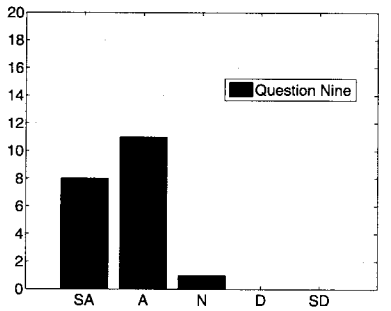
(b) Distribution of Question Six



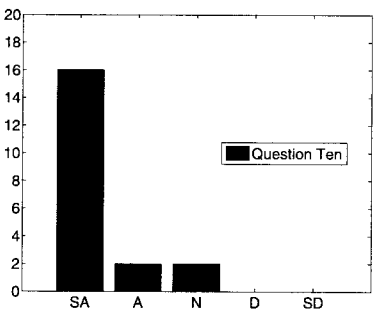
(c) Distribution of Question Seven



(d) Distribution of Question Eight



(e) Distribution of Question Nine



(f) Distribution of Question Ten

Figure 10: Distribution of answers to questions 5-10

Chapter 5

Strengthening Password Authentication Using Mobile Devices and Browser Extensions

5.1 Introduction

Text-based passwords are still the most common form of authentication adopted by online service providers such as banks, online stores, and social networking sites. Generally, a combination of a valid username/userid and a password is enough to obtain full access to a given account. Moreover, in many scenarios, the username/userid is not considered as sensitive information and can be easily leaked. Consequently, the protection of users' accounts relies heavily on the protection of users' passwords which makes passwords the hot target of identity theft criminals.

There are many techniques that are used by attackers to obtain and exploit passwords. Among these techniques, phishing has been one of the most popular and effective approaches. It is widely adopted by attackers all over the world, partially because it is easy to

launch: a forged website and/or a fake e-mail is usually enough to trick novice users and acquire their passwords and/or other sensitive information.

In addition to phishing, keylogging programs are also widely used by adversaries in order to collect sensitive information. Sometimes, people have to enter their sensitive information using untrusted machines (e.g., using public PCs in Internet cafes), which makes them particularly vulnerable to keyloggers. While computers in public places have a higher probability of being infected by keylogging programs, home computers can also be infected by such programs due to the flooding of spywares and botnets.

In this chapter, we propose a framework aiming at improving the security of online password authentication against phishing, keylogging and shoulder surfing. We also present a prototype implementation for our design. Our solution is inspired by the idea presented in [47]: instead of sending the user's password, *pwd*, in plaintext to a remote server, we first extract the domain name of the remote server *dom* and then calculate the hash value *hash(pwd, dom)*, which is sent to the remote server as the password. Here, domain name *dom* works as a salt in this hash function (in cryptography, the term salt is used to denote a string of random bits that are used as one of the inputs to a key derivation function.) This design does not require any change to the server side. Moreover, since domain names are unique, the hash values of passwords for any two domain *hash(pwd,* dom_1*)* and *hash(pwd,* dom_2*)* are very unlikely to collide. As a result, passwords acquired by phishing sites are not useful for login at any other domain. Furthermore, in order to protect against keyloggers and shoulder surfing, users do not type in their passwords. Instead, these passwords are securely stored on the users' mobile devices (such as PDAs and cell phones) and are sent directly to the PCs through the Bluetooth link. The rest of this chapter is organized as follows. The related work is reviewed in section 5.2. Details of our proposed solution and prototype implementation are presented in section 5.3. Finally, the security of the proposed framework is analyzed in section 5.4.

5.2 Related Work

Several recent countermeasures to defend against phishing attacks have been proposed from both academia and industry. A popular method is to develop/find distinguishing features of phishing sites and warn the user if the features extracted from the accessed site matches these previously determined features. This approach has been implemented in several browser toolbars such as TrustBar [22], SpoofGuard [10], Netcraft Toolbar [39], eBay-Toolbar [13] and SpoofStick [55]. These heuristics toolbars detect malicious URLs, and some of them provide more information about the domain they are visiting, such as the true domain name, location and country name, the creation date of the domain. Users are warned when these toolbars consider or judge the visited site as fraud. Because of the inherent imprecision of these heuristic methods, some users are bothered since they are required to make the final decision whether they should trust the toolbar advice or not. Moreover, many factors considered by these toolbars are transparent which opens a window for attackers to adapt to them in order to bypass the underlying heuristics. Additionally, 13-54% of users, who are warned by anti-phishing toolbars, ignore the warning and continue their browsing [70].

Kirda and Kruegel [30] presented a Firefox extension, AntiPhish, that aims to protect users against spoofed website-based phishing attacks. This solution is inspired by automated form-filler applications which usually have a master password to protect sensitive information. Whenever the user enters information into text field elements of type *text*, *password* and *textarea*, AntiPhish checks the list of previously captured values. If each value is identical to the one just provided by the user, the domain is checked with the previously stored one. In other words, this AntiPhish tracks the sensitive information of a user and generates warnings whenever the user attempts to submit sensitive information to a website which is considered to be untrusted by the application.

Parno *et al.* [41] proposed a mechanism, Phoolproof, that leverages a trusted device

to perform mutual authentication between client and server. This solution eliminates reliance on perfect user behavior, thwarts Man-in-the-middle (MITM) attacks and protects users' account information against keyloggers and most forms of spywares. Phoolproof is a two-factor authentication system in which the trusted device works as an additional authenticator. Therefore, the attacker must compromise the trusted device and the user's credentials in order to impersonate the user. A long-term secret is required to be stored on the mobile device, and most cryptographic operations are conducted on the trusted mobile device. As mentioned by the authors, this design is vulnerable to session hijacking attacks and needs modification on the server side.

Mannan *et al.* [36] presented a simple approach, using a mobile device, to strengthen the authentication process from an untrusted computer. In their design, the user's long-term secret is cryptographically separated from the untrusted computer which means that a client PC can only access temporary secrets even though most computations are performed in this untrusted PC. The user's long-term secret (typically short and of a low-entropy) is input through a trusted personal device such as a mobile phone or a PDA. The long-term secret is never stored in this personal device. The trusted personal device only provides a user's long-term secret to the client PC after encrypting the secret with a pre-installed and valid public key of a remote server. This proposed solution is intended to protect sensitive information from various attacks such as keylogging, phishing and pharming attacks, as well as to provide transaction security to foil session hijacking. However, in order to achieve these goals, a modification of the server side is mandatory.

5.3 Proposed Solution and Prototype Implementation

5.3.1 Main Idea

Nowadays, computers and mobile devices with Bluetooth functionality are very common. Moreover, in many practical scenarios, these mobile devices can be consider as trusted devices or at least far more trustworthy than PCs. Based on this observation, we argue that it is better to shift the sensitive login burden from untrusted computers to these relatively more trusted devices (e.g., cell phone).

As mentioned above, the password sent to the server is constructed by hashing the user supplied password, *pwd*, with a salt value obtained from the login site. The objective of using a salt in the password hashing process is to generate different passwords for different domains. Thus, the generated password will be resistant to spoofing websites because the domain of a spoofed site is very likely to be different from the genuine one. Different possible values can be considered as candidates for the salt such as the domain name of the site hosting the current page and the SSL certificate of the target domain. As mentioned in [47], different choices have their own advantages and disadvantages. However, based on the availability and security concerns, the current domain seems to be the most suitable choice to be used as a salt. For instance, using SSL certificate has several drawbacks such as being hard to replicate manually. It also suffers from compatibility problems when the target site has no SSL certificate.

Some popular websites have multiple domains for different countries or territories. For instance, eBay has *ebay.ca* for Canada and *ebay.cn* for China. So when a user creates an account using *ebay.ca* as a salt, it is not possible for the user to login to the site for China. On the other hand, if a user creates an account at *ebay.ca*, usually the user should seldom login to *ebay.cn*. Even if the user has to do so, she can just create a new account for the specific domain name in this situation. Yahoo and Wikipedia have a different way

to organize their domain names. For example, the domain name for Yahoo Canada is *ca.yahoo.com* while the domain for Yahoo China is *cn.yahoo.com*. In this scenario, we can just extract the two top-level domains and use them as the salt for all the domains that match "*.yahoo.com". Fortunately, most sites only have one consistent domain.

Another challenge that we faced is that some sites have special restrictions on the chosen passwords in order to force users to avoid choosing weak or easy to guess passwords. Some of these restrictions require that the password should contain at least one non-alphanumeric character and/or at least one uppercase character. In order to overcome this problem, the hash encoding algorithms, running on the mobile device should satisfy various restrictions imposed on the servers' passwords (e.g., by including at least one uppercase, one lower case, and one numeric character in its output.)

5.3.2 System Architecture

Our prototype implementation consists of three components: a J2ME program (MIDlet) installed on the mobile phone, a Java desktop application on the PC and a Mozilla Firefox extension.

The first component of the system is a MIDlet application running on the user's mobile phone. This program, which can run on any J2ME-enabled mobile phone, receives the domain name of the current login page, from the Java application running on the PC, through Bluetooth communication. Then it searches this domain name in a lookup table which contains all the websites that the user wants to secure using this framework. If a match is found, the user is required to input the corresponding username and password. Users can either choose to input their credentials manually into this MIDlet, or call the saved credentials in the mobile phone. The security implications of both options are further discussed in section 5.4.2. Before being sent to the Java application on the PC, the username remains

the same while the password is hashed using the following pseudorandom function (PRF):

$$PRF_{pwd}(dom) = hash(pwd, dom), \tag{15}$$

where the user's password *pwd* is used as the key while the website's domain *dom* is used as input to the PRF. In other words, as mentioned above, *dom* is used as the hash salt.

The second component of our solution is a Java application that has to be installed on the computer running the browser. It receives the URL of the current login page from the Firefox extension, and then forwards it to the MIDlet through the Bluetooth link. Once the user sends the credentials to this Java application, it forwards them to the Firefox extension.

The third component of our solution is a Firefox extension which extracts the URL of current login page, and sends it to the desktop Java application. It then waits for the response from the desktop Java application. After receiving the username and hashed password from the desktop Java application, this extension passes them to the web server. Our solution does not require any modifications to the server side.

Figure 11 illustrates the main components of the prototype and information flow among them. In what follows we provide a more detailed description for these components.

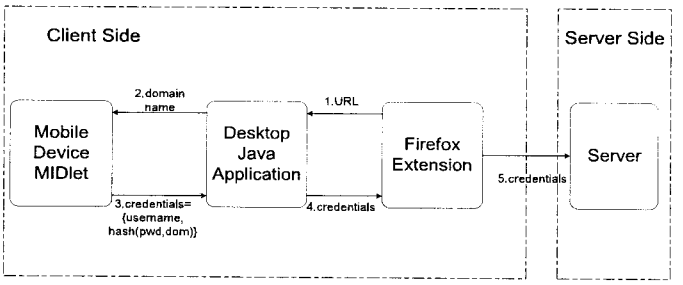


Figure 11: Architecture and information flow of the proposed solution

Firefox Extension

Similar to many add-ons of the Firefox Browser, our extension is developed in XML User Interface Language (XUL) and JavaScript. The developed extension works as a toolbar added to the Firefox Browser, which is activated by the user who presses a "start" button whenever the user wants to login to a remote server using our solution. The functionality of this extension is to extract the current URL of the login page and send it to the desktop Java application as an asynchronous JavaScript and XML (AJAX) request. It then waits for the reply from the desktop Java application. Once the response is received, this extension extracts the username, hash value of the password from the response string, and fills them in the login page. In the password field, the star symbols ($*$) appear instead of the symbols and characters of the typed password. When the extension tries to fill the username field automatically, it searches certain id values for the username from the source code of the login page. Most websites use common ids for the username field, such as *Email* (Google Gmail), *username* (Yahoo Mail) and *email* (Facebook). If this extension can not identify the username field from the source code of the login page, then the user is prompted to manually input the username into the username field. Throughout our experiments, we did not face this problem for the password field that has the statement $type = \text{“}password\text{”}$.

Desktop Java Application

The desktop Java application works as an information relay station between the Firefox extension and the MIDlet. We can separate this application into two parts: http server module and Bluetooth module. We were not able to use socket communications in order to establish a communication link between the Firefox extension and this Java application because JavaScript does not support sockets for security reasons. Instead, AJAX technique is chosen in the solution for this specific communication requirement. In order to adopt AJAX, one part of desktop Java application is developed as a simplified http server which

opens the http service at the localhost of the computer using port 18000 (the selection of port 18000 is arbitrary. The reader is referred to [23] for further details about port numbers.) This http server keeps checking for the AJAX request from the Firefox extension. After receiving the request, it parses it and sends the domain name to the Bluetooth module. Whenever this http module receives the credentials from the Bluetooth module, it forwards them to the Firefox extension as an AJAX response. The Bluetooth module is developed using bluecove 2.1.1 which is a Java API for Bluetooth. This module has two threads: a client thread and a server thread. The client thread (discovery thread) searches for new Bluetooth devices. Whenever a new device is found, a service search is performed to the found device. If the service is found successfully, a connection is established to this device. On the other hand, the server thread waits for new connections from other Bluetooth devices. After the connection is constructed with the MIDlet of the mobile phone, the communication channel is open. When this module receives the domain name of the login page from the http server module, it forwards it to the MIDlet. Then, it keeps checking for the response of the MIDlet. Whenever the Bluetooth module receives a response from the MIDlet, it passes it to the http server module.

MIDlet

A MIDlet is an application conforming to the Mobile Information Device Profile (MIDP) standard. In order to develop MIDlets for mobile phones, some infrastructure is needed. Sun provides a wireless toolkit for Connected Limited Device Configuration (CLDC), namely, Sun Java Wireless Toolkit 2.5.2 (WTK 2.5.2). Our MIDlet is divided into the following three modules: Bluetooth module, Hash module and Record Management module. Similar to the desktop Java application, this Bluetooth module also has two threads: a client thread and a server thread. When this module receives the domain name from the desktop Java application through the Bluetooth channel, it searches the domain name in

a lookup table which contains all the websites that the user wants to associate with this solution. If the domain name matches one item in the table, the user is prompted to input the corresponding username and password. Otherwise, an error message is displayed. Two options are then provided to the user. If the user decides to input the username and password manually, the record management module will not be activated. However, the user can also call the saved credentials using the record management module. The password is then hashed using SHA-256 which returns a 256 bit message digest. The generated result is Base64 encoded which yields a 28 character string. Then the string is shrunk to a pre-specified length (e.g., 10). The length can be adjusted depending on the server side requirements. Finally, the generated hashed password and the username are sent to the desktop Java application through the Bluetooth channel.

In order to use our system, the Firefox extension and the Java application are assumed to be installed in the computer running the user's browser. In many scenarios, users may not have the access privileges to install two modules on the used computers (e.g., computers in Internet cafe). To address this issue, a secondary option is provided by the MIDlet through a Graphical User Interface (GUI) to help users generate their hashed passwords where the user only needs to input the password and the domain name. Then, the MIDlet computes and displays the hashed password to the user who can then input it in the password field of the login page using the keyboard of the browser's computer.

5.4 Security Analysis of the Proposed Solution

In this section, we investigate some security threats associated with our solution.

5.4.1 Security of the Bluetooth Link

Since the hashed version of the password is sent from mobile phone to the PC through the Bluetooth channel, the security level of this part definitely affects the overall security of our solution. When the Bluetooth connection between mobile phone and computer is set up without activating the link encryption, it can be easily eavesdropped. Moreover, it is also easy for an attacker to replace the original payload data with other data. Thus, when our solution is deployed, link encryption between these two units must be activated.

In Bluetooth, encryption is perfomrmed using the stream cipher E_0 [66]. The core of E_0 is built on four independent linear feedback shift register (LESR) and a finite state machine as a combining circuitry which introduces sufficient nonlinearity. A detailed analysis of this algorithm is beyond the scope of this work. Instead, in here, we summarize some known attacks on E_0. Jakobsson and Wetzel [28] pointed out two attacks on the E_0, the first of which is of $O(2^{100})$ time complexity while the other one requires $O(2^{63})$ time effort using 2^{34} observed symbols. Fluhrer and Lucks [15] presented an attack on E_0 using observed keystream and the public knowledge of the encryption mechanism. This attack can recover the initial state of E_0 in $O(2^{68})$ using 2^{43} observed or known cleartext. However, since the required number of known symbols is very large compared to the symbols in a frame, the proposed attack can not directly reveal the link key. Krause [32] reported a more powerful attack on E_0 in terms of the required number of symbols. This attack has a time complexity of $O(2^{77})$ while only requires 128 known symbols. An improved correlation attack is presented by Golic [16] which achieves $O(2^{70})$ time complexity using less than one frame of known symbols. Courtois [17] also presented an attack which requires $O(2^{49})$ time complexity if $2^{23.4}$ bits are available to the attacker. While the above attacks on the E_0 cipher are of complexity less than $O(2^{128})$, which is the complexity of exhaustive search through the key space, the effort required by all these attacks is impractical and does not present any threat to our implementation.

In order to avoid the high time complexity of a direct attack on E_0, one may try to attack the Bluetooth link during the pairing procedure. Prior to Bluetooth 2.1 specification, Bluetooth technology was somewhat sensitive to passive and active attacks on the pairing procedure. However, in Bluetooth 2.1 + EDR, a new feature called Secure Simple Pairing (SSP) is introduced in order to increase the level of security. Now, the latest specification of Bluetooth is Bluetooth 4.0 which also supports the SSP.

In summary, if link encryption is activated between the mobile phone and the PC, the Bluetooth communication channel can be considered as a secure channel for our solution.

5.4.2 Storing Credentials on Mobile Phones

As mentioned before, users can choose to store their credentials in the mobile phone and call the saved credentials whenever needed. Otherwise, users may choose to input the credentials to the MIDlet manually. If users choose the former, it is easy for them to conduct the authentication process since they do not need to be bothered by typing in this information. However, saving credentials on the mobile phone may have some security implications, e.g., if the mobile phone is lost or stolen. A master password which protects unauthorized access to the mobile device may mitigate this risk. On the other hand, the manual credential entry option may not be convenient for some users, especially when they are not used to the cell phone small keypad. In conclusion, users can choose either one of these options depending on their specific needs and security requirements.

5.4.3 Phishing Attacks

Using the proposed solution, each site would receive a hash value of the password with domain name as a salt. Moreover, since the domain names are unique, the hash values of passwords for any two domain hash(pwd, dom_1) and hash(pwd, dom_2) are almost certain to be different. Consequently, when a phishing site receives the hash value of the user's

password , the attacker can not use this value to directly hijack the user's account. However, it is possible for the attacker to launch an offline dictionary attack. A straightforward method to mitigate this problem is to use the so called *slow hash function*, which is already widely implemented in the UNIX system to increase the computing resources needed to launch a dictionary attack [21,45].

5.4.4 Shoulder Surfing Attacks

According to our design, users can choose to store their credentials on the mobile phone. When they are required to input these credentials, they can simply retrieve them. In this case, it is impossible for shoulder surfers to obtain the users' passwords. On the other hand, some users may prefer not to store their credentials on the mobile phone and input their passwords manually, which opens a small window for shoulder surfers. However, the task of the shoulder surfer is much harder in this case because the keypad of the mobile phone is much smaller than the PC. Also, many mobile phones use one key to control three or more characters. Thus, it is reasonable to assume that the probability to mount a successful shoulder surfing attack is very slim in this scenario. Another possible chance for shoulder surfing is when users have to use the secondary option (see section 5.3.2) to generate the hashed passwords and enter them using the keyboard of the PC. In this case, the hashed passwords might be leaked to the shoulder surfer, especially when the shoulder surfer has some recording capabilities.

5.4.5 Keylogging Attacks

In the normal setup of our solution, the keyboard of the computer is not used to input any sensitive information. Instead, all the sensitive user's credentials are sent directly by the mobile device to the Firefox application. Even if the keylogging program has some features such as screen grabbing, it will still not be able to acquire any useful credentials

from the users. However, our solution might be vulnerable to sophisticated keyloggers that are able to monitor communications on certain ports or webform grabbers that can get web submissions. In order to defend against this threat, we can apply the one time password (OTP) technique to the proposed solution. In particular, we can replace $hash(pwd, dom)$ by $hash(pwd, dom, time)$.

Chapter 6

Conclusions and Future Work

Today, users' sensitive information is severely threatened by many prevalent attacks. Throughout this work, we focused on three threats raised by identity thieves: shoulder surfing threats to terminal applications (e.g., ATM, PoS and PDA), shoulder surfing of click-based graphical passwords and phishing attacks.

In order to address the shoulder surfing threats against terminal applications, we proposed two shoulder surfing resilient PIN entry schemes that achieve a good balance between security and usability. Our analysis shows that, our position-based scheme is resilient to shoulder surfing, given that the attacker has limited capability in recording the login process. Different variants of the proposed rotary scheme display a trade-off in resisting different types of adversaries, and thus the selection of the appropriate scheme can be optimized considering the types and frequencies of risks encountered in the application.

To defend against shoulder surfing attacks on click-based graphical passwords, we proposed a shoulder surfing resistant click-based graphical password scheme. Our security analysis indicates that the proposed solution offers a good level of security when the attacker's capability of recording the login procedure is limited, while still provides promising results in term of usability.

We have also proposed an approach to strengthen the authentication process aiming at

protecting users' sensitive credentials of the Internet services. The proposed solution provides improved security against some prevalent attacks (e.g., shoulder surfing and phishing attacks) to Internet authentication progress. An advantage of our proposed solution is that it can be directly deployed to some current services without the need to change the server side.

While the above proposed solutions present a good addition to the tools and methods that can be used to defend against identity theft, further research is required to address other practical issues that were not addressed in this work.

In particular, a thorough and long term usability study of the proposed PIN entry and click based graphical password schemes is required. This study should address the long term memory of users, an issue which was not addressed by our usability tests.

Quantifying and mitigating the threat raised by the hotspots issue in graphical passwords is another interesting, yet challenging, research direction.

While the proposed solution, in chapter 5, provides resistance against many classes of attacks, it does not address the Man-in-the-middle (MITM) threat. Modifying this scheme to defend against this class of attack is another promising research direction.

Bibliography

[1] APWG. Anti-phishing working group. http://www.antiphishing.org/.

[2] APWG. Phishing activity trends report 1^{st} half 2009. http://www.antiphishing.org/reports/apwg_report_h1_2009.pdf.

[3] ATMScam. Bank ATMs converted to steal bank customer ids. http://www.utexas.edu/police/alerts/atm_scam/.

[4] Larry Barrett. Identity theft cost victims $54B in 2009. http://www.esecurityplanet.com/trends/article.php/3864616/Identity-Theft-Cost-Victims-54B-in-2009.htm.

[5] Robert Biddle, Sonia Chiasson, and P. C. van Oorschot. Graphical passwords: Learning from the first generation. Technical report, Carleton University, 2009.

[6] Mark Brader. Shoulder-surfing automated. *Risks Digest*, 19, 1998.

[7] Sonia Chiasson, Robert Biddle, and P. C. van Oorschot. A second look at the usability of click-based graphical passwords. In *Proc. of 3rd Symposium on Usable Privacy and Security (SOUPS 2007)*, pages 1–12, July 2007.

[8] Sonia Chiasson, Alain Forget, Robert Biddle, and P. C. Van Oorschot. Influencing users towards better passwords: persuasive cued click-points. In *Proc. of 22nd British*

HCI Group Annual Conference on HCI 2008 (HCI 2008), pages 121–130, September 2008.

[9] Sonia Chiasson, P. C. Van Oorschot, and Robert Biddle. Graphical password authentication using cued click-points. In *Proc. of 12th European Symposium on Research in Computer Security (ESORICS 2007)*, volume 4734, pages 359–374, September 2007.

[10] Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C. Mitchell. Client-side defense against web-based identify theft. In *Proc. of the 11th Annual Network and Distributed System Security Symposium (NDSS '04)*, February 2004.

[11] Darren Davis, Fabian Monrose, and Michael K. Reiter. On user choice in graphical password schemes. In *Proc. of 13th Conference on USENIX Security Symposium (USENIX 2004)*, June 2004.

[12] DIGITALSTRATEGY. http://www.digitalstrategy.govt.nz/Resources/Glossary-of-Key-Terms/.

[13] eBay. ebay toolbar. http://pages.ebay.com/ebay_toolbar/.

[14] McMaster eBusiness Research Centre (MeRC). Measuring identity theft in canada: 2008 consumer survey. http://www.merc-mcmaster.ca/working-papers/measuring-identity-theft-in-canada-2008-consumer-survey/.

[15] Scott R. Fluhrer and Stefan Lucks. Analysis of the E0 cryptosystem. In *Proc. of the 8th Annual International Workshop on Selected Areas in Cryptography (SAC 2001)*, volume 2259, pages 38–48, August 2001.

[16] Jovan Dj. Golic, Vittorio Bagini, and Guglielmo Morgari. Linear cryptanalysis of bluetooth stream cipher. In *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques: Advance in Cryptology (Eurocrypt 02)*, volume 2332, pages 238–255, April 2002.

[17] Jovan Dj. Golic, Vittorio Bagini, and Guglielmo Morgari. Fast algebraic attacks on stream ciphers with linear feedback. In *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques: Advance in Cryptology (Eurocrypt 03)*, volume 2729, pages 162–176, May 2003.

[18] Philippe Golle and David Wagner. Cryptanalysis of a cognitive authentication scheme (extended abstract). In *Proc. of the 2007 IEEE Symposium on Security and Privacy (S&P 2007)*, pages 66–70, May 2007.

[19] Krzysztof Golofit. Click passwords under investigation. In *Proc. of the 12th European Symposium on Research In Computer Security (ESORICS 2007)*, pages 343–358, September 2007.

[20] Anti-Phishing Working Group. Phishing activity trends report q2 2008. http://www.antiphishing.org/.

[21] J.Alex Halderman, Brent Waters, and Edward W. Felten. A convenient method for securely managing passwords. In *Proceedings of the 14th International World Wide Web Conference (WWW 2005)*, pages 471–479, May 2005.

[22] Amir Herzberg and Ahmad Gbara. Trustbar:protecting (even naive) web users from spoofing and phishing attacks. Cryptology ePrint Archive, Report 2004/155, 2004.

[23] Internet Assigned Numbers Authority (IANA). Port numbers. http://www.iana.org/assignments/port-numbers.

[24] Fraudwatch International. How your personal information is used from phishing scams. http://www.fraudwatchinternational.com/phishing-fraud/how-information-used/.

[25] Fraudwatch International. Phishing email methods. http://www.fraudwatchinternational.com/phishing-fraud/phishing-email-methods/.

[26] Fraudwatch International. Phishing web site methods. http://www.fraudwatchinternational.com/phishing-fraud/phishing-web-site-methods/.

[27] Markus Jakobsson. Modeling and preventing phishing attacks. *Financial Cryptography and Data Security*, August 2005.

[28] Markus Jakobsson and Susanne Wetzel. Security weaknesses in bluetooth. In *Proc. of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA (CT-RSA01)*, volume 2020, pages 176–191, April 2001.

[29] Keyghost. http://www.keyghost.com/sx/.

[30] Engin Kirda and Christopher Kruegel. Protecting users against phishing attacks with antiphish. In *Proc. of the 29 Annual International Computer Software and Applications Conference (COMPSAC 2005)*, pages 517–524, July 2005.

[31] Jeremy Kirk. Swedish police warn of tampered credit card terminals. http://www.pcworld.com/article/155525/.html?tk=rss_news, December 2008.

[32] Matthias Krause. Bdd-based cryptanalysis of keystream generators. In *Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques: Advance in Cryptology (Eurocrypt 02)*, volume 2332, pages 222–237, April 2002.

[33] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proc. of the 3rd Symposium on Usable Privacy and Security (SOUPS 2007)*, pages 13–19, July 2007.

[34] John Leyden. Gummi bears defeat fingerprint sensors. *ISI insight*, 11-05, November 2005.

[35] John Leyden. Spear phishers launch targeted attacks. http://www.theregister.co.uk/2005/08/02/ibm_malware_report/, August 2005.

[36] Mohammad Mannan and P.C. van Oorschot. Using a personal device to strengthen password authentication from an untrusted computer. *Financial Cryptography and Data Security*, 4886:88–103, February 2007.

[37] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. 2001.

[38] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

[39] Netcraft. Netcraft toolbar. http://toolbar.netcraft.com/.

[40] Allan Paivio, T. B. Rogers, and PADRIC C. Smythe. Why are pictures easier to recall than words? *Psychonomic Science*, 11(4):137–138, 1968.

[41] Bryan Parno, Cynthia Kuo, and Adrian Perrig. Phoolproof phishing prevention. *Financial Cryptography and Data Security*, 4107:1–19, February 2006.

[42] Passfaces. http://www.realuser.com/.

[43] phish. Oxford english dictionary online. http://dictionary.oed.com/cgi/entry/30004303/.

[44] Royal Canadian Mounted Police. Identity theft and identity fraud. http://www.rcmp-grc.gc.ca/scams-fraudes/id-theft-vol-eng.htm.

[45] Niels Provos and David Mazieres. A future-adaptable password scheme. In *Proc. of the Annual Conference on USENIX Annual Technical Conference*, June 1999.

[46] Paul F. Roberts. Spear phishing attack targets credit unions. http://www.eweek.com/c/a/Security/Spear-Phishing-Attack-Targets-Credit-Unions/, December 2005.

[47] Blake Ross, Collin Jackson, Nick Miyake, Dan Boneh, and John C. Mitchell. Stronger password authentication using browser extensions. In *Proc. of the 14th Conference on USENIX Security Symposium (USENIX 2005)*, pages 17–32, April 2005.

[48] Volker Roth, Kai Richter, and Rene Freidinger. A pin-entry method resilient against shoulder surfing. In *Proc. of 11th ACM Conference on Computer and Communication Security (CCS 2004)*, pages 236–245, October 2004.

[49] Schlage. Scramble keypad reader (SERIII-w).

[50] SearchMidmarketSecurity.com. Keylogger. http://searchmidmarketsecurity.techtarget .com/sDefinition/0,,sid198_gci962518,00.html.

[51] Roger N. Shepard. Recognition memory for words, sentences, and pictures. *Journal of Verbal Learning Verbal Behavior*, 6:156–163, 1967.

[52] Peipei Shi, Bo Zhu, and Amr M. Youssef. A PIN entry scheme resistant to recording-based shoulder-surfing. In *Proc. of the Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '09)*, pages 237–241, June 2009.

[53] Peipei Shi, Bo Zhu, and Amr M. Youssef. A rotary PIN entry scheme resilient to shoulder-surfing. In *Proc. of the 4th International Conference for Internet Technology and Secured Transactions (ICITST-2009)*, November 2009.

[54] silicon.com. Voice over ip. http://www.silicon.com/research/specialreports/voip /0,3800004463,39128854,00.htm.

[55] SpoofStick. Spoofstick home. http://www.spoofstick.com/.

[56] Chris Summers and Sarah Toyne. Gangs preying on cash machines. BBC NEWS Online, October 2003.

[57] Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. Graphical passwords: A survey. In *Proc. of Annual Computer Security Applications Conference (ACSAC'05)*, pages 463–472, December 2005.

[58] Swivel. PINsafe.

[59] Julie Thorpe and P. C. van Oorschot. Human-seeded attacks and exploiting hot-spots in graphical passwords. In *Proc. of 16th USENIX Security Symposium (USENIX 2007)*, pages 1–16, June 2007.

[60] Edward K. Vogel and Maro G. Machizawa. Neural activity predicts individual differences in visual working memory capacity. *Nature*, 428:748–751, April 2004.

[61] Daphna Weinshall. Cognitive authentication schemes safe against spyware (short paper). In *Proc. of the 2006 IEEE Symposium on Security and Privacy (S&P 2006)*, pages 295–300, May 2006.

[62] Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. Authentication using graphical passwords: effects of tolerance and image choice. In *Proc. of 2005 Symposium on Usable Privacy and Security (SOUPS 2005)*, July 2005.

[63] Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. Passpoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, 63:102–127, July 2005.

[64] Susan Wiedenbeck, Jim Waters, Leonardo Sobrado, and Jean-Camille Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proc. of the Working Conference on Advanced Visual Interfaces (AVI 2006)*, pages 177–184, May 2006.

[65] Wikipedia. Blue pill (malware). http://en.wikipedia.org/wiki/Blue_Pill_(malware).

[66] Wikipedia. E0 (cipher). http://en.wikipedia.org/wiki/E0_(cipher).

[67] Wikipedia. Hardware keylogger. http://en.wikipedia.org/wiki/Hardware_keylogger.

[68] Wikipedia. Keystroke logging. http://en.wikipedia.org/wiki/Keystroke_-logging#Hardware-based_keyloggers.

[69] Wikipedia. Likert scale. http://en.wikipedia.org/wiki/Likert_scale.

[70] Min Wu, Simson Garfinkel, and Rob Miller. Users are not dependable - how to make security indicators to better protect them. Talk presented at the Workshop for Trustworthy Interfaces for Passwords and Personal Information, June 2005.

[71] John C. Yuille. *Imagery, memory, and cognition: Essays in honor of Allan Paivio*, chapter 3, pages 65–89. Lawrence Erlbaum Assoc Inc, 1983.