3-1-2018

A Shoulder Surfing Resistant Graphical Authentication System

Hung-Min Sun
National Tsing Hua University

Shiuan-Tung Chen
National Tsing Hua University

Jyh-Haw Yeh
Boise State University

Chia-Yun Cheng
National Tsing Hua University

A Shoulder Surfing Resistant Graphical Authentication System

Hung-Min Sun, Chia-Yun Cheng, Jyh-Haw Yeh and Shiuan-Tung Chen

Abstract—Authentication based on passwords is used largely in applications for computer security and privacy. However, human actions such as choosing bad passwords and inputting passwords in an insecure way are regarded as ”the weakest link” in the authentication chain. Rather than arbitrary alphanumeric strings, users tend to choose passwords either short or meaningful for easy memorization. With web applications and mobile apps piling up, people can access these applications anytime and anywhere with various devices. This evolution brings great convenience but also increases the probability of exposing passwords to shoulder surfing attacks. Attackers can observe directly or use external recording devices to collect users' credentials. To overcome this problem, we proposed a novel authentication system PassMatrix, based on graphical passwords to resist shoulder surfing attacks. With a one-time valid login indicator and circulative horizontal and vertical bars covering the entire scope of pass-images, PassMatrix offers no hint for attackers to figure out or narrow down the password even they conduct multiple camera-based attacks. We also implemented a PassMatrix prototype on Android and carried out real user experiments to evaluate its memorability and usability. From the experimental result, the proposed system achieves better resistance to shoulder surfing attacks while maintaining usability.

Index Terms—Graphical Passwords, Authentication, Shoulder Surfing Attack.

1 Introduction

Textual passwords have been the most widely used authentication method for decades. Comprised of numbers and upper- and lower-case letters, textual passwords are considered strong enough to resist against brute force attacks. However, a strong textual password is hard to memorize and recollect [1]. Therefore, users tend to choose passwords that are either short or from the dictionary, rather than random alphanumeric strings. Even worse, it is not a rare case that users may use only one username and password for multiple accounts [2]. According to an article in Computer world, a security team at a large company ran a network password cracker and surprisingly cracked approximately 80% of the employees' passwords within 30 seconds [3]. Textual passwords are often insecure due to the difficulty of maintaining strong ones.

Various graphical password authentication schemes [4], [5], [6], [7] were developed to address the problems and weaknesses associated with textual passwords. Based on some studies such as those in [8], [9], humans have a better ability to memorize images with long-term memory (LTM) than verbal representations. Image-based passwords were proved to be easier to recollect in several user studies [10], [11], [12]. As a result, users can set up a complex authentication password and are capable of recollecting it after a long time even if the memory is not activated periodically. However, most of these image-based passwords are vulnerable to shoulder surfing attacks (SSAs). This type of attack either uses direct observation, such as watching over someone's shoulder or applies video capturing techniques to get passwords, PINs, or other sensitive personal information [13], [14], [15].

The human actions such as choosing bad passwords for new accounts and inputting passwords in an insecure way for later logins are regarded as the weakest link in the authentication chain [16]. Therefore, an authentication scheme should be designed to overcome these vulnerabilities.

In this paper, we present a secure graphical authentication system named PassMatrix that protects users from becoming victims of shoulder surfing attacks when inputting passwords in public through the usage of one-time login indicators. A login indicator is randomly generated for each pass-image and will be useless after the session terminates. The login indicator provides better security against shoulder surfing attacks, since users use a dynamic pointer to point out the position of their passwords rather than clicking on the password object directly.

1.1 Motivation

As the mobile marketing statistics compilation by Danyl, the mobile shipments had overtaken PC shipments in 2011, and the number of mobile users also overtaken desktop users at 2014, which closed to 2 billion [17]. However, shoulder surfing attacks have posed a great threat to users' privacy and confidentiality as mobile devices are becoming indispensable in modern life. People may log into web services and apps in public to access their personal accounts with their smart phones, tablets or public devices, like bank ATM. Shoulder-surfing attackers can observe how the passwords were entered with the help of reflecting glass windows, or let alone monitors hanging everywhere in public places. Passwords are exposed to risky environments, even if the passwords themselves are complex and secure. A secure authentication system should be able to defend against shoulder surfing attacks and should be applicable to all kinds of devices. Authentication schemes in the literature such as those in [6], [18], [19], [20], [21], [22], [23], [24], [25] are resistant to shoulder-surfing, but they have either usability limitations or small password space. Some of them are not suitable to be applied in mobile devices and most of them can be easily compromised to shoulder surfing attacks if attackers use video capturing techniques like Google Glass [15], [26]. The limitations of usability include issues such as taking more time to log in, passwords being too difficult to recall after a period of time, and the authentication method being too complicated for users without proper education and practice.

In 2006, Wiedenbeck et al. proposed PassPoints [7] in which the user picks up several points (3 to 5) in an image during the password creation phase and re-enters each of these pre-selected click-points in a correct order within its tolerant square during the login phase. Comparing to traditional PIN and textual passwords, the PassPoints scheme substantially increases the password space and enhances password memorability. Unfortunately, this graphical authentication scheme is vulnerable to shoulder surfing attacks. Hence, based on the PassPoints, we add the idea of using one-time session passwords and distractors to develop our PassMatrix authentication system that is resistant to shoulder surfing attacks.

1.2 Organization

This paper is organized as follows. Section 2 provides the backgrounds of related techniques about graphical authentication schemes and Section 3 describes attack models. The proposed PassMatrix is presented in Section 4. The user study and its results are available in Section 5 and Section 6 respectively. A security analysis is discussed in Section 7. Section 8 concludes the paper.

2 Background and Related Work

In the past several decades, a lot of research on password authentication has been done in the literature. Among all of these proposed schemes, this paper focuses mainly on the graphical-based authentication systems. To keep this paper concise, we will give a brief review of the most related schemes that were mentioned in the previous section. Many other schemes such as those in [27], [28], [29], [30], [31] may have good usability, they are not graphical-based and need additional support from extra hardware such as audio, multi-touch monitor, vibration sensor, or gyroscope, etc.

In the early days, the graphical capability of handheld devices was weak; the color and pixel it could show was limited. Under this limitation, the Draw-a-Secret (DAS) [6] technique was proposed by Jermyn et al. in 1999, where the user is required to re-draw a pre-defined picture on a 2D grid. We directly extract the figure from [6] and show it in Figure 1(b). If the drawing touches the same grids in the same sequence, then the user is authenticated. Since then, the graphical capability of handheld devices has steadily and ceaselessly improved with the advances in science and technology. In 2005, Susan Wiedenbeck et al. introduced a graphical authentication scheme PassPoints [7], and at that time, handheld devices could already show high resolution color pictures. Using the PassPoint scheme, the user has to click on a set of pre-defined pixels on the predestined photo, as shown in Figure 1(a) (this figure is extracted from [7]), with a correct sequence and within their tolerant squares during the login stage. Moreover, Marcos et al. also extended the DAS based on finger-drawn doodles and pseudosignatures in recent mobile device [32], [33]. This authentication system is based on features which are extracted from the dynamics of the gesture drawing process (e.g., speed or acceleration). These features contain behavioral biometric characteristic. In other words, the attacker would have to imitate not only what the user draws, but also how the user draws it. However, these three authentication schemes are still all vulnerable to shoulder surfing attacks as they may reveal the graphical passwords directly to some unknown observers in public.

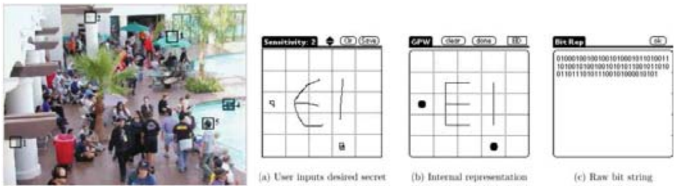

Fig. 1. (a) Pixel squares selected by users as authentication passwords in PassPoints [7]. (b) Authentication password drew by users and the raw bits recorded by the system database [6].

In addition to graphical authentication schemes, there was some research on the extension of conventional personal identification number (PIN) entry authentication systems. In 2004, Roth et al. [34] presented an approach for PIN entry against shoulder surfing attacks by increasing the noise to observers. In their approach, the PIN digits are displayed in either black or white randomly in each round. The user must respond to the system by identifying the color for each password digit. After the user has made a series of binary choices (black or white), the system can figure out the PIN number the user intended to enter by intersecting the user′s choices. This approach could confuse the observers if they just watch the screen without any help of video capturing devices. However, if observers are able to capture the whole authentication process, the passwords can be cracked easily.

In order to defend the shoulder surfing attacks with video capturing, FakePointer [35] was introduced in 2008 by T. Takada. We use Figure 2 (from [35]) below to show the usage of FakePointer. In addition to the PIN number, the user will get a new ”answer indicator” each time for the authentication process at a bank ATM. In other words, the user has two secrets for authentication: a PIN as a fixed secret and an answer indicator as a disposable secret. The answer indicator is a sequence of n shapes if the PIN has n digits. At each login session, the FakePointer interface will present the user an image of a numeric keypad with 10 numbers (similar to the numeric keypad for phones), with each key (number) on top of a randomly picked shape. The numeric keys, but not the shapes, can be moved circularly using the left or right arrow keys. During authentication, the user must repeatedly move numeric keys circularly as shown in the leftmost figure in Figure 2, until the first digit of the PIN overlaps the first shape of the answer indicator on the keypad and then confirm a selection by pressing the space key. This operation is repeated until all the PIN digits are entered and confirmed. This approach is quite robust even when the attacker captures the whole authentication process. However, there is still room to improve the password space. For example, if the device used for authentication is a smartphone, a tablet or a computer rather than a bank ATM, the password space can be enlarged substantially since the PIN could be any combination of alphanumeric characters rather than just numeric digits.

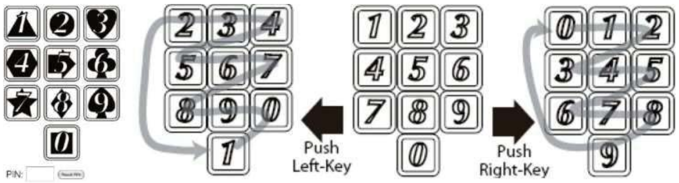

Fig. 2. FakePointer, where a user can move a numeric key layout circularly using right and left arrow keys. [35]

Wiedenback et al. [36] described a graphical password entry scheme in 2006, as shown in Figure 3(b) (the figure is extracted from [36]). This scheme is resistant to shoulder surfing attacks using a convex hull method. The user needs to recognize a set of pass-icons on the screen and clicks inside the convex hull formed by all these pass-icons. In order to make the password hard to guess, a large number of other different icons can be inserted into the screen to increase the password space. However, a large number of objects will crowd the display and may make objects indistinguishable.

In 2010, David Kim et al. [25] proposed a visual authentication scheme for tabletop interfaces called ”Color Rings”, as shown in Figure 3(a) (the figure is extracted from [25]), where the user is assigned i authentication (key) icons, which are collectively assigned one of the four color-rings: red, green, blue, or pink. During login, i grids of icons are provided, with 72 icons being displayed per grid. There is only one key icon presented in each grid. The user must drag all four rings (ideally with index finger and thumb from two hands) concurrently and place them in the grid. The distinct key icon should be captured by the correct color ring while the rest of rings just make decoy selections. The user confirms a selection by dropping the rings in position. The rings are large enough to include more than one icon and can thus obfuscate the direct observer. Unfortunately, these kinds of passwords can be cracked by intersecting the user’s selections in each login because the color of the assigned ring is fixed and a ring can include at most seven icons. Thus, the attacker only requires a limited number of trials to guess the user’s password.

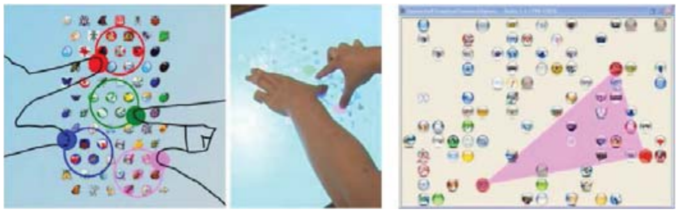

Fig. 3. (a) Color Rings method [25]. (b) Convex Hull method [36].

3 Problem Statement, Attack Model and Assumptions

3.1 Problem Statement

With the increasing amount of mobile devices and web services, users can access their personal accounts to send confidential business emails, upload photos to albums in the cloud or remit money from their e-bank account anytime and anywhere. While logging into these services in public, they may expose their passwords to unknown parties unconsciously. People with malicious intent could watch the whole authentication procedure through omnipresent video cameras and surveillance equipment, or even a reflected image on a window [37]. Once the attacker obtains the password, they could access personal accounts and that would definitely pose a great threat to one’s assets. Shoulder surfing attacks have gained more and more attention in the past decade. The following lists the research problems we would like to address in this study:

1) The problem of how to perform authentication in public so that shoulder surfing attacks can be alleviated.
2) The problem of how to increase password space than that of the traditional PIN.
3) The problem of how to efficiently search exact password objects during the authentication phase.
4) The problem of requiring users to memorize extra information or to perform extra computation during authentication.
5) The problem of limited usability of authentication schemes that can be applied to some devices only.

3.2 Attack Model

3.2.1 Shoulder Surfing Attacks

Based on previous research [20], [21], [25], [34], [35], users’ actions such as typing from their keyboard, or clicking on the pass-images or pass-points in public may reveal their passwords to people with bad intention. In this paper, based on the means the attackers use, we categorize shoulder-surfing attacks into three types as below:

1) Type-I: Naked eyes.
2) Type-II: Video captures the entire authentication process only once.
3) Type-III: Video captures the entire authentication process more than once.

The latter types of attacks require more effort and techniques from attackers. Thus, if an authentication scheme is able to resist against these attacks, it is also secure against previous types of attacks. Some of the proposed authentication schemes [4], [5], [6], [7], [25], [38], including traditional text-password and PIN, are vulnerable to shoulder surfing Type-I attacks and thus are also subject to Type-II and Type-III attacks. These schemes reveal passwords to attackers as soon as users enter their passwords by directly pressing or clicking on specific items on the screen. Other schemes such as those in [19], [34] can resist against Type-I but are vulnerable to Type-II and Type-III attacks since the attackers can crack passwords by intersecting their video captures from multiple steps of the entire authentication process.

3.2.2 Smudge Attacks

According to a previous study [39], authentication schemes that require users to touch or fling on computer monitors or display screens during the login phase are vulnerable to smudge attacks. The attacker can obtain the user's password easily by observing the smudge left on the touch screen (see Figure 4 which is directly extracted from [39]).

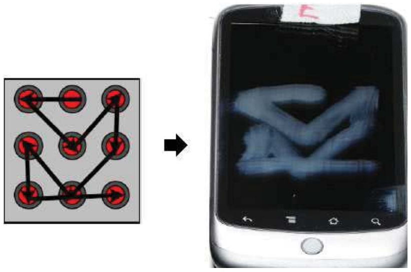

Fig. 4. (a) Android pattern screen lock in which a user draws a personal unlock pattern that connects at least four dots on screen [39]. (b) The residue from fingerprints left on the screen [39].

3.3 Assumptions

In this paper, we do not discuss the habitual movements and the preference of users that the attacker may take advantage of to figure out the potential passwords. In addition, we have four assumptions in this study:

1) Any communication between the client device and the server is protected by SSL so that packets or information will not be eavesdropped or intercepted by attackers during transmission.
2) The server and the client devices in our authentication system are trustworthy.
3) The keyboard and the entire screen of mobile devices are difficult to protect, but a small area (around 1.5 cm^2) is easy to be protected from malicious people who might shoulder surf passwords.
4) Users are able to register an account in a place that is safe from observers with bad intention or surveillance cameras that are not under proper management.

4 PassMatrix

To overcome (1) the security weakness of the traditional PIN method, (2) the easiness of obtaining passwords by observers in public, and (3) the compatibility issues to devices, we introduced a graphical authentication system called PassMatrix. In PassMatrix, a password consists of only one pass-square per pass-image for a sequence of n images. The number of images (i.e., n) is user-defined. Figure 5 demonstrates the proposed scheme, in which the first pass-square is located at (4, 8) in the first image, the second pass-square is on the top of the smoke in the second image at (7, 2), and the last pass-square is at (7, 10) in the third image.

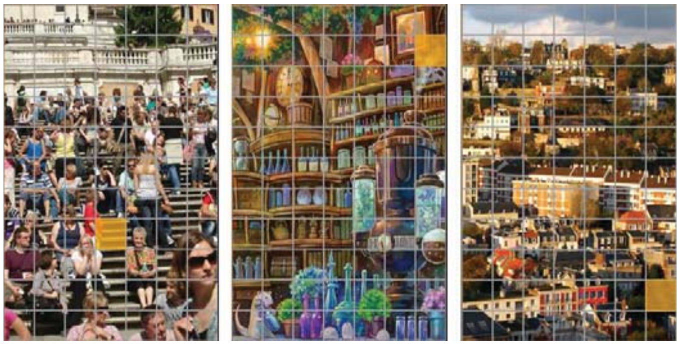

Fig. 5. A password contains three images (n=3) with a pass square in each. The pass squares are shown as the orange-filled area in each image.

In PassMatrix, users choose one square per image for a sequence of n images rather than n squares in one image as that in the PassPoints [7] scheme. Based on the user study of Cued Click Points (CCP) [40] proposed by Chiasson et al., the CCP method does a good job in helping users recollect and remember their passwords. If the user clicks on an incorrect region within the image, a different image will be shown to give the user a warning feedback. However, aiming at alleviating shoulder surfing attacks, we do not recommend this approach since the feedback that is given to users might also be obtained by attackers.

Due to the fact that people do not register a new account or set up a new screen lock frequently, we assume that these setup events can be done in a safe environment rather than in public places. Thus, users can pick up pass-squares by simply touching at or clicking on them during the registration phase.

4.1 Overview

PassMatrix is composed of the following components (see Figure 6):

- Image Discretization Module
- Horizontal and Vertical Axis Control Module
- Login Indicator generator Module
- Communication Module
- Password Verification Module
- Database

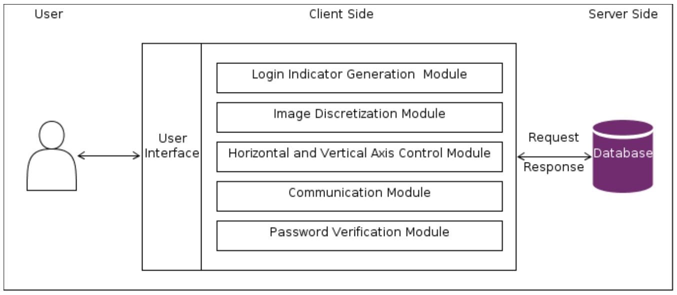
User
Client Side
Server Side
User Interface
Login Indicator Generation Module
Image Discretization Module
Horizontal and Vertical Axis Control Module
Communication Module
Password Verification Module
Request
Response
Database

Fig. 6. Overview of the PassMatrix system.

Image Discretization Module. This module divides each image into squares, from which users would choose one as the pass-square. As shown in Figure 5, an image is divided into a 7×11 grid. The smaller the image is discretized, the larger the password space is. However, the overly concentrated division may result in recognition problem of specific objects and increase the difficulty of user interface operations on palm-sized mobile devices. Hence, in our implementation, a division was set at 60-pixel intervals in both horizontal and vertical directions, since 60 pixels^2 is the best size to accurately select specific objects on touch screens.

Login Indicator Generator Module. This module generates a login indicator consisting of several distinguishable characters (such as alphabets and numbers) or visual materials (such as colors and icons) for users during the authentication phase. In our implementation, we used characters A to G and 1 to 11 for a 7×11 grid. Both letters and numbers are generated randomly and therefore a different login indicator will be provided each time the module is called. The generated login indicator can be given to users visually or acoustically. For the former case, the indicator could be shown on the display (see Figure 7(a)) directly or through another predefined image. If using a predefined image, for instance, if the user chooses the square $(5, 9)$ in the image as in Figure 7(b), then the login indicator will be (E, 11). For the acoustical delivery, the indicator can be received by an audio signal through the ear buds or Bluetooth. One principle is to keep the indicators secret from people other than the user, since the password (the sequence of pass-squares) can be reconstructed easily if the indicators are known.

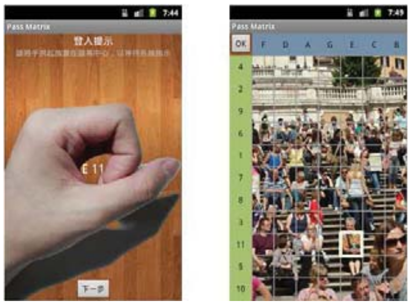

Fig. 7. (a) Obtain the login indicator (E,11) directly. (b) Obtain the login indicator through a predefined image.

Horizontal and Vertical Axis Control Module. There are two scroll bars: a horizontal bar with a sequence of letters and a vertical bar with a sequence of numbers. This control module provides **drag** and **fling** functions for users to control both bars. Users can fling either bar using their finger to shift one alphanumeric at a time. They can also shift several checks at a time by dragging the bar for a distance. Both bars are circulative, i.e., if the user shifts the horizontal bar in Figure 8(c) to left by three checks, it will become the bar shown in Figure 8(d). The bars are used to implicitly point out (or in other words, align the login indicator to) the location of the user's pass-square.

Communication Module. This module is in charge of all the information transmitted between the client devices and the authentication server. Any communication is protected by SSL (Secure Socket Layer) protocol [41] and thus, is safe from being eavesdropped and intercepted.

Password Verification Module. This module verifies the user password during the authentication phase. A pass-square acts similar to a password digit in the text-based password system. The user is authenticated only if each pass-square in each pass-image is correctly aligned with the login indicator. The details of how to align a login indicator to a pass-square will be described in the next section.

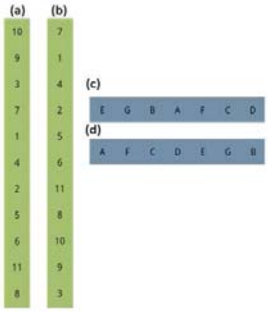

Fig. 8. Horizontal scroll bar (on the right/blue) and vertical bar (on the left/green).

Database. The database server contains several tables that store user accounts, passwords (ID numbers of pass-images and the positions of pass-squares), and the time duration each user spent on both registration phase and login phase. PassMatrix has all the required privileges to perform operations like insert, modify, delete and search.

4.2 PassMatrix

PassMatrix's authentication consists of a registration phase and an authentication phase as described below:

4.2.1 Registration phase

Figure 9 is the flowchart of the registration phase. At this stage, the user creates an account which contains a username and a password. The password consists of only one pass-square per image for a sequence of n images. The number of images (i.e., n) is decided by the user after considering the trade-off between security and usability of the system [42]. The only purpose of the username is to give the user an imagination of having a personal account. The username can be omitted if PassMatrix is applied to authentication systems like screen lock. The user can either choose images from a provided list or upload images from their device as pass-images. Then the user will pick a pass-square for each selected pass-image from the grid, which was divided by the image discretization module. The user repeats this step until the password is set.

4.2.2 Authentication phase

Figure 10 is the flowchart of the authentication phase. At this stage, the user uses his/her username, password and login indicators to log into PassMatrix. The following describes all the steps in detail:

1) The user inputs his/her username which was created in the registration phase.
2) A new indicator comprised of a letter and a number is created by the login indicator generator module. The indicator will be shown when the user uses

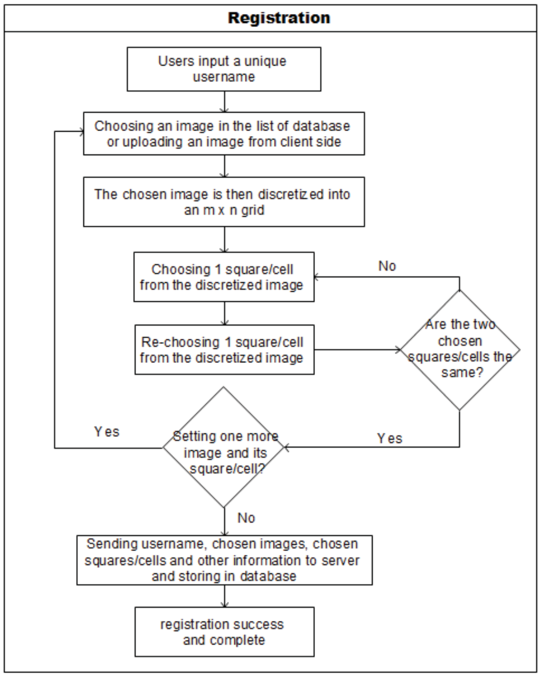
Registration
Users input a unique username
Choosing an image in the list of database or uploading an image from client side
The chosen image is then discretized into an m x n grid
Choosing 1 square/cell from the discretized image
No
Re-choosing 1 square/cell from the discretized image
Are the two chosen squares/cells the same?
Yes
Setting one more image and its square/cell?
Yes
No
Sending username, chosen images, chosen squares/cells and other information to server and storing in database
registration success and complete

Fig. 9. The flowchart of registration phase in PassMatrix.

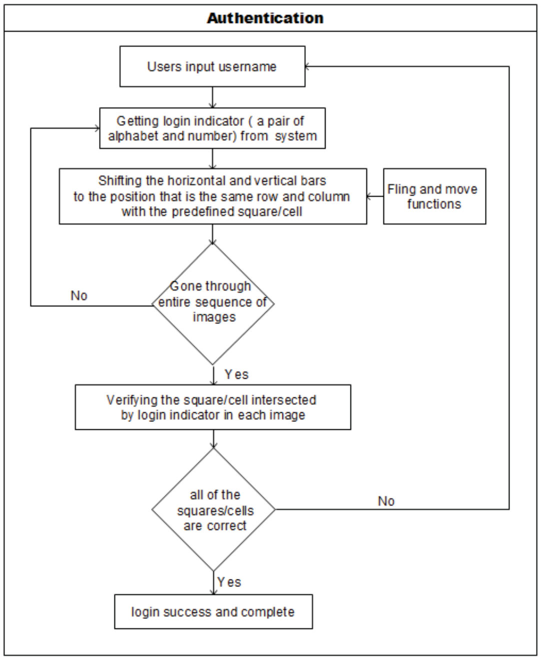
Authentication
Users input username
Getting login indicator (a pair of alphabet and number) from system
Shifting the horizontal and vertical bars to the position that is the same row and column with the predefined square/cell
Fling and move functions
Gone through entire sequence of images
No
Yes
Verifying the square/cell intersected by login indicator in each image
all of the squares/cells are correct
No
Yes
login success and complete

Fig. 10. The flowchart of authentication phase in PassMatrix.

his/her hand to form a circle and then touch the screen. In this case, the indicator is conveyed to the user by visual feedback. The indicator can also be delivered through a predefined image or by audio feedback that we have mentioned in the previous section.

3) Next, the first pass-image will be shown on the display, with a horizontal bar and a vertical bar on its top and left respectively. To respond to the challenge, the user flings or drags the bars to align the pre-selected pass-square of the image with the login indicator. For example, if the indicator is (E, 11) and the pass-square is at (5, 7) in the grid of the image, the user shifts the character ”E” to the 5th column on the horizontal bar and ”11” to the 7th row on the vertical bar (see Figure 12).
4) Repeat step 2 and step 3 for each pre-selected pass-image.
5) The communication module gets user account information from the server through HttpRequest POST method.
6) Finally, for each image, the password verification module verifies the alignment between the pass-square and the login indicator. Only if all the alignments are correct in all images, the user is allowed to log into PassMatrix.

5 Implementation and User Study

Although the PassMatrix prototype was implemented on an Android system which has a small screen, it is not limited to the applications on small screen devices, for example screen locking. In fact, it could be applied to a wide range of authentication scenarios. For instance, user account login in Windows 8, email account login on web browser, and application login/unlock on Android OS. It can also be applied to any client device such as personal computers, laptops, tablets, mobile phones, or bank ATM due to the fact that the method of authentication is simple and the entire authentication process can be completed by only touching or clicking on the screen.

In our implementation, we assumed that users download an application from Google Play [43] and register an account for later login to use the service. Since Android [44] is an open source operating system based on Linux kernel and is widely used in mobile devices such as tablet PCs and smart phones, we implemented a PassMatrix prototype on Android and carried out user experiments to evaluate its memorability and usability. In this section we will describe our PassMatrix implementation and the user study experimental design, environment, participants and procedures. The result of the user study will be shown in Section 6.

5.1 Implementation

The PassMatrix prototype was built with Android SDK 2.3.3 since it was the mainstream version of the distribution in 2012 [45]. After connecting to the Internet, users can register an account, log in a few times in practice mode, and then log in for the experiment with a client’s device (see Figure 11(a)). In the client side of our prototype, we used XML to build the user interface and used JAVA and Android API to implement functions, including username checking, pass-images listing, image discretization, pass-squares selection, login indicator delivery, and the horizontal and vertical bars circulation. In the server side of our implementation, we used PHP and MySQL to store and fetch registered accounts to/from the database to handle the password verification.

Although in our proposed system we mentioned that users can import their own images, we used a list of 24 fixed test images in our experiment (see Figure 11(b)). Each image

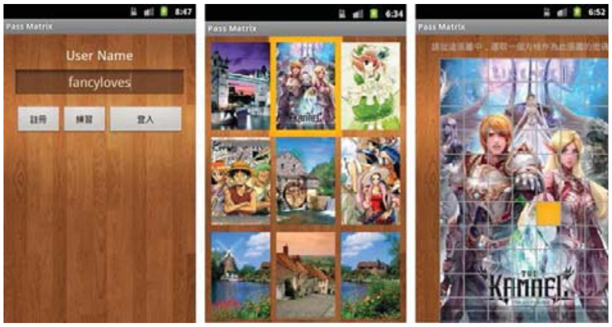

Fig. 11. (a) The Main page of PassMatrix, users can register an account, practice or start to log in for experiment. (b) Users can choose from a list of 24 images as their pass-images. (c) There are 7×11 squares in each image, from which users choose one as the pass-square.

is displayed in a size of 420×660 pixels and is discretized into 60×60 pixel squares. Thus, users have 7×11 squares to select in each image (see Figure 11(c)). After a user selects three to five images with one pass-square per image, the password will be stored as a list of coordinates in a database table (i.e., the locations of those selected pass-squares in the 7×11 grid). The password space depends on the number of images set by users. For instance, if a user creates an account with four images, the password space is $(7 \times 11)^4$.

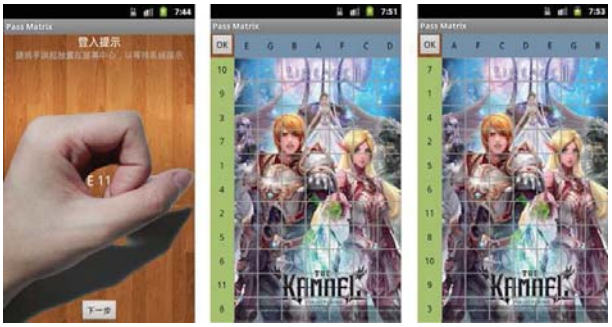

Fig. 12. (a) A visual way for users to obtain a one-time valid login indicator. (b) The permutations of alphanumerics in horizontal and vertical bars are randomly generated for each image. (c) Users can shift the bars to the correct position so that the login indicator aligns with the pass-square.

The first step in the login phase is getting the one-time valid login indicator from the system. There are many ways to obtain the indicator and we've illustrated several examples in Section 4.1. In our implementation, we adopted the simplest way: grasping the hand with a little space left in the center and then touching the screen of smart phones (see Figure 12(a)). To protect against shoulder surfing, the indicator is not shown until the hand touches the screen and will vanish immediately when the hand leaves the screen.

The number of elements on both the horizontal and vertical bars depends on the discretization degree of the images. In our implementation, there are 7 letters (from A to G) and 11 numbers (from 1 to 11) on the horizontal bar and on the vertical bar, respectively. They are used to align the one-time indicator with the pass-square in each pass-image during the authentication phase. In order to obfuscate and thus hide the alignment patterns from observers, we randomly shuffled the elements on both bars in each pass-image and let users shift them to the right position (see Figure 12(b) and (c)). We implemented two bar-shifting functions: dragging and flinging. Since the entire bar is shiftable and can be circulated on either side (i.e., bi-directional and circulative), users do not need to place their finger on a specific element in order to move it.

5.2 User Study

In this section, we introduce our user study experimental design, environment, participants and the detailed procedure that evaluates the accuracy and usability of PassMatrix.

5.2.1 Experimental Design

We conducted a user study for the proposed system to evaluate two performance metrics:

- Password memorability/recollection: How well do users remember their password and can they log into the system successfully after a period of time since registration?
- Usability: We measured the users' experience on PassMatrix, which includes the total time consumed for both registration and authentication, the successful login rate if they know their passwords, and the number of times users use the bar control functions to shift the elements to the right position during the authentication phase.

In order to analyze the memorability, we asked the experiment participants to register a PassMatrix account first and then come back to log into their account two weeks later. We marked a login attempt as a failure if the number of retries exceeded 5 times.

To analyze the usability, we recorded the time each participant spent on both registration and login to see whether the PassMatrix's authentication is time-consuming. We also recorded the number of times participants fling their finger to shift the horizontal and vertical bars to measure the effort they have to make during the authentication phase. Furthermore, to calculate the successful login rate, we asked the participants to log into PassMatrix three times right after they register an account in the first session and re-log into the system another three times in the second session (two weeks later). We can use these statistic data from each participant to evaluate how well users operate on the system's authentication interface. The detailed procedure is described in Section 5.2.4 and the results will be shown in Section 6.

5.2.2 Environment

We deployed and installed PassMatrix.apk on Samsung Nexus S with Android version 2.3.6 and a display size of 480×800 pixels. MySQL $5.0.51$ and PHP were installed on a server with Intel(R) Core(TM) 2 Quad CPU, 3GB RAM and Ubuntu $8.04.4$ OS.

5.2.3 Participants

30 novice users (11 females and 19 males), who are unfamiliar with PassMatrix or even graphical authentication schemes, were recruited from our university to participate in this study. At the time of this study, the participants are

24.53 years old on average (StdDev=3.14), in which three of them are post-doctors and the majority of the rest are graduate students. All participants are either in Computer Science, Information System Management or other information technology majors. 76.67% of them have a background of information security. As Figure 13(b) shows, 73 % of participants have less than one year or no experience at all of using smart phones, whereas 20% of participants have more than one but less than two years of experience and the final 7% are veteran users with more than two years of experience. The survey showed that in the past, they went through authentication processes on an average of 12.6 times a day in public (see Figure 13(a)).

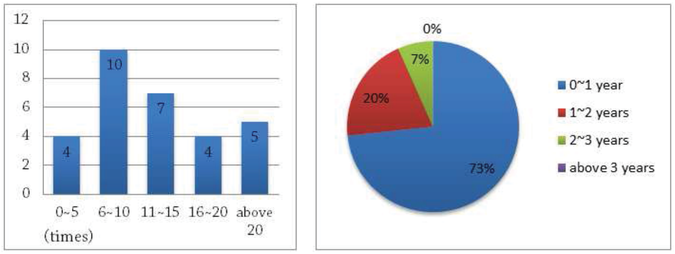

Fig. 13. Trends of (a) Number of times a participant went through authentication processes in public per day, and (b) User experience in smart phones with touch screens.

5.2.4 Procedure

Participants carried out the usability study in two sessions - an initial session and a follow-up session. In the first session, participants were asked to create an account in PassMatrix. The following describes how the first session was conducted in detail.

1) Introduction phase: We explained the basic idea and purpose of PassMatrix with a presentation and showed participants how to use the system with some simple animations.
2) Registration phase: Participants created an account consisting of a username and a password in PassMatrix. In the introduction phase, participants were educated by our tutorial so that
   a) They knew that they should register their account in a private place. Hence it is safe to choose pass-squares by simply clicking on them during the registration phase.
   b) They knew that they should choose the pass-squares that do not contain light objects but are meaningful to them.
   c) They knew that they should re-choose the chosen square in each pass-image for confirmation.
   d) They knew that they should set three or more pass-images.
3) Practice phase: Participants were told to log into their account in a practice mode. They repeated this step until they thought they knew how to control the horizontal and vertical bars. The PassMatrix system gives the authentication feedback to users only after the whole password input process is completed, not in between each pass-image.
4) Login phase: After practicing, participants were requested to log into their account formally in a login mode.
5) Participants were also asked to answer a short demographic questionnaire about some simple personal data and their personal experience on mobile phones or authentication systems.
6) Each participant was then given an answer sheet, containing the information of a third person's two previous login records. Participants were asked to figure out the third person's pass-squares from these two given login records. An incentive gift was provided if they are able to successfully crack the password in ten tries (i.e., ten guesses on the answer sheet). Two weeks were given to crack the password.

In the follow-up session (two weeks later), participants were asked to

1) log into PassMatrix repeatedly until three successful logins,
2) answer another questionnaire sheet about their user experience on PassMatrix, and
3) turn in their answer sheet for the password cracking experiment.

6 Collected Results

We analyzed the collected data from our experiments and surveys to evaluate the effectiveness of the proposed system. The results are presented in two perspectives: accuracy and usability. The accuracy perspective focuses on the successful login rates in both sessions, including the practice logins. The usability perspective is measured by the amount of time users spent in each PassMatrix phase. The results of these two analyses strongly suggested that PassMatrix is practical to use. At the end of this section, we also presented the statistics of the survey data from participants about their personal background and user experience on smart phones and PassMatrix.

6.1 Accuracy

In the practice phase of the first session, participants practiced the login process on an average of 4 times ranging from 1 to 14 (excluding one outlier) and then moved onto the authentication (login) phase. As we defined in the previous section, participants can keep trying to log in to their account until they have failed six times. In other words, a successful attempt means that a user, in less than or equal to six tries, is able to pass the authentication with a correct password. If all six tries failed, this attempt will be marked as failure. Below, we define two terms First Accuracy and Total Accuracy that were used in our experiment:

$$First\ Accuracy = \frac{Successful\ attempts\ in\ first\ Try}{Total\ attempts} \tag{1}$$

$$Total\ Accuracy = \frac{Successful\ attempts}{Total\ attempts} \qquad (2)$$

TABLE 1
The accuracy of practice/authentication(login) in two sessions

	First session		Second session	
	First	Total	First	Total
Practice Phase	60.00%	100%	-	-
Login Phase	86.67%	100%	66.67%	93.33%

Table 1 shows the First Accuracy and the Total Accuracy of the practice and login phases in both sessions with 30 participants. On average, 3.2 pass-images were selected by each participant. The result shows that both the First and Total Accuracies in the first session are higher than those in the second session. In the first session, 26 out of 30 (86.67%) participants were able to log into the system successfully with just one try and all of them were authenticated within six tries (i.e., the Total Accuracy is 100%). After more than two weeks (for an average of 16.3 days), the First Accuracy in the second session was down to 66.67%, but the Total Accuracy is still 93.33%. We surveyed the participants for the possible reasons of the big drop in the First Accuracy and also analyzed those failed login attempts in the second session. We found out that the participants did not really forget their passwords. Most of them still remember the locations of their pass-squares. However, they accidentally shifted the horizontal or vertical bar to a wrong position and submitted without checking. Most of them could log into the system successfully in the very next try and that is why the Total Accuracy (93.33%) is much higher than the First Accuracy (66.67%) in the second session.

Table 2 shows the average number of re-tries until the user finally logged in successfully. Even after more than two weeks, participants were able to log into the system successfully in an average of 0.64 (Median=0) re-tries, or in other words a total of 1.64 tries. 25 out of 30 (83.33%) participants were able to log into their account within three tries. For the rest, 4 participants logged in successfully within ten tries and only one participant failed to log in after trying ten times. According to the data recorded, these 5 participants failed to log in within 3 tries were all having trouble to pass only one of the three pass-images they set in the registration phase.

In summary, we conclude that the passwords of our PassMatrix are easy to memorize. Users can log into the system with only 1.64 (Median=1) authentication requests on average, and the Total Accuracy of all login trials is 93.33% even after two weeks.

TABLE 2
The mean, median and standard deviation of the number of retries in a successful attempt.

	First Session			Second Session		
	Mean	Median	S.D	Mean	Median	S.D
Practice Phase	0.41	0	0.50	-	-	-
Login Phase	0.13	0	0.35	0.64	0	2.64

6.2 Usability

We counted the number of shifts and the elapsed time per pass-image in our experiment to measure the usability of our PassMatrix in practice.

TABLE 3
The mean, median and standard deviation of total time in the registration phase

	Registration(1st)		
	Mean	Median	S.D
Total Time(s)	106.6	90.5	55.58

Table 3 shows the elapsed time that participants consumed in the registration phase. The registration took 1 minute and 46 seconds on average. Though it seems the average registration time is a bit lengthy in records, 73.33% of participants felt that the registration process is actually not time consuming and 10% of them said that they spent most of their registration time in finding pass-squares that are meaningful to them. Based on the survey data from participants, we concluded that the time required for registration is acceptable to users in practice. During registration, participants can choose 3 to 5 pass-images as their passwords. In our experiment, all but five participants chose 3 images (mean=3.2 images). The average time each user spent on practice and login (see Table 4) in the first session was 47.86 seconds and 31.31 seconds respectively. The required time to log into PassMatrix is reduced by 16.55 seconds after practicing 4 times on average to get familiar with the shifting (i.e., dragging and flinging) operations on touch screens. The results are good due to the fact that 73% of participants have either no or less than one year of experience of using smart phones (see Figure 13). Furthermore, even after more than two weeks (16.3 days on average), the average login time was still as low as 37.11 seconds, not far away from that (31.11 seconds) in the first session. The reason that the time was slightly increased was because participants needed to recall their passwords. A survey showed that the time spent in the login process is acceptable to 83.33% of participants. They felt that spending a little bit extra time is worthwhile if the authentication system can protect their passwords from being seen by others peeking over their shoulders.

For the shifting operations, while aligning a login indicator to a pass-square in each pass-image (see Table 4), there is no significant difference (F=3.6, $p > 0.05$) in the number of such operations in the practice phase and in the login phase in both sessions, where F means the F-test (http://en.wikipedia.org/wiki/F-test) and p means the p-value (http://en.wikipedia.org/wiki/P-value). Because the login indicator is randomly generated for each pass-image and elements in both the horizontal bar and vertical bar are also randomly shuffled, the number of shifting operations used to move the login indicator to the right position may differ as well. There are two types of shifting operations, which are dragging (aligning the login indicator with the pass-square in a single move) and flinging (fast finger movement on the screen; only shifting one unit at a time). As shown in the experimental results, participants only shifted 4 to 5 times per pass-image on average.

TABLE 4
The mean, median and standard deviation of total time and the number of shifts in practice/authentication phase

	Practice(1st)		**Login(1st)**		**Login(2nd)**	
	Mean	Median	Mean	Median	Mean	Median
Time(s)	47.86	41	31.31	29.5	37.11	34
shift	5.67	5	4.91	5	4.9	4

In summary, the experimental results showed that all participants can operate the login process through the PassMatrix's authentication interface. Thus, our PassMatrix is friendly to use in practice. Users may need to spend more time to log into PassMatrix in the practice phase (47.86 seconds on average) right after registering their accounts. However, they can log into the system more quickly, even two weeks after registration (in between 31.31 seconds and 37.11 seconds). The results also showed that users could easily control the horizontal and vertical bars to align login indicators with pass-squares. Hence, our PassMatrix is practical in the perspectives of easy-to-use and efficiency.

TABLE 5
Questionnaire responses. Scores are 1 to 5.

Questions	Mean	Median
Some information is exposed when authenticating in public.	4.13	4
I would have serious loss if my passwords were cracked.	4	4
PassMatrix can protect my passwords from being attacked by SSA.	4.23	4
Compared to text passwords and PIN, PassMatrix is more secure.	4.27	4
PassMatrix is secure and trustable.	4.27	4
It's difficult to find out the pass-square of others even if I had screen shots or videos of one's login process.	4.27	4
It's easy and fast to create an account in PassMatrix.	3.87	4
In general, PassMatrix is a user-friendly system and is easy to use.	4.2	4
The time consumed for using PassMatrix is acceptable.	4.07	4
The login indicator is clear and easy to memorize temporarily.	3.9	4
The login indicator is safe from being peeped at by others.	4	4
I tend to choose squares that are eye-catching.	3.83	4
I tend to choose squares that are obtrusive.	2.6	2

6.3 Questionnaires

Participants were asked to fill out questions about their personal background and a set of Likert-scale questions about their experience in using PassMatrix at the end of the first and the second sessions. We adopted five-point Likert scales, in which 1 represents a strong disagreement and 5 indicates a strong agreement.

Table 5 shows all the questions with their mean and median scores. As the result shows, participants felt it is insecure to use traditional text passwords or PIN methods, and they believed that using PassMatrix to log in can protect their passwords from being shoulder surfing attacked. For the user experience on PassMatrix, the mean scores of the series of questions are high, ranging from 3.87 to 4.20. All participants agreed that PassMatrix is easy to use and the majority of them (93.33%) considered the time spent (or in other words, complexity) for the PassMatrix's login process is acceptable. For an in-depth investigation at the number of pass-images users may accept in different authentication scenarios, see Figure 14, we found out that users tended to set only one pass-image as their password for screen lock in their mobile phones, 2 to 3 for OS user login and e-mail service login, and 5 or more for bank accounts.

We told participants to choose meaningful pass-squares to be their passwords and some of them really did. They spent more time in finding and memorizing special squares meaningful to them. However, when being asked how they decided pass-squares, most of them expressed that they chose a square that is conspicuous in an image. This might result in hot-spot problems [46] and we will discuss this issue in the next section.

In summary, the overall feedback from the participants is good. They have no trouble using PassMatrix to log in and they believe PassMatrix can protect their pass-squares from being known by others when operating in public. In addition, the required time for login is acceptable to most of them and we believe the login time will be decreased when users gain more experience in using PassMatrix.

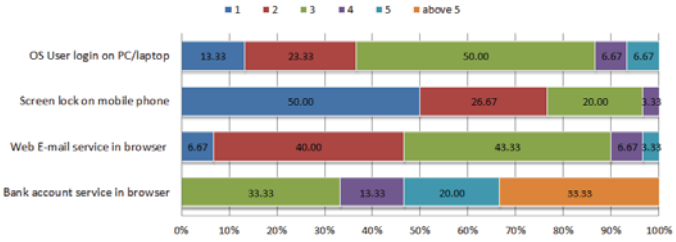

Fig. 14. Number of pass-images users may accept in different authentication scenarios.

6.4 Cracking PassMatrix

In order to brainstorm different ways, if any, to crack PassMatrix through shoulder surfing attacks, we aggregated the pass-images and pass-squares chosen by participants during the registration phase and randomly picked 10 to-be-cracked accounts, each with three pass-images. 30 participants were divided into 10 groups, each with 3 participants. Each group was asked to crack one account. We simulated the owners of these 10 accounts to log into PassMatrix two times and screenshot the entire login processes. These two login records of each account along with an answer sheet were given to three participants. They were asked to crack (identify) the pass-square in each pass-image from the given two login records. 25 participants returned their answer sheets after two weeks.

Feedback from participants showed that no one was able to get any clue from the two login records. For each pass-image, they couldn't figure out the location of the pass-square since they didn't know the login indicator of the image. The given two login records didn't make it easier to crack the pass-squares because the operations on different pass-images within a login trial are independent, and the operations on the same pass-image in two different login trials are also independent. Most of them said that they just filled out the answer sheet by random guessing or hot-spot guessing. Each participant had ten chances to guess the pass-squares in the given three pass-images, thus we received 250 guessing cases. The participant was considered successful if he got any answer case (out of 10 guessing cases) correct on all three pass-squares. As the result showed in Table 6, none of these 250 guessing cases were correct on all three pass-squares, 1 case had correct guesses on two pass-squares, 23 cases guessed right on one pass-square, and all other 226 cases totally failed to guess any.

TABLE 6
Result of cracking PassMatrix.

		Number of case (250 in total)
Success	(3 correct)	0
Fail	(2 correct)	1
	(1 correct)	23
	(0 correct)	226

We further investigated the experimental data and found out that there were some more likely selected squares in each image probably because they are more memorable, obvious or unique. Therefore, some pass-images were relatively easier to guess due to the fact of lacking abundant objects in that image. This hot-spot problem [46], [47] is common for most graphical-based authentication schemes. We chose three representative pass-images from our image list to demonstrate the hot-spot problem. Two of these three images are relatively more complex and contain rich information, whereas the other one is an image of scenery with a simple vision. 6 out of 25 returned password cracking answer sheets were actually guessing the pass-squares on these three images. With 10 guessing cases per answer sheet, there were a total of 60 guessing cases. Figure 15 shows the password guessing statistics, where the number within a square represents the number of times the square were selected as part of a guessed password. The squares selected as possible passwords in the first two images are relatively more uniformly distributed over 38.96% and 44.16% of the total area respectively. However, in the third image, the squares selected are concentrated in only 28.57% area of the image. The squares with red frames are the real pass-squares set by participants in the registration phase. Some of these real pass-squares were also selected by the majority of password cracking participants as their guessed passwords. This experiment showed that an attacker has a higher probability (than a random guess) to break into an account by conducting hot-spot guessing attacks.

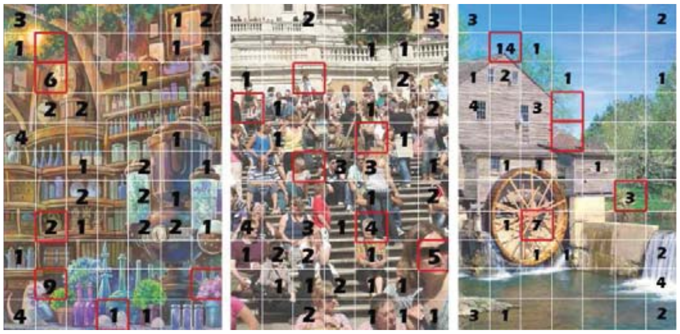

Fig. 15. Possible pass-squares chosen by participants. A large number in a square means the square is a hot spot.

7 Security Analysis

In this section we evaluate the security of the proposed authentication system against three types of attacks: random guess attack, shoulder surfing attack, and smudge attack.

7.1 Random Guess Attack

To perform a random guess attack, the attacker randomly tries each square as a possible pass-square for each pass-image until a successful login occurs. The key security determinants of the system are the number of pass-images and the degree of discretization of each image. To quantify the security of PassMatrix against random guess attacks, we define the entropy of a password space as in equation 3. Table 7 defines the notations used in the equation. If the entropy of a password space is k bits, there will be 2^k possible passwords in that space.

$$Entropy = \log_2((D_x \times D_y)^i)^n \tag{3}$$

TABLE 7
The definition of notations used in equation 3.

Notation	Definition
D_x	The number of partitions in x-direction
D_y	The number of partitions in y-direction
i=1	Obtain login indicators by touching the screen with hand grasped
i=2	Obtain login indicators by predefined images
n	The number of pass-images set by user

TABLE 8
The entropy (bits) of PassMatrix against random guess attacks and corresponding entropy of text passwords and PIN passwords, varied from 1 to 5 pass-images (or 1-5 click-point(s)).

n: Number of images			1	2	3	4	5
PassMatrix	7×11	type 1	6.27	12.53	18.80	25.07	31.33
		type 2	12.53	25.07	37.60	50.13	62.67
	32×20	type 1	9.32	18.64	27.97	37.29	46.61
		type 2	18.64	37.29	55.93	74.58	93.22
Text Passwords			6.57	13.14	19.71	26.28	32.85
PIN			3.32	6.64	9.97	13.29	16.61

In our implementation, we built a prototype of PassMatrix on Android OS, and conducted the user study on Samsung Nexus S, which is a device with a small screen. Due to the size limitation, the tolerance squares are 60×60 pixels, so that users can distinguish squares clearly. Since the size of images is 420×660 pixels, we have 7 partitions on the x-axis and 11 partitions on the y-axis, which means we have a total of 7×11 squares in each image.

Table 8 shows the entropies of PassMatrix type 1 (getting login indicators by touching the screen with hand grasped) and type 2 (getting login indicators through predefined images) from one pass-image to five pass-images (1-5 click points), as well as the corresponding entropies of text passwords and PIN passwords. We can see that the entropies of type 1 are larger than those of PIN passwords. With 3 pass-images the entropy of our proposed system is 18.8 bits, which is larger than the entropy 18.57 bits of the Android pattern screen lock [48], currently the primary authentication mechanism of Android devices. For the type 2 of our proposed system, i.e., obtaining login indicators through graph, random guess attackers require more effort to crack the system because they need to discover not only the correct pass-square but also the correct login indicator in each grid. Even if attackers know the pass-square of a pass-image, they still couldn't break into the account without the correct login indicator of that trial. Therefore, the entropies of the PassMatrix type 2 systems are significantly larger than those of PIN passwords and those of the Android pattern screen lock with just two pass-images.

Furthermore, in some applications such as the OS user login or web applications on PCs/laptops with big screens, where images with high resolution can be displayed entirely on the screen without scroll bars, we can set a high discretization degree for images to increase the security strength of PassMatrix. For instance, with the size of 40×40 pixels for each square, a 1280×800 image is divided into 32 and 20 partitions on the x-axis and on the y-axis respectively, so that the entropy can be increased to $\log_2(32 \times 20)^n$.

7.2 Shoulder Surfing Attack

Due to the fact that shoulder surfing has been a real threat to authentication systems with either textual or graphical passwords, many novel authentication schemes were proposed to protect systems from this attack. Unfortunately, most of them were unsuccessful to alleviate the threat if the shoulder-surfing attack is camera-based. For instance, some schemes such as PIN-entry method [34] and spy-resistant keyboard [19] were designed based on the difficulties of short-term memory. Camera-based shoulder surfing attacks can easily crack the passwords of these schemes. The password spaces of other schemes such as those in CAPTCHA-based method [24], Pass-icons [18] and Color-rings [25] can be narrowed down by camera-based shoulder surfing attacks.

The proposed authentication system PassMatrix takes full advantage of adding extra information to obfuscate the login process, using an approach to point out the locations of pass-squares implicitly instead of typing or clicking on password objects directly. Since the horizontal and vertical bars are circulative and thus cover the entire area of the image, the password space will not be narrowed down even if the whole authentication process is recorded by attackers. Furthermore, the login indicator for each pass-image varies so that each pass-image is an independent case. Thus, no pattern can be extracted from a set of pass-images in an authentication trial, neither from multiple login processes. With the above security features, PassMatrix should be strong enough to resist shoulder surfing attacks, even if the attacks are camera-equipped.

7.3 Smudge Attack

A smudge attack [39] is an implicit attack where attackers attempt to extract sensitive information from recent users' input by inspecting smudges left on touch screens. Since both the horizontal and vertical bars in PassMatrix are scrollable, shifting on any element within the bar can circulate the whole bar. Thus, users do not have to shift the bars by touching the login indicators. The smudge left by users may be quite fixed, but it only indicates the habitual stretching range of the thumb or finger. The length of the smudge left on the screen also provides no useful information since the login indicator is generated randomly for each pass-image and the permutations of elements on both bars are also randomly re-arranged in each pass-image and in each login session. Therefore, the proposed PassMatrix is immune from smudge attacks.

8 Conclusion

8.1 Discussion

Although we discretized each image into a grid of 7×11, which holds 77 squares per image, we still think it is not secure enough to resist against brute force attacks or random guessing attacks. According to the result of questionnaires in the user study, users expressed that they prefer to use only 1 to 2 pass-images for screen lock after considering the trade-off between security and usability. This means that the entropies of PassMatrix in screen lock will be 6.27 bits and 12.53 bits for one pass-image and two pass-images respectively, and can be strengthened to 12.53 and 25.07 bits when obtaining login indicators through the graphical method. In order to be more secure than the existing Android pattern password with entropy 18.57 bits against brute force attacks, users have to set two pass-images and use the graphical method to obtain the one-time login indicators.

Like most of other graphical password authentication systems, PassMatrix is vulnerable to random guess attacks based on hot-spot analyzing. This weakness can be improved by letting users upload their own images and therefore make it more difficult for attackers to collect statistics of hot-spots. Moreover, because images with less independent objects usually suffer more on the hot-spot problem, carefully selecting images with rich objects can alleviate the hot-spot based random guess attacks.

Here we summary the features of PassMatrix. Compared with DAS [6], PassPoints [7] and Marcos's finger-drawn doodles [33], PassMatrix is strong enough to resist shoulder surfing attacks, even if the attacks are camera-equipped. And the PIN-entry method [34], FakePointer [35], Wiedenback's scheme [36] and Color Rings [25] do not have enough computational complexity while facing to random guess attack, which PassMatrix can withstand.

8.2 Conclusion

With the increasing trend of web services and apps, users are able to access these applications anytime and anywhere with various devices. In order to protect users' digital property, authentication is required every time they try to access their personal account and data. However, conducting the authentication process in public might result in potential shoulder surfing attacks. Even a complicated password can be cracked easily through shoulder surfing. Using traditional textual passwords or PIN method, users need to type their passwords to authenticate themselves and thus these passwords can be revealed easily if someone peeks over shoulder or uses video recording devices such as cell phones.

To overcome this problem, we proposed a shoulder-surfing resistant authentication system based on graphical passwords, named PassMatrix. Using a one-time login indicator per image, users can point out the location of their pass-square without directly clicking or touching it, which is an action vulnerable to shoulder surfing attacks. Because of the design of the horizontal and vertical bars that cover the entire pass-image, it offers no clue for attackers to narrow down the password space even if they have more than one login records of that account. Furthermore, we implemented a PassMatrix prototype on Android and carried out user experiments to evaluate the memorability and usability. The experimental result showed that users can log into the system with an average of 1.64 tries (Median=1), and the Total Accuracy of all login trials is 93.33% even two weeks after registration. The total time consumed to log into PassMatrix with an average of 3.2 pass-images is between 31.31 and 37.11 seconds and is considered acceptable by 83.33% of participants in our user study.

Based on the experimental results and survey data, PassMatrix is a novel and easy-to-use graphical password authentication system, which can effectively alleviate shoulder-surfing attacks. In addition, PassMatrix can be applied to any authentication scenario and device with simple input and output capabilities. The survey data in the user study also showed that PassMatrix is practical in the real world.

References

[1] S. Sood, A. Sarje, and K. Singh, "Cryptanalysis of password authentication schemes: Current status and key issues," in *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, Dec 2009, pp. 1–7.

[2] S. Gurav, L. Gawade, P. Rane, and N. Khochare, "Graphical password authentication: Cloud securing scheme," in *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*, Jan 2014, pp. 479–483.

[3] K. Gilhooly, "Biometrics: Getting back to business," *Computerworld, May*, vol. 9, 2005.

[4] R. Dhamija and A. Perrig, "Deja vu: A user study using images for authentication," in *Proceedings of the 9th conference on USENIX Security Symposium-Volume 9*. USENIX Association, 2000, pp. 4–4.

[5] "Realuser," http://www.realuser.com/.

[6] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The design and analysis of graphical passwords," in *Proceedings of the 8th conference on USENIX Security Symposium-Volume 8*. USENIX Association, 1999, pp. 1–1.

[7] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 102–127, 2005.

[8] A. Paivio, T. Rogers, and P. Smythe, "Why are pictures easier to recall than words?" *Psychonomic Science*, 1968.

[9] D. Nelson, U. Reed, and J. Walling, "Picture superiority effect," *Journal of Experimental Psychology: Human Learning and Memory*, vol. 3, pp. 485–497, 1977.

[10] S. Brostoff and M. Sasse, "Are passfaces more usable than passwords? a field trial investigation," *PEOPLE AND COMPUTERS*, pp. 405–424, 2000.

[11] A. De Angeli, M. Coutts, L. Coventry, G. Johnson, D. Cameron, and M. Fischer, "Vip: a visual approach to user authentication," in *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, 2002, pp. 316–323.

[12] B. Ives, K. Walsh, and H. Schneider, "The domino effect of password reuse," *Communications of the ACM*, vol. 47, no. 4, pp. 75–78, 2004.

[13] J. Long and K. Mitnick, *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*. Elsevier Science, 2011.

[14] T. Kwon, S. Shin, and S. Na, "Covert attentional shoulder surfing: Human adversaries are more powerful than expected," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 716–727, June 2014.

[15] "Google glass snoopers can steal your passcode with a glance," http://www.wired.com/2014/06/google-glass-snoopers-can-steal-your-passcode-with-a-glance/.

[16] M. Sasse, S. Brostoff, and D. Weirich, "Transforming the weakest linka human/computer interaction approach to usable and effective security," *BT technology journal*, vol. 19, no. 3, pp. 122–131, 2001.

[17] "Mobile marketing statistics compilation," http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/.

[18] D. Hong, S. Man, B. Hawes, and M. Mathews, "A password scheme strongly resistant to spyware," in *Proceedings of International conference on security and management*, 2004.

[19] D. Tan, P. Keyani, and M. Czerwinski, "Spy-resistant keyboard: Towards more secure password entry on publicly observable touch screens," in *Proceedings of OZCHI-Computer-Human Interaction Special Interest Group (CHISIG) of Australia. Canberra, Australia: ACM Press*. Citeseer, 2005.

[20] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 2007, pp. 13–19.

[21] H. Zhao and X. Li, "S3pas: A scalable shoulder-surfing resistant textual-graphical password authentication scheme," in *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, vol. 2. IEEE, 2007, pp. 467–472.

[22] X. Bai, W. Gu, S. Chellappan, X. Wang, D. Xuan, and B. Ma, "Pas: predicate-based authentication services against powerful passive adversaries," in *2008 Annual Computer Security Applications Conference*. IEEE, 2008, pp. 433–442.

[23] Z. Zheng, X. Liu, L. Yin, and Z. Liu, "A stroke-based textual password authentication scheme," in *Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on*, vol. 3. IEEE, 2009, pp. 90–95.

[24] L. Wang, X. Chang, Z. Ren, H. Gao, X. Liu, and U. Aickelin, "Against spyware using captcha in graphical password scheme," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2010, pp. 760–767.

[25] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. Nicholson, J. Nicholson, and P. Olivier, "Multi-touch authentication on tabletops," in *Proceedings of the 28th international conference on Human factors in computing systems*. ACM, 2010, pp. 1093–1102.

[26] "Black hat: Google glass can steal your passcodes," https://www.technologyreview.com/s/529896/black-hat-google-glass-can-steal-your-passcodes/.

[27] A. De Luca, M. Harbach, E. von Zezschwitz, M.-E. Maurer, B. E. Slawik, H. Hussmann, and M. Smith, "Now you see me, now you don't: Protecting smartphone authentication from shoulder surfers," in *Proceedings of the 32Nd Annual ACM Conference on*

Human Factors in Computing Systems, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 2937–2946.

[28] E. von Zezschwitz, A. De Luca, and H. Hussmann, "Honey, i shrunk the keys: Influences of mobile devices on password composition and authentication performance," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, ser. NordiCHI '14. New York, NY, USA: ACM, 2014, pp. 461–470.

[29] A. Bianchi, I. Oakley, V. Kostakos, and D. S. Kwon, "The phone lock: Audio and haptic shoulder-surfing resistant pin entry methods for mobile devices," in *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '11. New York, NY, USA: ACM, 2011, pp. 197–200.

[30] A. Bianchi, I. Oakley, and D. S. Kwon, "The secure haptic keypad: A tactile password system," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1089–1092.

[31] I. Oakley and A. Bianchi, "Multi-touch passwords for mobile device access," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ser. UbiComp '12. New York, NY, USA: ACM, 2012, pp. 611–612.

[32] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "The doodb graphical password database: Data analysis and benchmark results," *Access, IEEE*, vol. 1, pp. 596–605, 2013.

[33] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "Graphical password-based user authentication with free-form doodles," *IEEE Transactions on Human-Machine Systems*, vol. PP, no. 99, pp. 1–8, 2015.

[34] V. Roth, K. Richter, and R. Freidinger, "A pin-entry method resilient against shoulder surfing," in *Proceedings of the 11th ACM conference on Computer and communications security*, ser. CCS '04. New York, NY, USA: ACM, 2004, pp. 236–245.

[35] T. Takada, "fakepointer: An authentication scheme for improving security against peeping attacks using video cameras," in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on*. IEEE, 2008, pp. 395–400.

[36] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of the working conference on Advanced visual interfaces*, ser. AVI '06. New York, NY, USA: ACM, 2006, pp. 177–184.

[37] B. Laxton, K. Wang, and S. Savage, "Reconsidering physical key secrecy: Teleduplication via optical decoding," in *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 2008, pp. 469–478.

[38] X. Suo, Y. Zhu, and G. Owen, "Analysis and design of graphical password techniques," *Advances in Visual Computing*, pp. 741–749, 2006.

[39] A. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. Smith, "Smudge attacks on smartphone touch screens," in *USENIX 4th Workshop on Offensive Technologies*, 2010.

[40] S. Chiasson, P. van Oorschot, and R. Biddle, "Graphical password authentication using cued click points," *Computer Security–ESORICS 2007*, pp. 359–374, 2007.

[41] "Secure socket layer ssl," http://en.wikipedia.org/wiki/Transport_Layer_Security.

[42] L. Cranor and S. Garfinkel, *Security and Usability*. O'Reilly Media, Inc., 2005.

[43] "Google play," https://play.google.com/store/.

[44] "Android developer," http://developer.android.com/index.html.

[45] "Android version of distribution," http://developer.android.com/resources/dashboard/platform-versions.html.

[46] J. Thorpe and P. van Oorschot, "Human-seeded attacks and exploiting hot-spots in graphical passwords," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*. USENIX Association, 2007, p. 8.

[47] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "Authentication using graphical passwords: effects of tolerance and image choice," in *Proceedings of the 2005 symposium on Usable privacy and security*. ACM, 2005, pp. 1–12.

[48] "Android 2.2 platform highlights," http://developer.android.com/sdk/android-2.2-highlights.html,.

Hung-Min Sun received the B.S. and M.S. degrees in applied mathematics from National Chung-Hsing University, in 1988 and 1990, respectively, and the Ph.D. degree in computer science and information engineering fromNational Chiao-Tung University, in 1995.

He was an associate professor with the Department of Information Management, Chaoyang University of Technology, from 1995 to 1999, the Department of Computer Science and Information Engineering, National Cheng-Kung University, from 2000 to 2002, and the Department of Computer Science, National Cheng-Kung University, from 2002 to 2008. Currently, he is working as a full professor with the Department of Computer Science, National Tsing Hua University. He has published more than 150 international journal and conference papers. He was the program cochair of 2001 National Information Security Conference, and the program committee members of many international conferences. He was the honor chairs of 2009 International Conference on Computer and Automation Engineering, 2009 International Conference on Computer Research and Development, and 2009 International Conference on Telecom Technology and Applications. He serves as the editor-in-chief of the International Journal of Digital Content Technology and its Applications, and the editor members of many international journals including ISRN Communications and Networking, and International Journal of Security, Advances in Information Sciences and Service Sciences: an International Journal of Research and Innovation, International Journal of Intelligent Information Processing, and Journal of Next Generation Information Technology. He won many best paper awards in academic journal and conferences, including the annual best paper award from the Journal of Information Science and Engineering in 2003, the best paper award in MobiSys09, NSC05, NISC06, NISC07, CISC09, and ICS2010. He won the Y. Z. Hsu Scientific Paper Award, Far Eastern Y. Z. Hsu Science and Technology Memorial Foundation, 2010. His research interests include network security, cryptography, and wireless networks.

Chia-Yun Cheng received her B.S. degree in Department of Information Management from National Sun Yat-sen University in 2010, her M.S. degree in Department of Information System and Application from National Tsing Hua University in 2012. Her research interests include wireless network security, multimedia security and applied cryptography.

Jyh-haw Yeh received a B.A. degree in Applied Mathematics from National Chung-Hsing University, Taiwan in 1984, a M.S. degree in Computer Information Science from Cleveland State University, Cleveland, Ohio in 1993, and a Ph.D degree in Computer and Information Science and Engineering from University of Florida in 1999. He joined the department of Computer Science at Boise State University in 2000 and was awarded tenure at 2006. He served as an editor, track chair or PC member in an array of international journals and conference proceedings. He also served as a Panelist of 2008 NSF Cyber Trust Program. His current research interests are in the area of computer security, cloud security and cryptography.

Shiuan-Tung Chen received his B.S. degree in Department of Applied Mathematics from National Chung Hsing University in 2004, his M.S. degree in Department of Information System and Application from National Tsing Hua University in 2007. He is a Ph.D. student in Department of Computer Science of National Tsing Hua University from 2007 to now, and has already been a Ph.D. candidate. His research interests include forward error correction codes, reliable data transmission and applied cryptography.