uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Hai Tao

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Pass-Go, a New Graphical Password Scheme

TITRE DE LA THÈSE / TITLE OF THESIS

Carlisle Adams

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

A. Miri

A. Matrawy

P. Van Oorschot

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

# Pass-Go, a New Graphical Password Scheme

**HAI TAO**

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the Master of Applied Science degree in Electrical and Computer Engineering

University of Ottawa

*Your file Votre référence*
*ISBN: 978-0-494-18470-7*
*Our file Notre référence*
*ISBN: 978-0-494-18470-7*

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| ASCII | American Standard Code for Information Interchange |
| BMP | Bitmap |
| DAS | Draw-A-Secret |
| EKE | Encryption Key Exchange |
| FAQ | Frequently Asked Question(s) |
| FTP | File Transfer Protocol |
| GIF | Graphics Interchange Format |
| IDE | Integrated Development Environment |
| JSP | JavaServer Pages |
| JPEG | Joint Photographic Experts Group |
| MD5 | Message-Digest algorithm 5 |
| PDA | Personal Digital Assistant |
| PNG | Portable Network Graphics |
| PKI | Public Key Infrastructure |

# List of Symbols

| | |
|---|---|
| $L_{max}$ | Maximum password length (i.e., passwords longer than $L_{max}$ have a probability of zero to be chosen) |
| $N_m$ | The number of the corresponding alphanumeric or well-known symbols |
| $N_r$ | The number of rectangles in which an alphanumeric or well-known symbol can be held |
| $N(l,G)$ | The number of strokes of length equal to $l$ in a $G \times G$ grid, as defined in each scheme |
| $N_{(x,y)}$ | The neighbors of a cell or an intersection $(x,y)$ |
| $n(x,y,l,G)$ | The number of strokes of length $l$ ending at the intersection or the cell $(x,y)$ in a $G \times G$ grid, as defined in each scheme |
| $N_w$ | The number of ways in which the symbols could be drawn |
| $S_{1a}$ | The subset of passwords whose components are symmetric about the center 3 horizontal and/or vertical axes, and are drawn in a "symmetric manner" |
| $S_{1b}$ | The subset of passwords whose components are symmetric about the center horizontal and/or vertical axes only, and are drawn in a "symmetric manner" |
| $S_2$ | The subset of passwords with a maximum stroke-count of 4 |
| $S_{1b} \cap S_2$ | The intersection of $S_{1b}$ and $S_2$ |

## Abstract

Inspired by an old Chinese game, Go, we have designed a new graphical password scheme, Pass-Go, in which a user selects intersections on a grid as a way to input a password. While offering an extremely large password space (256 bits for the most basic scheme), our scheme provides acceptable usability, as empirically demonstrated by, to the best of our knowledge, the largest user study (167 subjects involved) on graphical passwords, conducted in the fall semester of 2005 in two university classes.

Our scheme supports most application environments and input devices, rather than being limited to small mobile devices (PDAs), and can be used to derive cryptographic keys. We study the memorable password space and show the potential power of this scheme by exploring further improvements and variation mechanisms. We believe that Pass-Go could be a solid platform for future research and study.



## Key Words

Password, Pass-Go, memorable password space, dictionary attack.

# Acknowledgments

I would like to thank my supervisor, Dr. Carlisle Adams, under whose supervision I could complete this thesis. I appreciate the valuable comments, advice and suggestions he gave me. Besides the knowledge, I also learned from him the attitude and manner to conduct research, which I appreciate the most.

My appreciation also goes to Julie Thorpe for providing reference data and Paul Van Oorschot, in whose class I learned the topic of graphical password and conceived the idea of Pass-Go. Many thanks as well to Guy-Vincent Jourdan for his assistance in conducting the user study.

This thesis is dedicated to my daughter, Qiner.

Hai Tao

June 12, 2006

# Chapter 1 Introduction

## *1.1 Problem statement*

Conventional textual passwords use a string of alphanumeric characters (or printable ASCII characters) to identify a user. However, it is well known that textual passwords are vulnerable to small dictionary attack [Feldmeier and Karn 1989; Morris and Thompson 1979; Wu 1990], in which an attacker exhaustively searches candidate passwords from a "small dictionary". Due to the limitation of human memory, users frequently choose those passwords which are easy to remember, causing a significant fraction of user-chosen passwords to fall into this small dictionary. A well-designed dictionary is a tiny subset of the full password space, which is further prioritized according to the probabilities of the likelihood for a password to be chosen [Openwall Project 2004a; 2004b].

This "small dictionary" attack is so successful that in Klein's case study [Klein 1990], about 25% of 14,000 passwords were cracked by a dictionary with only 3 million entries (the size of the dictionary is 21.5 bits). Following the same method used in [Van Oorschot and Thorpe 2005], such a dictionary can be exhausted by a 3.2GHz Pentium™4 machine in only 0.22 second. Therefore, it is widely believed that the security of a password scheme is related more closely to the size of its memorable password space, rather than that of its full password space. This weakness also renders deriving a safe cryptographic key from a user-chosen password extremely hard, thus causing some strong and complicated protocols such as PKI and EKE to be required in order to secure network communications.

Graphical passwords, which require a user to remember and repeat visual information, have been proposed to offer better resistance to dictionary attack. Psychological studies

support the hypothesis that humans have a significant capability to recognize and to recall visual images [Paivio et al. 1968; Bower et al. 1975; Standing 1973]. If users are able to remember more complex graphical passwords (i.e., from a larger password space), an attacker has to build a bigger dictionary, thus spend more time or deploy more computational power to achieve the same success as for textual passwords.

In 1999, Jermyn et al. [Jermyn et al. 1999] suggested a graphical password scheme called DAS (draw-a-secret), which requires a user to draw a secret design on a grid as a way to input a password. Surprisingly, they found that DAS could offer very large password space for reasonable parameters. On a 5×5 grid, the total number of passwords of length 12 or less ($L_{max}$=12 and passwords longer than $L_{max}$ are considered to have a probability of zero to be chosen) is $2.3\times10^{17}$, larger than that of textual passwords composed of 8 printable ASCII characters ($95^8 = 6.6\times10^{15}$).

Thorpe and Van Oorschot [Thorpe and Van Oorschot 2004a; 2004b; Van Oorschot and Thorpe 2005] studied the memorable password space of DAS and introduced the concept of a symmetric graphical dictionary, based on psychological theories that people prefer images that exhibit (especially mirror) symmetric patterns. They classified symmetric passwords into several subclasses according to the axes considered. The size of the smallest subclass $S_{1b}$ (the subset of passwords whose components are symmetric about the center vertical and/or horizontal axes only and drawn in a "symmetric" manner) was quantified to be 43 bits (for $L_{max}$=12 on a 5×5 grid). The size of such a graphical dictionary can be further reduced by restricting stroke-count (the number of composite strokes in a DAS password) to a maximum of 4, assuming DAS passwords with more strokes are complicated to remember and thus less likely to be chosen. The cardinality of such a reduced dictionary is only 31 bits, significantly smaller than that of the full password space (58 bits), and can be effectively exhausted by a fast computer. Then the remaining question is this: how many user-chosen passwords can be captured by their graphical dictionaries? In other words, how successful would such attacks be?

## 1.2 Thesis contribution

We propose a new grid-based graphical password scheme, Pass-Go, in which users select intersections on a grid to authenticate a system. Our scheme was inspired by an old Chinese game, Go (see Figure 1), which is famous with its inexhaustible variety and simple rules. Having developed in China between 3,000 and 4,000 years ago, Go (called Wei Chi in China and Baduk in Korea) is played by more than 100 million people around world [Usgo 2006a] today (particularly in eastern Asian countries).

Pass-Go can be considered as an improvement of DAS, as it keeps most of the advantages of DAS and achieves stronger security and better usability. These improvements are believed to arise from our innovative design, as elaborated in §3.2.

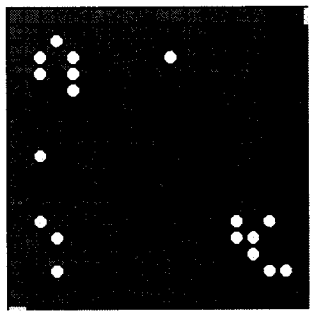


**Figure 1 Go game**

We conducted an informal user study in two university classes in the fall semester of 2005. A simple teaching management system was developed on an Internet website, through which students could access their grades and study materials by logging in with Pass-Go passwords. Over a three month period, the system was accessed successfully 5291 times by 167 subjects (32 times per user on average).

The overall result of our user study is positive. First, the remaining question (mentioned above) on DAS is partially answered: a significant portion of users choose mirror symmetric passwords (40% fall into $S_{1b}$). On the other hand, the Pass-Go

passwords chosen by our users are much longer and contain more strokes and dots (stroke of length 1) than previously conjectured for DAS. By excluding one extreme case (76), the password length (the number of intersections selected) ranges from 8 to 41, with an average of 16.88. For our calculations, then, we consider 40 as a reasonable value for $L_{max}$ in Pass-Go; this results in an extremely large full password space of 256 bits (374 bits if color is considered). Even if we reduce $L_{max}$ to a much smaller value, such as 16, for the multi-account attack model (an attacker targets any password out of multiple accounts), the size of the full password space is still 102 bits (150 bits with color), large enough to derive a secure cryptographic key.

We subsequently applied Thorpe and Van Oorschot's methods to measure the resistance of our scheme to graphical dictionary attacks in a conservative manner. By restricting the stroke-count to a maximum of 4, the size of $S_{1b}$ can only be reduced to 43.4 bits (a major improvement over 31 bits, which was the case for DAS). Moreover, such an attack can at most capture a small fraction (15%) of passwords, implying that our scheme offers substantially stronger resistance to such off-line dictionary attacks than the DAS scheme.

## *1.3 Organization of thesis*

This thesis is organized as follows: we review prior work in chapter 2; in chapter 3, we discuss the design of Pass-Go; we describe our user study and analyze the results in chapter 4; in chapter 5, we study and quantify the memorable password space; in chapter 6, we explore variations on the basic Pass-Go scheme; in chapter 7, we draw some conclusions and discuss future work.

# Chapter 2 Background

According to different types of graphical backgrounds used, we divide graphical password schemes into two major categories: image-based schemes and grid-based schemes.

## *2.1 Image-based schemes*

As the name implies, image-based schemes use images, including photo graphics, artificial pictures, or other kind of images as background. Based on the number of images displayed, we further divide image-based schemes into two subclasses: single-image schemes and multiple-image schemes. The latter is also called a recognition based scheme in some literature.

### 2.1.1 Single-image schemes

Single-image based schemes use one single image as a background, and require a user to repeat several actions with an input device, such as clicking or dragging, in the same manner as in the registration stage.

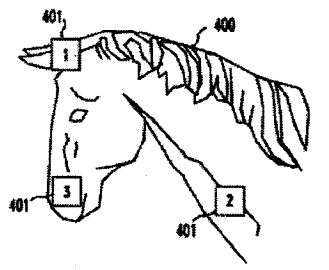



**Figure 2 Graphical password scheme suggested by Blonder [Blonder 1996]**

Blonder [Blonder 1996] gave the initial idea of graphical password. In his scheme, a user is presented with one predetermined image on a visual display and required to select one or more predetermined positions ('tap regions", see Figure 2) on the displayed image in a particular order to access the restricted resource. The major drawback of this scheme is that users cannot click arbitrarily on the background. The memorable password space was not studied by the author either.


**Figure 3 VisKey [Sfr 2006]**

VisKey [Sfr 2006] is a commercial software product released by SFR, a German company. VisKey was designed for small mobile devices. In this scheme, pictures in common formats (e.g., JPEG, GIF, PNG, and BMP) are stored in the device storage and can be selected by a user as the graphical password background in the password initialization phase. The graphical password consists of the defined spots and their order (see Figure 3). To enter a password correctly, a user has to touch the same spots in the same order as in the registration stage. Because it is difficult (and usually not possible) to touch the exact points, visKey allows all input within a certain tolerance area around it. The size of this area can be predefined by users. This input precision needs to be set carefully, as it will directly influence the security and the usability of the graphical password. If the input precision is too big, the visKey will be relatively easy to crack since there are fewer possible password combinations. If the input precision is too small,

users might have difficulty to enter a password correctly. For a reasonable setting of parameters, a four-spot visKey can offer theoretically almost 1 billion (approximately 30 bits) possibilities to define a password, comparable to the password space of a textual password composed of five alphanumeric characters ($\log_2 62^5 \approx 30$). Such a password space, however, is not large enough to resist off-line attacks by a fast computer. Therefore, in order to defend against brute-force attacks, more spots (e.g., 8) need to be defined for a password.

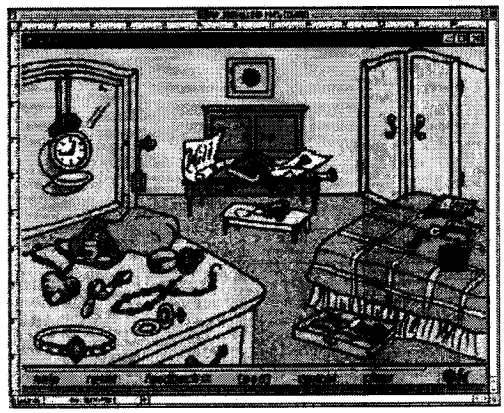


**Figure 4 V-GO [Passlogix 2006]**

V-GO [Passlogix 2006] is also a commercial security solution provided by Passlogix Inc.. V-GO allows a user to create a graphical password by navigating through an image, as shown in Figure 4 . To enter a password, a user can click and/or drag on a series of items within that image. In a kitchen environment, for example, a user can take food from a fridge and put it in the oven, dip the vegetables in water, or set a timer on the clock. "The Cocktail Lounge" allows users to "mix a martini" to log into a system. Other similar settings include dialing a phone number, preparing a meal by selecting and cooking ingredients, setting the date and time on a clock, making a stock trade, or choosing a hand of cards. One major drawback of V-GO is that the full password space is small. For example, there are only a few places that one can take food from and put into, thus causing the V-GO passwords to be exhaustible by a fast computer. In addition, a user-chosen password might be easily guessable. For example, a hand of cards chosen by a

user might be quite predictable (e.g. in the same color, or in a row), as remembering a hand of random cards is a challenge for most users' memory. The size of the memorable password space of V-GO is therefore small.

**Figure 5 Passpoints [Wiedenbeck et al. 2005]**

Birget et al. [Birget et al. 2003] released the restriction associated with Blonder-style scheme (mentioned above) and allowed a user to click on any point inside a background image. With a multi-grid method, which they call "robust discretization", as long as the user clicks within a predetermined tolerance distance of the originally chosen point, the clicking will be encoded as same as that for the original one. This allows the password to be stored as the result of a hash function. However, the information about the safe grid (one out of three grids referred for each click) cannot be hashed; this might leak information once obtained by an attacker. Also in this scheme, users can choose any image as background, including their own. However, in such a case, the login process has to begin with an extra process, in which a bidirectional communication is needed to submit a user id to the server and to transmit the corresponding image back to the user after making a search in its database.

Wiedenbeck et al. conducted a user study on the same scheme, with their implementation (Passpoints) [Wiedenbeck et al. 2005]. Forty members of a university

community, including staff, students, and faculty, participated and were asked to input Passpoints passwords (see Figure 5) and alphanumeric passwords respectively. Three phases were monitored and studied: creation phase, learning phase, and retention phase. Their results show that it is easier to create Passpoints passwords than alphanumeric passwords. It took only 64 seconds on average to create a new Passpoints password, and only one out of twenty participants had to make two attempts to successfully create a Passpoints password. Alphanumeric passwords, on the other hand, were shown to be harder to create, as it took on average 81 seconds to create an alphanumeric password successfully and more failed attempts were observed. In the learning phase, for Passpoints it took more trials to achieve 10 correct password inputs than did the alphanumeric passwords. The study also observed that graphical passwords took longer to input, compared to alphanumeric passwords, but the difference was not due mainly to the mechanics of movement and selection, but to the think time to locate the correct click region and determine precisely where to click.

The common problems of single-image schemes include:

a) the background image has to be intricate and rich enough that many memorable points are available. Such images are difficult to compress effectively because of the low content redundancy, therefore more storage and network bandwidth resources are required for the bulky image files;
b) it is difficult to input a password through a keyboard, the most common input device; if the mouse doesn't function well or a light pen is not available, the system cannot work properly;
c) looking for small spots in a rich picture might be tiresome and unpleasant for users with weak vision.

### 2.1.2 Multiple-image schemes

In multiple-image schemes, on the other hand, multiple images are presented and a user is required to recognize and identify one or more of them, which are previously seen and selected by the user.

Psychological studies suggest that people are much better at imprecise recall, particularly in recognition of previously experienced stimuli [Intraub 1980]. This class of passwords was shown to be remembered by users for a long period after short perception [Dhamija and Perrig 2000]. This section discusses a number of proposed schemes in this category, and some simple improvements to multiple-image schemes are suggested in Appendix B.

**Figure 6 Passfaces™ [Passfaces 2006]**

Passfaces™, a commercial product by Passfaces Corporation, requires a user to select previously seen human face pictures as a password [Passfaces 2006], as shown in Figure 6. One problem with Passfaces™ is that some faces displayed might not be welcomed by certain users. In other words, if a user has to look at some faces he/she does not like or even dislike, the login process will become unpleasant. Another drawback of Passfaces™ is that it cannot be used by people who are face-blind (a disease which affects a person's ability to tell faces apart).

[Brostoff and Sasse 2000] conducted a user study (34 subjects involved) on this scheme and their result suggests that Passfaces™ is easier to remember than textual passwords. [Davis et al. 2004] suggested a similar scheme, the story scheme, in which a user's password is a sequence of $k$ images selected by the user to make a story, as shown in Figure 7.

**Figure 7 Story scheme [Davis et al. 2004]**

Davis et al. empirically studied and compared these two schemes by surveying 154 computer engineering and computer science students from two universities. Their result shows that in Passfaces™ the user's choice is highly affected by race, the gender of the user, and the attractiveness of the faces.

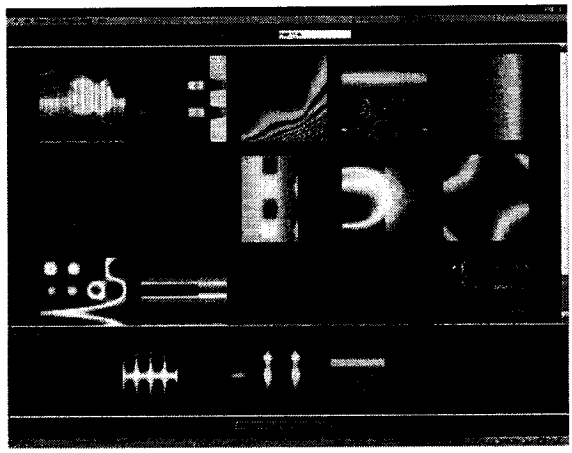


**Figure 8 Déjà Vu [Dhamija and Perrig 2000]**

By exploiting hash visualization techniques [Perrig and Song 1999], another scheme called Déjà Vu [Dhamija and Perrig 2000] was designed with non-describable abstract images (see Figure 8), rather than photographs. The advantage of introducing these kinds of images is that they can be generated deterministically by small initial seeds through a method called Random Art, thus removing the need to store and transmit those cumbersome images back and forth. A user study was also conducted and 20 participants were asked to create Déjà Vu (selecting 5 images among 20 "decoy" images) and textual passwords (at least 6 characters) simultaneously and authenticate themselves by using both respectively. The results showed that it took longer to create a graphical password then a textual password. In addition, 90% of the authentication attempts using Déjà Vu succeeded, compared to 70% using textual passwords. Considering the fact that the password space of textual passwords is much larger than that of Déjà Vu (which is only 53,130), it is not convincing to conclude that Déjà Vu is easier to remember.

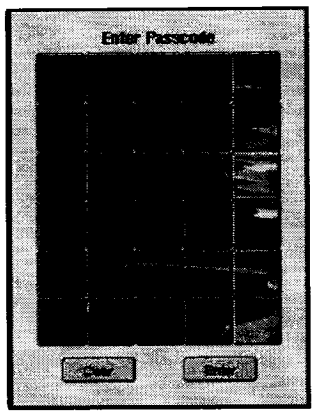



**Figure 9 Picture password [Jansen et al. 2003].**

"Picture password" was suggested by Jansen et al. [Jansen et al. 2003]. The scheme was also designed for mobile devices (PDAs). When a password is to be created, a user selects a theme first (e.g. seashore, kitten and so on) which consists of thumbnail photos. The user then selects a sequence of thumbnail photos as a password (see Figure 9). To repeat the password, the user needs to recognize and identify the thumbnail photos (previously selected) in the same order as in the registration stage. A numerical value is assigned for each thumbnail photo, and the sequence of selection will generate a numerical password. The concept of "akin" was introduced, which serves as a shift key in a traditional keyboard for each thumbnail photo in the theme. For example, instead of picking only one thumbnail photo, a user can select one or two thumbnail photos as one single action. The corresponding alphabet size is then expanded (e.g., from 30 to 930 if the theme consists of 30 thumbnail photos, as in Figure 9). Such expansion significantly enlarges the full password space, and makes an exhaustive search infeasible in practice. However, the difficulty to remember a password is also increased significantly at the same time. The authors also discussed the use of the "Zooming" technique (which magnifies the area of the screen close to the cursor and facilitates the handling of small objects on a display). While "Zooming" can make small thumbnail photos easy to choose, it introduces great complexity in creating and handling themes.

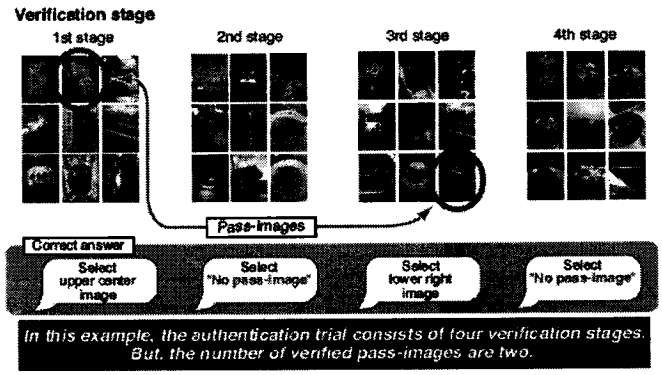



**Figure 10 Image-based authentication for mobile phones [Takada and Koike 2003]**

Takada and Koike discussed image-based authentication for mobile phones using users' own images [Takada and Koike 2003]. In the password registration phase, a user chooses his/her own images as pass-images, and then is required to recognize and identify them among decoy images in the authentication phase (see Figure 10). The user needs to go through several rounds of verification to ensure the password is secure enough. For each round, the user has to select a pass-image or choose nothing if there is not any pass-image displayed. The authentication will succeed if all verifications are successful.

In general, multiple-image schemes suffer from following common shortcomings:

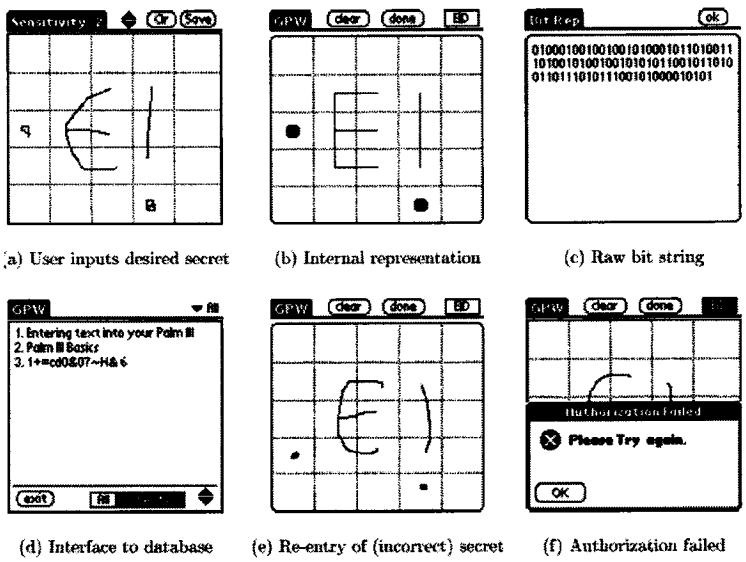
a) considerably large display space is needed to hold multiple images;
b) the password space is small. For example, if one image needs to be distinguished from a 3×3 image matrix for 6 rounds, the full password space is only 531,441, which is even smaller than that for a 3 printable ASCII character textual password ($95^3$=857,375). Such a small password space is subject to off-line attack;
c) passwords have to be stored in the clear, therefore the authentication server is required to be strongly protected;
d) the password is difficult to write down. While this was claimed as a desirable feature, as it could be an effective measure to prevent social engineering attacks, it makes password sharing difficult, thus making system-generated passwords difficult to be

sent to a human user. In other words, this security feature is achieved by sacrificing some degree of usability.

## 2.2 Grid-based schemes

Proposed by Jermyn et al. [Jermyn et al. 1999], DAS (draw-a-secret, see Figure 11) led graphical passwords to a grid background. Using a grid as background has several advantages: first, it eliminates the need to store a graphical database on the server side and removes the overhead to transfer images through network. Second, as a grid is a simple object, such schemes minimize the quality requirement for displays, which is an essential factor in image-based schemes.

Moreover, different from most schemes, grid-based schemes do not impose a limit on the length of a password; a user can draw a password as long as desired. Theoretically, the full password space of a grid-based scheme is infinite. Finally, the passwords in grid-based schemes can be stored as the output of a one way function or used to derive a cryptographic key. The DAS scheme and related literature will be reviewed in §3.1.




(a) User inputs desired secret (b) Internal representation (c) Raw bit string

(d) Interface to database (e) Re-entry of (incorrect) secret (f) Authorization failed

**Figure 11 DAS (draw-a-secret) shceme [Jermyn et al. 1999]**

## 2.3 Common problems and shoulder surfing solutions

One common problem for graphical passwords is the shoulder surfing problem: an onlooker can steal a user's graphical password by watching in the user's vicinity. Unfortunately, there has been no such a satisfactory solution which can convincingly solve this problem. For this reason, most graphical password schemes recommend small mobile devices (PDAs) as the ideal application environment. A mobile device has a relatively narrow viewing angle, and usually has a small size, making it easy to shield data entry with one's body.

Paulson [Paulson 2002] described a graphical password scheme which can resist the shoulder surfing problem. In this scheme, a user navigates through a virtual world which consists of cities and buildings. Users can construct worlds randomly in the registration stage. After creating a password (building a virtual world), a user needs to authenticate a system by navigating to a site that is randomly chosen each time he/she logs in. Because users must navigate by memory to different virtual sites every time they login, this method makes it harder for attackers to learn passwords via shoulder surfing,

[Sobrado and Birget 2002] discuss a number of techniques which aim to solve the shoulder surfing problem. In the first technique (which they call "triangle scheme"), a number of pass-objects are presented, which were previously seen and selected by a user in the registration stage, along with many other "decoy" objects. Then the user is required to find the pass-objects and click inside the convex hull formed by all the pass-objects, as shown in Figure 12. Because the area of the convex hull can be large, the probability of successful login by random clicking for one single round can be high. Therefore, this scheme requires this process to be run for sufficient times (e.g. 10). In order to make the password space large enough, the authors also suggest 1000 objects to be displayed on the login interface. In other words, this scheme requires considerably large display space and significant patience to find pass-objects.

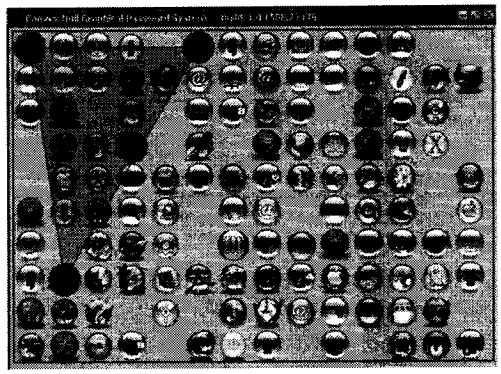


**Figure 12 The triangle scheme [Sobrado and Birget 2002]**

The second scheme is called "movable frame scheme", which also requires a user to recognize some of previously seen pass-objects. This time however, only 3 pass-objects are displayed and one of them is placed in a movable frame (see Figure 13). The user is then required to move or rotate the frame until the pass-object in the frame lines up with the other two pass-objects. This process is also required to run sufficient times to achieve reasonable security.

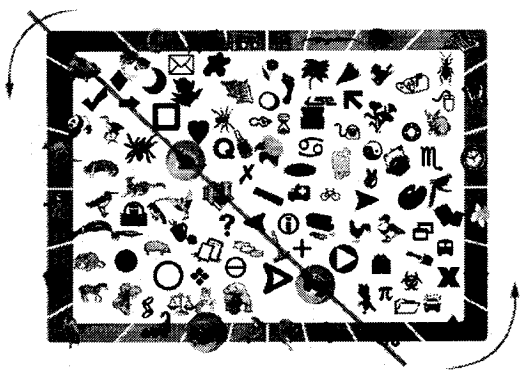


**Figure 13 The movable frame scheme [Sobrado and Birget 2002]**

In the last scheme, four pass-objects are displayed each time, and the user is required to click near the intersection of two invisible lines, which are formed by these four pass-objects, as shown in Figure 14.

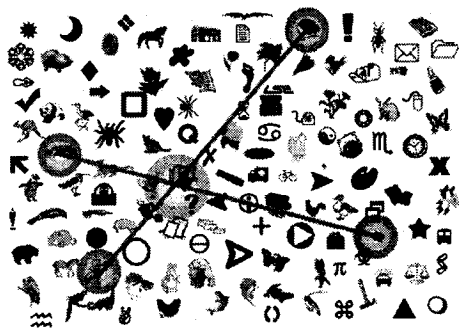


**Figure 14 The intersection scheme [Sobrado and Birget 2002]**

The major drawback of these schemes suggested by [Sobrado and Birget 2002] is the poor usability. First, suggesting 1000 decoy objects to be displayed on a screen is not a good idea, because this will dramatically increase the difficulty to find pass-objects. The objects must be very small to fit on the screen. For example in Figure 12, there are only about 120 objects, and the screen looks already very crowded. Distinguishing 3 or 4 objects from such an image is very time-consuming. Second, requiring this process to run 10 times significantly slows down the authentication process and might be beyond many users' patience.

[Man et al. 2003] suggested another shoulder surfing resistant solution, where a string of textual characters is input by a user, prompted by variable pass-objects, as shown in Figure 15. To create a password, a user has to choose a number of pass-objects and remember the textual characters associated with each variant of their pass-objects. When the user authenticates a system, the variants (randomly chosen) of the pass-objects are displayed in a scene which might contain 400 to 500 decoy objects. The user has to input a string of textual characters, according to the variant and order of the pass-objects displayed. The advantage of this scheme is that even if the process of login is filmed by a video camera, an attacker still cannot derive the password, as the password is time variant (namely different for each login).

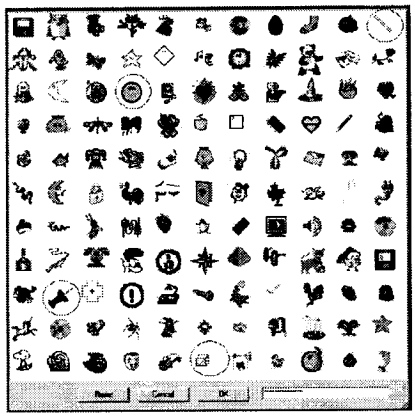


**Figure 15 Shoulder surfing solution of [Man et al. 2003]**

[Hong et al. 2004] improves the scheme of [Man et al. 2003], and allows users to choose their own words to associate with each pass-object variant. For example, "3" can be used to be associated with a pass-object variant which exhibits a shape similar to the shape of "3"; this facilitates the task of password recall. However, this significantly extends the process of password registration. For instance, if each pass-object has five variances and there are eight pass-objects, a user has to make 5×8=40 associations before a password can be created successfully. Moreover, these 40 associations must be remembered by the user. If any one is forgotten, the user might not be able to login successfully. Apparently this substantially increases the difficulty to remember a password and may discourage users to use this password scheme.

Another common problem for graphical passwords is that it takes longer to input graphical passwords than textual passwords. In other words, the process of login is slow and might be a challenge for impatient users. For example, in a multiple-image scheme, several rounds usually need to be passed to complete the process of inputting a password. For each round, the user needs to recognize one or more images and click on them. Such a lengthy login process discourages users if the graphical passwords are frequently used.

[Stubblefield and Simon 2004] suggested a scheme based on the Rorchach inkblot test, where a series of inkblots are generated and displayed to aid users to create and

memorize strong textual passwords. For example, one might associate the inkblots in Figure 16 with the word "men". This process runs for a series of times and the final password is derived from these words (e.g., concatenating the first and last letters of each word). It is supported by psychological studies that users will choose dissimilar associations (the password selected by a user will not be easily guessed), and retain their associations for long period of time (the passwords are easily memorable). The first feature can also be exploited to defend shoulder surfing problem, as an attacker cannot derive the password by watching the Rorchach inkblots only. One drawback of this scheme is that, if an attacker replaces inkblot images sent to a user by other inkblot images, a list of popular letter pairs associated with each substitute image can be readily built. This list can subsequently be used to crack a password of the user. Another problem is that only a small fraction of the available printable ASCII characters can be associated easily. There were no techniques described which encourage users to use upper case letters, numbers and other punctuation characters.

We don't categorize the password schemes suggested by [Man et al. 2003; Hong et al. 2004; Stubblefield and Simon 2004] into the scope of graphical passwords, but rather as mnemonic strategies for textual passwords, such as Passphrases [Yan et al. 2000].

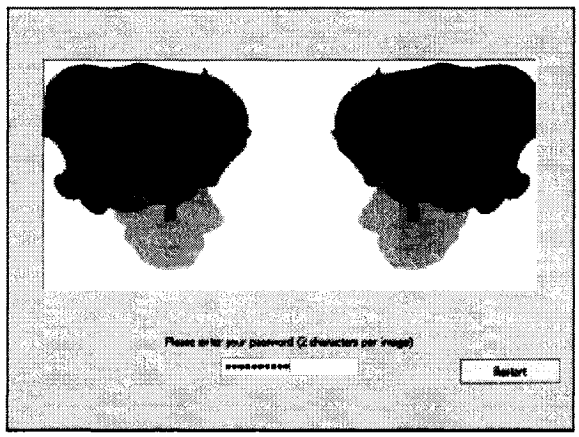


**Figure 16 A scheme based on the Rorchach inkblot test [Stubblefield and Simon 2004]**

One might notice that our categorization of graphical passwords differs from most literature [Suo et al. 2005; Wiedenbeck et al. 2005; Van Oorschot and Thorpe 2005; Dhamija and Perrig 2000], where a boundary line was drawn between recognition and recall. Monrose and Reiter [Monrose and Reiter 2005] recently gave a new categorization of graphical passwords, based on image recognition, tapping or drawing, and image interpretation. However, it seems that recognition and recall are closely interweaved and supplemental aspects in the process of human memorization for any kind of graphical password scheme. For example, in Passpoints, which is widely deemed as a recall based scheme, the major part of a login process is for a user to recognize and identify which points were previously selected. Our categorization was motivated by such an observation and aims to be more intuitive.

# Chapter 3 Design and analysis of a new grid-based scheme

In this chapter, we review the DAS scheme and related work, point out its drawbacks, and then discuss the design of a novel graphical password scheme, Pass-Go.

## *3.1 Review of DAS and relevant works*

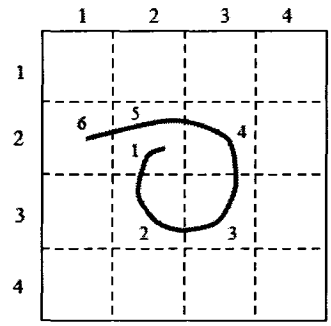



**Figure 17 DAS example password [Jermyn et al. 1999]**

In DAS, a user draws lines on a grid and the display shows the actual trace, as shown in Figure 17. The password is encoded by a sequence of grid cells, represented by two-dimensional coordinate pairs, with "penup" events, represented by distinguished coordinate pairs, inserted into the place where a pen is lifted from the display surface, or a mouse button is released. For example, the password in Figure 17 can be encoded as:

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

where (5,5) is the special coordinate pair used to signify a penup event. Some basic terminology has been defined as follows:

- Neighbors, $N_{(x, y)}$, of a cell *(x,y)* are the subset of the set of cells *{ (x-1, y), (x+1, y), (x, y-1), (x, y+1)}* whose elements exist in the grid. The number of neighbors varies from 2 to 4, depending on where the cell *(x, y)* is;
- A stroke is a sequence of cells $\{c_i\}$, in which $c_i \in Nc_{i-1}$, and which does not contain a penup event;
- The length of a stroke is the number of coordinate pairs it contains;
- The length of a password is the sum of the lengths of its component strokes (excluding the penups).

The full password space of DAS was computed recursively, based on the assumption that passwords of total length greater than some fixed value ($L_{max}$) have a probability zero to be chosen. On a 5 × 5 grid, the full password space is $2^{96}$ if $L_{max}$=20, and $2^{58}$ when $L_{max}$=12, which is larger than that for textual passwords (8 printable ASCII characters $95^8 \approx 2^{53}$).

Jermyn et al. analyzed the memorable password space of their scheme by modeling user choice. First, drawing only two rectangles in different ways could produce $2.56 \times 10^6$ passwords, which is approximately the size of a textual dictionary [Klein 1990]. Second, the authors assume that passwords describable by short algorithms are memorable, thus create a larger memorable password space.

The major drawback of DAS is that diagonal lines are difficult to draw, as stated in the paper: "difficulties might arise however, when the user chooses a drawing that contains strokes that pass too close to a grid-line". However, it is not yet clear how close is "too close". Users have to draw their input sufficiently away from the grid lines and intersections in order to enter the password correctly. If a user draws a password close to the grid lines or intersections, the scheme may not distinguish which cell the user is choosing. Given the fact that a grid is made of grid lines and intersections, this requirement seems to be too strict. Users might get frustrated if consecutive logins fail due to the difficulty to input a password.

This limitation also causes this scheme to require that the cells must be sufficiently large and must not be too small. For this reason, the grid size has to be limited to a small number, like 5, to prevent the grid from occupying too much space on a PDA, which was recommended as the ideal application environment for this scheme. The scalability of DAS is therefore restricted. This limitation further sacrifices the ease of inputting passwords, restricts freedom of choosing passwords (and shapes of drawings), and subsequently reduces the memorable password space and the security provided.

Goldberg et al. [Goldberg et al. 2002] conducted a small scale user study on a similar scheme (Passdoodle). Thirteen participants took part in the study and each of them was asked to draw passdoodles using a pen and paper, rather than in a real system. Passdoodles were required to consist of at least two strokes and could be drawn in multiple colors. A doodle is considered as a full match if it is drawn in exactly the same order as when the user initially drew the passdoodle, and is considered as a visual match if it is not a full match due to stroke order, stroke direction, or number of strokes. They found that the order in which a password is drawn introduced much complexity to graphical passwords and suggested to neglect the order.

Thorpe and Van Oorschot [Thorpe and Van Oorschot 2004a] studied the memorable password space of DAS, and introduced the concept of a symmetric graphical dictionary, based on psychological theories [Attneave 1955] that people prefer images that are (especially mirror) symmetric. It is conjectured that a significant portion of DAS passwords will exhibit mirror symmetric patterns, and be drawn in a "symmetric manner". Several subclasses were defined, including:

- $S_{1a}$: the subset of passwords whose components are symmetric about the *center 3* horizontal and/or vertical axes, and are drawn in a "symmetric manner";
- $S_{1b}$: the subset of passwords whose components are symmetric about the *center* horizontal and/or vertical axes only, and are drawn in a "symmetric manner".

The bit-sizes of $S_{1a}$ and $S_{1b}$ for $L_{max}$=12 on a 5×5 grid are 48 and 43 bits respectively, significantly smaller than that of the full password space (58 bits). The times for such off-line attacks using one or 1000 3.2GHz Pentium™4 machines were estimated, assuming that passwords are hashed by the MD5 algorithm. Their result shows that, for one such machine, it would take 255 or 6 days to exhaustively search $S_{1a}$ or $S_{1b}$ respectively. In the case of 1000 machines, it takes only 6.1 hours or 8.7 minutes to finish such a search of $S_{1a}$ or $S_{1b}$ respectively.

Nali and Thorpe [Nali and Thorpe 2004] conducted a small scale user study, in which 16 computer science and engineering undergraduate students were instructed to draw DAS passwords on a 6×6 grid on paper (similar to the method used in [Goldberg et al. 2002]), rather than in a real system. Subjects could create their passwords without time restriction. Their result shows that the DAS scheme is easy to understand. Also it shows that the location of start and end points for each stroke were scattered evenly on the grid. Finally, approximately 45% of users chose symmetric passwords, 2/3 of which were mirror symmetric. Nali and Thorpe gave detailed information about the distribution of stroke-count: a major proportion of user-chosen passwords (80%) contain only 1-3 strokes, implying it will be effective to exclude passwords with higher stroke-count, as they are less likely to be chosen. Among those symmetric passwords, 19% were symmetric about the vertical axes, and 8% symmetric about the horizontal axes. The user study also found a serious usability problem of the DAS scheme, as 29% of the passwords were invalid. The invalid passwords passed too close either to an intersection or to a grid line, thus in a real system it might be difficult to distinguish which cells they belong to. Ninety-three percent of the invalid passwords passed too close to an intersection, implying there were many attempts to draw diagonal lines. However, they did not provide the information about password length, one of the most important properties of DAS passwords. The information about the number of dots was not provided either. Without this information, it would be difficult to compute the sizes of graphical dictionaries.

Thorpe and Van Oorschot [Thorpe and Van Oorschot 2004b] further studied the impact of stroke-count and number of dots (stroke of length 1) on the security of DAS. By limiting the stroke-count to 4, the resulting subset ($S_2$) of the full DAS password space is only 40 bits when $L_{max}$=12 on a 5×5 grid. They also found that passwords solely composed of dots made up a significant portion (25%) of the full password space, and about half of the all passwords were composed of only strokes of length of 1 or 2. If users do not draw any dots in their DAS passwords, the size of the password space will decrease to 40 bits (similar to the effect of limiting the stroke-count to 4). Among the complexity properties examined in this paper, the stroke-count was shown to influence password space the most. Although the password length also has a significant impact on the size of the password space, its impact is not as strong as the stroke-count. For example, increasing $L_{max}$ by 1 (when there are fewer than $L_{max}/2$ strokes) results in fewer possible passwords than increasing the stroke-count by 1. This is also one of the reasons that we applied stroke-count policies in our user study (see §4.3).

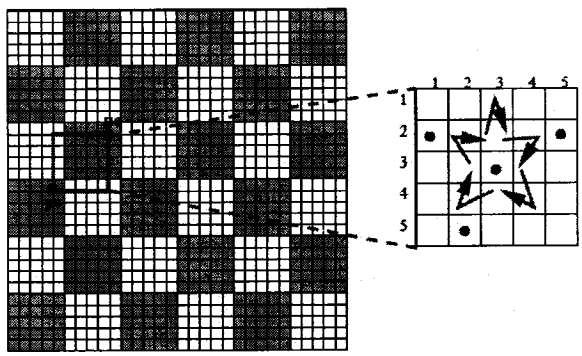



**Figure 18 Grid selection [Thorpe and Van Oorschot 2004b]**

Thorpe and Van Oorschot also suggested a grid selection technique to enhance the security of the DAS scheme. In this technique, a fine-grained grid of larger size (e.g., 30×30) is presented and a user needs to select a small drawing grid and then input their DAS password in the drawing grid, as shown in Figure 18. Some cells in the selection grid are shaded, similar to the idea of reference aids in the design of our new scheme (see §3.2.4). "Zooming in", which was initially discussed by [Birget et al. 2003], was also

suggested to improve usability. By this method, the security of the DAS scheme can be improved by approximately 16 bits.

One drawback of "grid selection" is that it introduces additional tasks to memorize and to input a password. The technique can actually be thought of as a separate password scheme, as it can work individually. In other words, the security improvement is achieved by sacrificing some degree of usability and memorability. Moreover, the fine-grained selection grid might require a large display space, thus limiting the flexibility of the scheme.

Van Oorschot and Thorpe [Van Oorschot and Thorpe 2005] suggested that an attacker might prioritize his/her dictionary and search the intersection of $S_{1b}$ and $S_2$ first. The size of that subset of the password space ($S_{1b} \cap S_2$) is only 31 bits when $L_{max}$=12 on a 5×5 grid, which could be effectively exhausted by one 3.2GHz Pentium™4 machine in only 2.1 minutes.

To conclude that the security of DAS had been originally overestimated, one question has to be clearly answered: how many user-chosen passwords will fall into their graphical dictionaries (i.e., how successful would such attacks be)? Unfortunately, there has been no user study to explicitly answer this question.

## 3.2 Design of Pass-Go

In this section, we explain the design of Pass-Go. Pass-Go can be thought of as an improvement or an alternative (discrete) implementation of DAS, which is easier to use and simultaneously allows a larger grid size, and therefore leads to better (stronger) passwords. Nevertheless, we treat Pass-Go as a new scheme to facilitate our comparison with DAS and the introduction of the subsequent variation mechanisms, e.g., curved line indicators (see §6.3), which can not be easily applied to DAS.

### 3.2.1 Select intersections instead of cells

Pass-Go is a grid-based scheme. However, different from DAS, Pass-Go requires a user to select (or touch) intersections, instead of cells, as a way to input a password. Consequently, the coordinate system refers to a matrix of intersections, rather than cells as in DAS.

As an intersection is actually a point, which doesn't have an area, theoretically it would be impossible for a user to touch it without an error tolerance mechanism. Therefore we introduce sensitive areas to address this problem. A sensitive area is an area surrounding each intersection, as shown in Figure 19.

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

The most obvious advantage of changing from cell to intersection is that drawing diagonal lines becomes feasible, as shown in the letter "h" in Figure 19. A user can draw a shape more freely, compared to the DAS scheme. Moreover, a $G \times G$ grid in Pass-Go is actually a $(G-1) \times (G-1)$ grid in DAS. With the same display space, therefore, Pass-Go can implement a grid of larger size, thus offering stronger security.

Below we use Pass-Go-$G$ to denote a Pass-Go implementation based on a grid which is comprised of $G$ horizontal and vertical lines, and DAS-$G$ to denote a DAS implementation based on a $G \times G$ grid, as defined in [Jermyn et al. 1999] (i.e., a grid

comprised of $G+1$ horizontal and vertical lines).

### 3.2.2 Indicators

While in DAS the trace of the input device's actual movement is shown, in Pass-Go, dot and line indicators are displayed to show the intersections and grid lines that correspond most closely with the input trace, as shown in Figure 19. A dot indicator appears when one intersection is selected (or clicked), and a line indicator appears when two or more intersections are touched continuously (by dragging the input device). The thickness and pattern of indicators can be optimized to give the best visual perception effect for users.

Instead of slightly different trace every time, as occurs in DAS, Pass-Go passwords are always acknowledged by invariable indicators, which we believe will repetitively impact a user's brain and thus accelerate the process of memorization.

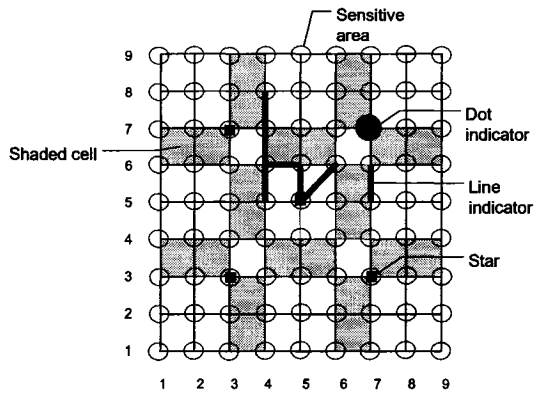



**Figure 19 Pass-Go design**

### 3.2.3 Encoding

The password is then, similar to DAS, encoded as a sequence of intersections, represented by two-dimensional coordinate pairs, with penup events, represented by (0,0) here, inserted into the place where breaks occur. For example, the password in Figure 19 can be encoded as:

(4,8), (4,7), (4,6), (4,5), (0,0), (4,6), (5,6), (5,5), (6,6), (0,0), (7,7), (0,0), (7,6), (7,5),(0,0)

We have the definitions similar to DAS, as follows:

- The length of a password is the total number of coordinate pairs, excluding penups, in the encoding of a password;
- The stroke-count of a password, is the total number of penups in the encoding of a password;
- The dot-count of a password, is the total number of strokes of length 1;
- $L_{max}$, represents the maximum length, beyond which a password is considered with zero possibility of being chosen;
- Neighbors, $N_{(x,y)}$ of a cell *(x,y)* are the subset of the set of cells *{ (x-1, y-1), (x-1, y), (x-1, y+1), (x, y-1), (x, y+1), (x+1, y-1), (x+1, y), (x+1, y+1)}* whose elements exist in the grid. The number of neighbors varies from 3 to 8, depending on where the cell *(x, y)* is.

### 3.2.4 Reference aids

The idea of reference aids was borrowed from the Go game, where 9 small dots (called stars), evenly distributed on a 19×19 Go board, have aided people to play for thousands of years. In Pass-Go, besides 5 stars, we introduce shaded cells as shown in Figure 19, to improve usability, memorability, and scalability.

After introducing reference aids, we surprisingly found that a grid of size 9×9 was a better choice for most applications. One might worry that a 9×9 grid occupies too much display space or reduces usability, compared to a 5×5 grid in DAS. However, in our implementation the 9×9 grid only takes 25 $cm^2$ (5cm×5cm), which can be held in a normal PDA easily. Our user study shows that the usability of such an implementation is quite acceptable (see §4). Increasing the grid size from 5 to 9 significantly enlarges the password space, as we will show in §3.3. Also in a larger grid, users have more freedom to choose the sizes and locations of their drawings; the memorable password space is thus enlarged.

We even implemented Pass-Go-19 in a 36 $cm^2$ (6cm×6cm) area on our user study website, and experienced no difficulty to input passwords (with a well-functioning mouse). This reminds us that a 9×9 grid might not be the optimal choice for Pass-Go. The usability and security of Pass-Go based on a larger grid size (e.g. 11, 13 or even 15) should be further studied. We leave this as an open problem.

### 3.2.5 Colored Pass-Go

Color is an important part of human vision and can be utilized in grid-based schemes to strengthen security. In our implementation, we chose eight colors: black, red, blue, yellow, green, pink, cyan, and magenta. The color of the input device can be switched by clicking on each color button, which can be designed to surround the grid. Colored indicators display accordingly. The default color can be set to black; that is, a user does not need to perform any extra operation if he/she only uses black. Color codes can be inserted into the place where color switching occurs and be excluded when counting the password length (along with penup). If the first stroke of a password is black, the color code for black will be automatically added to the beginning of the encoding for the first stroke, even though there is actually no color switching event. We define the color-count

as the total number of color codes in the password encoding, and a colored password as a password which contains one or more color codes other than black.

## 3.3 Full password space

We use the same recursive method described by Jermyn et al. [Jermyn et al. 1999], to compute the full password space of Pass-Go. Different from DAS, the maximum number of neighbors for an intersection is 8 in Pass-Go, so we modify the function $n(x,y,l,G)$ [1] from

$$n(x,y,l,G) = n(x-1,y,l-1,G) + n(x+1,y,l-1,G) + n(x,y-1,l-1,G) + n(x,y+1,l-1,G)$$

to

$$n(x,y,l,G) = n(x-1,y-1,l-1,G) + n(x-1,y,l-1,G) + n(x-1,y+1,l-1,G) + n(x,y-1,l-1,G) + n(x,y+1,l-1,G) + n(x+1,y-1,l-1,G) + n(x+1,y,l-1,G) + n(x+1,y+1,l-1,G)$$

Let us define *NumberOfColors* as the number of colors available in a Pass-Go implementation (throughout this thesis, we assume *NumberOfColors=8).* To compute the full password space for colored Pass-Go, we further modify the function $N(l,G)$ [2] from

$$N(l,G) = \sum_{(x,y)\in[1..G]x[1..G]} n(x,y,l,G)$$

to

$$N(l,G) = \sum_{(x,y)\in[1..G]x[1..G]} n(x,y,l,G) \times NumberOfColors$$

From the above modifications, we see that for the same size grid (as defined in each scheme) and the same parameters, the full password space of DAS is a subset of that of

[1] $n(x,y,l,G)$ is the number of strokes of length $l$ ending at the intersection or the cell $(x,y)$ in a $G \times G$ grid, as defined in each scheme.

[2] $N(l,G)$ is the number of strokes of length equal to $l$ in a $G \times G$ grid, as defined in each scheme.

Pass-Go, which is a subset of that of colored Pass-Go. The bit-sizes of full password spaces for Pass-Go-9 and colored Pass-Go-9 are given in Table 1. Values given are $log_2$(number of passwords).

Table 1 Full password space in bit-size for Pass-Go-9 and colored Pass-Go-9 (*NumberOfColors*=8)

| $L_{max}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Pass-Go-9 | 6 | 13 | 19 | 26 | 32 | 39 | 45 | 52 | 58 | 64 |
| Colored Pass-Go-9 | 9 | 19 | 28 | 37 | 47 | 56 | 65 | 75 | 84 | 94 |
| $L_{max}$ | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Pass-Go-9 | 71 | 77 | 83* | 89* | 96* | 102* | 109* | 115* | 121* | 128* |
| Colored Pass-Go-9 | 103* | 112* | 121* | 131* | 140* | 150* | 159* | 168* | 178* | 187* |
| $L_{max}$ | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| Pass-Go-9 | 134* | 141* | 147* | 153* | 160* | 166* | 173* | 179* | 185* | 192* |
| Colored Pass-Go-9 | 196* | 206* | 215* | 224* | 234* | 243* | 252* | 262* | 271* | 280* |
| $L_{max}$ | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| Pass-Go-9 | 198* | 205* | 211* | 217* | 224* | 230* | 236* | 243* | 249* | 256* |
| Colored Pass-Go-9 | 290* | 299* | 308* | 318* | 327* | 336* | 346* | 355* | 365* | 374* |

* Values are computed approximately with the method given in Appendix B, with a maximum error of 3.25 for Pass-Go, and 0.43 for colored Pass-Go

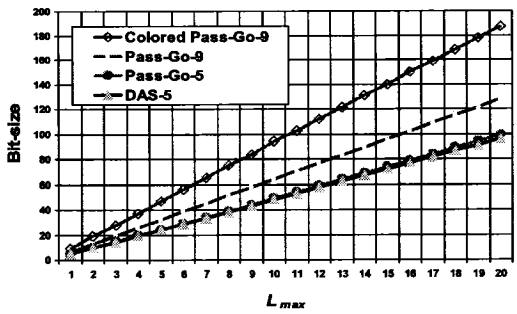



**Figure 20 Comparison of full password spaces**

In Table 1, we consider the maximum value of $L_{max}$ as 40, as justified by our user study (see details in §4.5.1), and we see that Pass-Go offers an extremely large full password space (256 bits for Pass-Go-9 and 374 bits for colored Pass-Go-9).

Figure 20 shows the comparison of the password spaces of colored Pass-Go-9, Pass-Go-9, and DAS-5. To make a more reasonable comparison between Pass-Go and DAS, we also add the password space growing line of Pass-Go-5. We see that the password space for colored Pass-Go-9 grows at an exponential rate of approximately 9.3 bits per unit increase in password length, and that for Pass-Go-9 grows at a corresponding rate of approximately 6.5. For DAS-5 the password space grows at a corresponding rate of approximately 4.8.

It is interesting to note that the full password space of Pass-Go-5 is only larger than that of DAS-5 by an indiscernible degree. This means that although increasing the maximum number of neighbors from 4 to 8 extends the full password space, the difference is not significant. We will discuss how the memorable password space is affected by the increased neighbor count in §4.5.5.

## *3.4 An efficient and human readable encoding scheme*

A password is encoded by the sequence of the encodings of each composite stroke, which is separated by a penup code (and color code if applicable). A stroke is encoded by the coordinate of the first selected intersection, followed by one or more movement encodings (if the length of the stroke is greater than one). Finally, a movement encoding is composed of a direction code and the number of intersections subsequently passed in this direction.

In Pass-Go-9, for instance, let 0 represent penup, and 1-8 represent the direction codes, as shown in Table 2.

Table 2 Direction codes

| Direction code | Direction | Direction code | Direction |
|---|---|---|---|
| 1 | Right | 5 | Left |
| 2 | Up-right | 6 | Down-left |
| 3 | Up | 7 | Down |
| 4 | Up-left | 8 | Down-right |

Color codes (if applicable) can be encoded by 01-08. The password in Figure 19, for example, can then be encoded as:

48730461171210770 76710

For the first stroke, 48 is the coordinate for the first intersection. The following 7 is a direction code to represent "Down", and the 3 represents the number of intersections subsequently passed in this direction. The stroke is then ended by 0 (penup code). The rest of the strokes are encoded in the same manner.

One advantage of this encoding scheme is that it is more efficient than the encoding scheme given in §3.2.3. For example, it uses only 22 digits to encode the password in Figure 19, compared to 30 digits which is the case for the original encoding scheme. Another advantage of this encoding scheme is that its length (the number of digits in the password encoding) could more closely reflect the time (or difficulty) to input a password than the original encoding scheme.

For example, locating the first intersection includes two sub-tasks: locating vertically and horizontally. This action is encoded with 2 digits. Each movement in one direction also includes two sub-tasks: moving in the direction and stopping at the place where a penup occurs or a turn is to be made. This action is also encoded with 2 digits. Penup takes less time than locating an intersection or making movement in one direction. It is thus encoded with only 1 digit. As the encoding of a password is the concatenation of the

encoding of each composite strokes (with penup codes inserted), compared to the original encoding scheme, the length of a password in this new encoding scheme can be used to measure the time (difficulty) to draw the password in a closer way. Generally speaking, in this new encoding scheme, the more digits the encoding of a password contains, the more difficult it is to input it.

We believe that the user's choice on Pass-Go passwords will be influenced by the difficulty of inputting a password. Users will prefer a password which is easier to input, if the same level of security can be achieved. If a password takes too long to input (e.g., contains too many penup or turns), it will be less likely to be chosen. Our new encoding scheme can thus be used to reduce the size of (or prioritize) a graphical dictionary. For example, if the encoding of a password is too long (i.e., contains too many digits), such a password can be excluded from the graphical dictionary, as it will take too long to input and thus be less likely to be chosen.

An alternative for the movement encoding is to repeat the direction codes multiple times based on the number of intersections subsequently passed. For example, the password in Figure 19 can be encoded as:

4877704617207707670

For the first stroke, 48 is the coordinate for the first intersection. The following "777" means that the input device will be dragged downwards for 3 units. This stroke ends with a "0" (penup). The following "46" is the coordinate for the first intersection of the second stroke, and the "172" represents that the input device will be dragged toward "right", "down", and "top-right" respectively for one unit each. The rest of the strokes are encoded in the same manner. For this example, we see that this encoding is only 19 digits, shorter than the previous one (22 digits).

As our user study shows that the average number of intersections passed per movement (in one direction and excluding the first one) is 2.68, this alternative might be

more efficient for Pass-Go on a grid of smaller size, where movements in one direction may be shorter.

For a human readable encoding, we suggest to replace direction codes with symbols like "→ ↗ ↑ ↖ ← ↙ ↓ ↘", and penup with ",", to improve the readability. Then the previous password can be encoded as:

$$(4, 8)\downarrow_3, (4, 6)\rightarrow_1\downarrow_1\nearrow_1, (7, 7), (7, 6)\downarrow_1,$$

## *3.5 Keyboard input and textual display support*

Although input devices other than keyboard have been widely used, (e.g., mice, stylus, touch screen), we consider the situations when a mouse does not work well, or a stylus is not available (e.g., lost or broken). We provide a keyboard input support solution in this section to improve the functionality of Pass-Go.

Based on the new encoding scheme, implementing keyboard support is feasible. By using the number key pad, which has been standardized for a traditional keyboard, 8 direction codes can be input through the 8 number keys surrounding the center "5". A cursor which indicates the location of the "current intersection", can thus be moved by pressing the direction keys. The "enter" key can work as the left button of a mouse to select intersections. When a line is to be drawn, the "enter" key can be held until penup occurs.

Furthermore, when a Pass-Go password needs to be input with a textual display (e.g., a dumb terminal), a user may just input the encoding of the password by keyboard. Textual assistance can be designed to aid users. For example, direction symbols (e.g. ↓) can be displayed when a movement code is entered. With this method, a Pass-Go password can be used even for applications like telnet and ftp, in a DOS format interface.

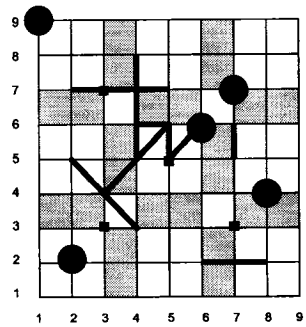



**Figure 21 Disguising indicators**

## *3.6 Solutions to shoulder surfing*

In this section, we suggest two solutions which might work to some extent to alleviate the shoulder surfing problem. One is not to show indicators at all. By only observing the movement of the pointer, we assume that it is very hard to learn a password. We implemented this method in our user study, and will discuss the result in §4.4.3.

The other is to use disguising indicators (see Figure 21). In response to each user input, one or more disguising dot or line indicators may be displayed in random positions along with the true ones. A disguising dot indicator or disguising line indicator has the same style, shape, color and size as the real dot indicator or line indicator.

## *3.7 Dynamic password policy*

Conventional password policies make rules to disallow certain kinds of weak passwords [Yan, 2001; Summers et al. 2004]. For example, a textual password policy can reject passwords which do not contain any capital letters. Such a policy, however, can be taken advantage of by an attacker, as all those forbidden passwords (some of them might be

strong passwords) can be excluded from the attacker's dictionary, whose size is thus reduced.

One solution is to monitor the password file (or database), and apply a password policy dynamically, according to the state of the password distribution. For example, for typed passwords, let $q$ denote the percentage of the passwords which contain at least one upper-case letter. Also let us assume the ideal value for $q$ is $q'$. When the value of $q$ falls below $q'$, the password policy will begin to work (i.e., the subsequent new passwords will be required to contain at least one upper-case letter); if the value of $q$ rises above $q'$, the password policy will stop working (i.e., the subsequent new passwords will not be required to follow the policy). In this way, the distribution of passwords can be controlled to keep it close to $q'$. However, as nowadays passwords are hashed in most systems, such monitoring is not feasible for most systems.

We propose a dynamic password checking method, in which a policy takes effect with a probability of $p$, rather than 100% in conventional password policy (which we call a static policy). Given sufficient times of password creation, change, or cancellation, the password distribution in a password file can thus be manipulated. We will discuss the result of this dynamic password checking method in §4.5.7.

# Chapter 4 User study

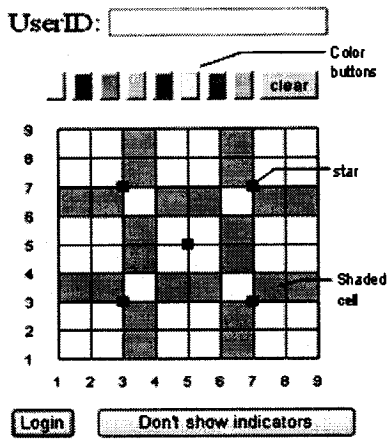



**Figure 22 Main login interface**

## *4.1 Objective*

One aspect of our user study aimed to test the usability of Pass-Go. Is it simple? Is it easy to understand and convenient to use? How difficult would it be to create a new password, if certain password policies are applied?

Another goal of our user study was to learn the characteristics of user-chosen passwords in a real system (i.e., in an environment where the passwords will be used frequently over a period of time); for example, how long they are, how many strokes, dots, and colors they contain, where they start and end, what patterns they exhibit. In short, will they be easy to guess?

As well, we tried to find out how stroke-count policies can affect the users' choices. Will the dynamic password checking method suggested in §3.7 work in the way we expect?

## 4.2 Implementation

Colored Pass-Go-9 was implemented as a Java Applet, using SUN netBeans IDE 4.1 as the developing environment. A simple teaching management system was also developed with JSP (JavaServer Pages) technology on a website, which is accessible through the Internet by Java enabled browsers. The size of the applet is 230 (width)×270 (height) pixels. For a regular 14" display with screen resolution of 1024×768 pixels, the 9×9 grid occupies an area of $25cm^2$ (5cm×5cm). The Java Applet was embedded into the main login web page, as shown in Figure 22.

We notice that the shape and size of sensitive areas have a strong impact on the usability of Pass-Go. The smaller the size of the sensitive areas is, the more difficult for a user to touch them. If the sensitive area is too small, the user has to slow down the speed of drawing a line; otherwise, the sensitive areas might not be touched. On the other hand, if it is too big, then the gaps between the sensitive areas will be small. In other words, the sensitive areas are too close. This will result in a high probability of touching neighbor sensitive areas when drawing a line (especially at a fast speed).

After consecutive tests over a period of time (about 2 weeks), we determined to use a round circle as the shape of sensitive areas, and $0.4 \times d$ (where $d$ is the side length of a grid cell) as the radius of the circle. According to our test as well as our user study, we believe this setting is close to the optimal value for Pass-Go-9. The optimal value should be particularly studied along with theories and technologies from other fields (e.g., mechanics or psychology). We leave this as an open problem.

To improve usability, we added a "clear" button which can erase all the previously inputted strokes. In addition, a "don't show indicators" button was deployed to switch from "show indicators" to "not show indicators" mode and vice versa. This is one of the solutions we suggested in §3.6 to alleviate the "shoulder surfing" problem.

A relational database (MySQL 4.1.13) was used on the server side to store user names, passwords, and other relevant data. Passwords were stored in the clear to facilitate our access and study.

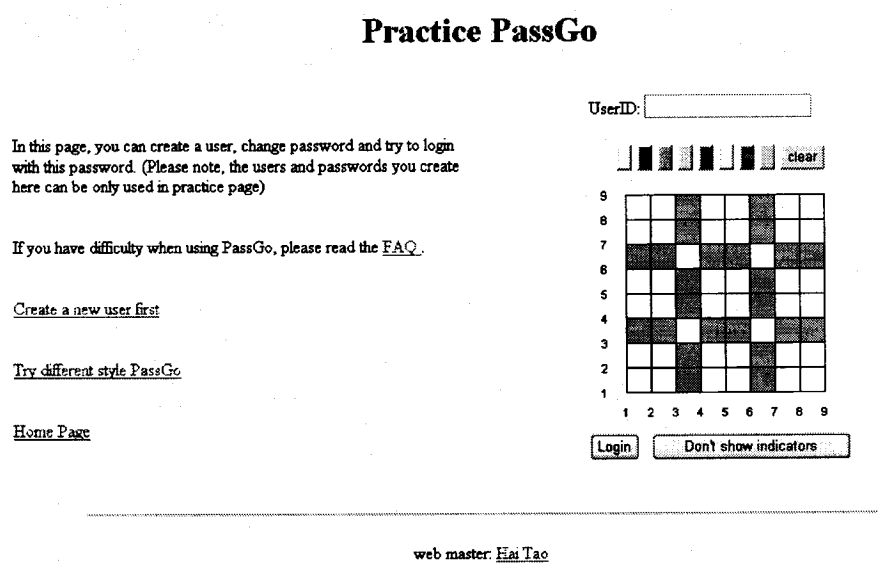



**Figure 23 Practice page interface**

In addition, a practice page was set up as shown in Figure 23. In this webpage, a similar applet for Pass-Go-9 was embedded. The only difference with the one used in the main login interface is that "stars" were removed from the grid. In other words, the only reference aid used in this applet was the shaded cells. The main purpose was to examine how the "stars" affect the distribution of starting points of user-chosen Pass-Go passwords.

Any Internet user who visited our website was allowed and encouraged to create an account (a user/password pair) in this practice page, and was able to change his/her password after registration. Once an account was created, the user could login from the practice page to see his/her password encoding, an extremely limited function. The collected passwords from this practice page will be discussed in §4.6.

Moreover, in order to show and compare the scalability and usability of Pass-Go and its variations, we implemented Pass-Go-19, a smaller Pass-Go-9 (compared to those used in the main login interface and the practice page), and a simplified version of curved Pass-Go-9 (see §6.3), and embedded the corresponding applets into a demo page, as shown in Figure 24. In this demo page, visitors could draw Pass-Go passwords on the grids, but could not create an account in our system or log into anywhere.

The 19×19 grid for Pass-Go-19 takes an area of 36 $cm^2$ (6cm × 6cm), and the small Pass-Go-9 occupies an area of only 7.29 $cm^2$ (2.7cm × 2.7cm). Such implementations test the flexibility and usability of Pass-Go for some extreme situations (e.g., extreme high security levels are needed or considerably small display space is available).

The 9×9 grid for the curved Pass-Go-9 occupies an area of 29.16 $cm^2$ (5.4cm × 5.4cm). Both the sensitive areas and cell centers (see §6.3) for the curved Pass-Go-9 have a round circle shape, whose radius is $d/4$ (where $d$ is the side length of a grid cell). The sizes of the sensitive areas and cell centers for the curved Pass-Go-9 were also determined after consecutive tests over the same period of time as for Pass-Go-9 (about 2 weeks). For simplicity, in this version of curved Pass-Go-9, diagonal curved lines (e.g., the example stroke on the right of Figure 43) are not allowed to draw.

In this way, we provided a method for Internet visitors who are interested in our new scheme or website to learn and practice Pass-Go. Another purpose of this demo page was to facilitate our study on the usability of Pass-Go and its variation mechanisms. At the testing stage, I asked family, friends, and classmates, to visit the website and draw a couple of passwords. Their feedback was taken into account when the final decision of the size of sensitive areas and cell centers were made.

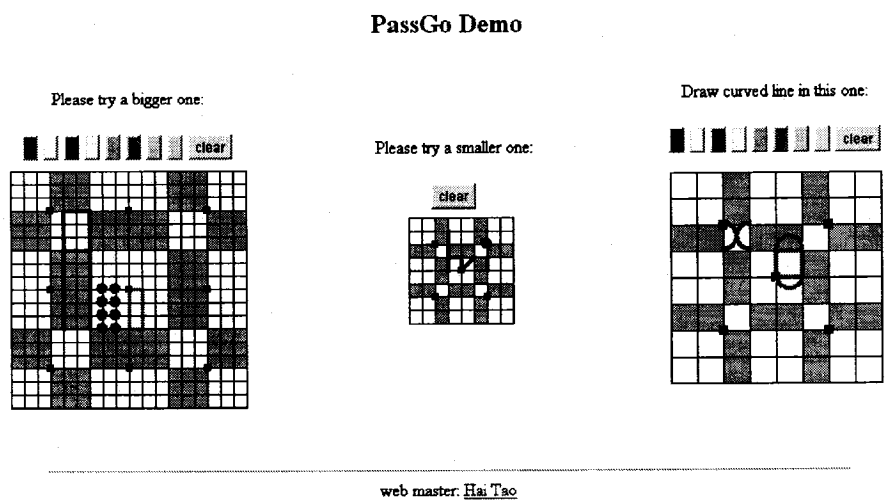



**Figure 24 Demo page**

## *4.3 Outline of User study*

The user study was conducted in the fall semester of 2005 in two fourth-year Computer Science and Electrical Engineering university classes, over a three month period, from late September to late December. In total 167 subjects participated, including 158 undergraduate students, 7 graduate students (Teaching Assistants) and 2 professors. Professors and TAs posted marks, assignments, laboratory instructions, and relevant materials on the teaching management system, all of which were protected by Pass-Go passwords.

Shortly after the beginning of the semester, participants were given a 15 minute tutorial in the class by the author of this thesis. Because our user group consists of experienced computer users with solid computer knowledge and represents a relatively high education level, they might perform better than the general population in understanding our scheme. To somewhat compensate for this, we only used plain language in the tutorial (no technical terms were mentioned, such as coordinate system or how a password is encoded).

Also, to simulate a real application environment, we did not make any effort to encourage attendance: attendance was not required, recorded, or marked, and the date/time of the tutorial was not pre-announced. We estimate the attendance rate on the day of the tutorial was approximately 80%. Our password scheme was then explained. Ways to draw dots and lines on the grid along with one or two sample passwords were demonstrated. One sample password used in the tutorial is given in Figure 25. Some basic concepts, such as password length, stroke-count and "stars" were clarified, in order for them to understand the policies they were going to face. Students were not given suggestions about how to choose a secure password or any mnemonic strategy. Existing analysis on grid-based schemes (such as that symmetric and small stroke-count passwords might be subject to dictionary attack) was not mentioned.

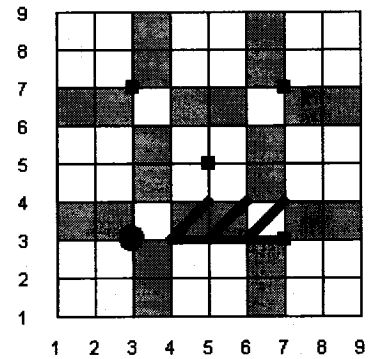



**Figure 25 Sample password used in the tutorial**

They were informed that a FAQ page (see detail in Appendix C) was available on the website in case they need help. A link to the website was given on the course webpage for those students who did not attend the tutorial. (It may also be worth mentioning that the first language of the author of the thesis is not English, so a better understanding may be expected if the tutorial had been given by a fluent English speaker.)

Each participant had to login with a common initial password to change their password, and then the website content becomes available. The length of a password must be at least eight, which is considered as a basic requirement. Participants were

randomly divided into 5 groups and each group was subject to one specific password policy, as shown in Table 3.

When a password was to be changed, the system would prompt the user according to the policy that applied to him/her, as shown in Figure 26. If the password chosen by the user did not satisfy the policy, the password would be rejected. We recorded how many times passwords were accepted or rejected (and for what reasons), in order to examine the difficulty of creating a new password under different password policies (see detail in §4.4.2).

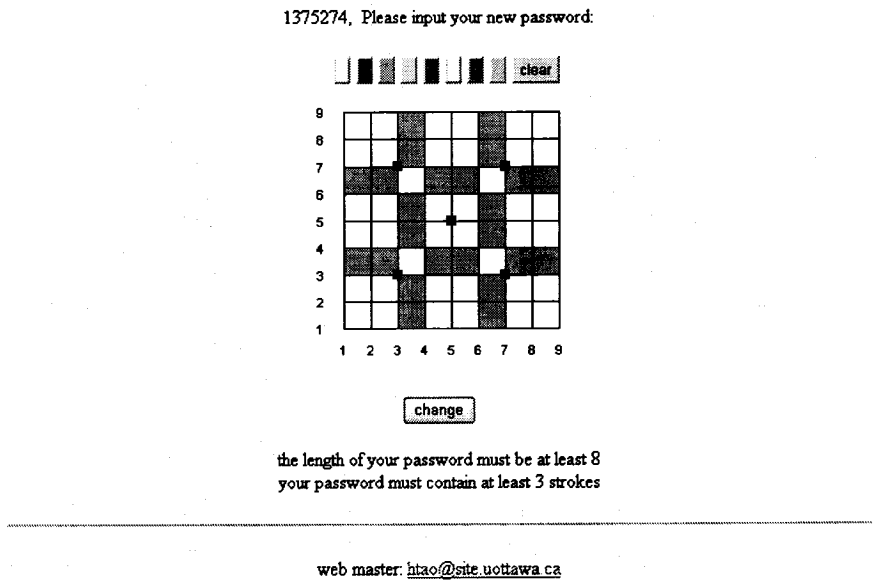



**Figure 26 Interface of changing password**

Because the stroke-count was shown to have the largest impact on password space [Van Oorschot and Thorpe 2005], stroke-count policies, which require the minimum stroke-count to be from 1 to 4, were applied to groups 1 to 4 respectively, to examine how stroke-count policies will affect user choices (there was actually no policy in group 1, as any password with length of at least eight must contain at least 1 stroke). In group 5, however, we applied the dynamic password policy suggested in §3.7, with a probability

of (81-9)/81≈89%, to disallow passwords which start from stars or corners (corners refer to the intersections (1,1), (1,9), (9,1), and (9,9)). The purpose was to measure if this dynamic policy could make the distribution of starting point more uniform. If a password was forgotten, the user had to inform the author of the thesis, who then reset the password after authenticating the user's identity.

While we did suggest that users change their password at their earliest convenience after the tutorial, to prevent others from breaking into their accounts with the well known common initial password (which, as it turns out, never happened), it was left to the users themselves to make the decision when (or even if) to change their passwords.

Once a password is changed the first time from the initial common password, it will go to our user-chosen password database. Therefore, the entries of this database began to grow gradually after the tutorial. It took almost one month for our user-chosen password database to collect about 150 passwords, and the total number of passwords finally stabilized at 167 at the end of the 2$^{nd}$ month, when the grades for the midterm were available.

At the end of the semester, participants (including professors and TAs) were asked to fill out a questionnaire, which was anonymous and also voluntary. To achieve a better return rate, we only asked some simple questions and demographic information. Eighty-eight questionnaires were returned; we assume this sample represents all our participants and their opinions.

The result of questionnaire shows that about 93% of our participants are male and 7% are female; the ages range from 20 to 40 with average of 23. Twenty-one languages are spoken by the participants, so apparently our participants represent a multicultural community.

## *4.4 Analysis on usability*

### 4.4.1 Training and login

Regarding the simplicity of Pass-Go, 59% of the participants indicated that they fully understood the scheme right after the 15 minute tutorial and 12% after reading the FAQ page (most possibly those people who did not attend the tutorial). However, 22% of the participants had to experience a couple of successful logins to learn precisely how the scheme works, and 7% of the participants were still not sure about the scheme even at the time of the questionnaire.

The latter two kinds of users sum up to nearly 30% of all participants and map to about 48 users. Some participants commented that having to draw the password in exactly the same order was the most difficult part to understand and often caused failure to login. This shows that our training was somewhat insufficient. This insufficiency directly results in the low login success rate and high password-forgotten report rate in the first month.

During this three month period, there were in total 6800 login attempts and 5291 of them were successful (the success rate is 78%). The weekly login success rate is shown in Figure 27.

Figure 27 shows that the login success rate was low in the first three weeks, but kept going up until the $7^{th}$ week, when it became stable at around 90%. The opposite trend was observed for the password-forgotten report rate, as shown in Figure 28. There were in total 21 password-forgotten reports, 95% of these were reported in the first 5 weeks, with only 1 report in the subsequent 2 months.

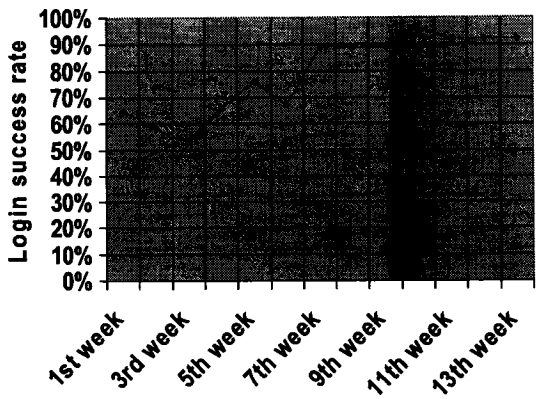



**Figure 27 Weekly login success rate**

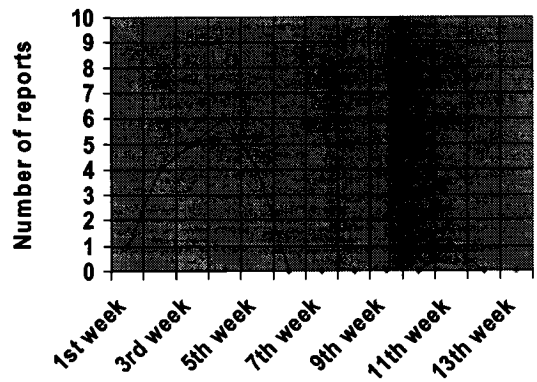



**Figure 28 Password-forgotten reports**

We believe the low login success rate and high password-forgotten rate at the beginning arise from our insufficient user training and the difficulty to understand that "the order matters". If a user hasn't fully understood that a password has to be repeated in exactly the same order, it is very possible that they are rejected because of the wrong order in which they draw the password strokes. Moreover, even with a more thorough user training, we suspect that the same difficulty with a less degree would still be met, as 22% of our participants indicated that they had to experience a couple of successful logins to fully understand how the scheme works.

Our speculation is supported by the stable high login success rate and low password-forgotten report rate in the subsequent two months. Given the fact that any new password scheme needs a certain time to become familiar, we believe that such a period in our scheme is acceptable.

### 4.4.2 Create a new password

Table 3 shows the performance of different groups when users created new passwords following different password policies. The overall success rate is 71%, meaning that on average 1.4 attempts are required to create a password successfully. Comparing each group, we see that the success rate decreases from 86% to 50% from group 1 to 4. On the other hand, 42% of the attempts were rejected due to "less stroke-count than required" in group 4, compared to 12% in group 2. It appears that the higher the required stroke-count, the more difficult it is for users to create a password successfully.

The dynamic policy and the basic requirement on length seem to be easier to understand and obey, as only 18% of the attempts were rejected because the passwords they chose were shorter than 8, and only 16% of the attempts in group 5 were refused due to "starting from stars or corners".

In general, the overall performance of Pass-Go is acceptable in terms of the ease in creating a new password under certain policies, because even in group 4, the success rate is 50%, meaning that a user needs only two attempts to successfully create a new password. This suggests to us that such password policies in Pass-Go can be deployed when higher levels of security are needed, with relatively low cost in usability.

Table 3 Success rate to create a new password under various password policies

| Group | Group1 | Group2 | Group3 | Group4 | Group5 | Whole |
|---|---|---|---|---|---|---|
| Policy | | Stroke-count ≥2 | Stroke-count ≥3 | Stroke-count ≥4 | Dynamic policy on starting point | |
| Number of subjects | 46 | 30 | 33 | 32 | 26 | 167 |
| Password accepted | 86% | 79% | 64% | 50% | 72% | 71% |
| Rejected because stroke-count is less than required (1-4) | N/A | 12% | 28% | 42% | N/A | NA |
| Rejected because length is shorter than required (8) | 13% | 14% | 12% | 30% | 11% | 18% |
| Rejected because starting from stars or corners | N/A | N/A | N/A | N/A | 16% | N/A |

### 4.4.3 Other issues

Regarding the "don't show indicators" option, 62% of the participants think it works (65% of them think it affects their logins somehow though), 29% think it doesn't work, and 9% said that they had never used it. Over the three month period, only 189 login attempts (3%) were made without the help of indicators; however, 144 of them were successful, with a success rate of 76%, almost the same as the login success rate with indicators (78%), implying that hiding indicators does not increase the difficulty of inputting a password. We believe that the option would have been used more often if our users had made more effort to prevent others from stealing their passwords.

In addition, a couple of users complained that it was difficult to draw, especially diagonal lines, when they used a laptop touch pad, or in the school labs, where some computer mice did not work very well. This reminds us the weakness of a mouse and confirms the need for our keyboard input support solution.

A few users also suggested an "undo" button which can erase only the most recent stroke instead of the whole thing, to ease the operation when a minor error is made.

## *4.5 Security analysis on the ultimate password database*

During the three month period, passwords were changed a total of 204 times (excluding the resetting for forgotten passwords); therefore, fewer than 37 users changed their passwords two or more times. It is reasonable that users change their passwords if they find that their previous one was too difficult to remember or took too long to input, and thus choose an easier or shorter one.

This speculation is supported by our observation that the average length of all passwords gradually decreased from 18.56 at the beginning to 16.88 at the end. The same trend was roughly observed for stroke-count, dot-count, and color-count. Our following analysis is based on the 167 passwords in the password database at the end of our user study, which should represent the ultimate choice of our users, and thus avoid overestimating the security of Pass-Go.

Please note that only 20% of the participants indicated on the questionnaire at the end of the study that they did their best to create a secure password, while 80% of the participants admitted that they just picked a password that was as easy or as simple as possible, since the website did not contain very sensitive information. This is not surprising as it is common that grades (with corresponding student numbers) are posted on public course websites. This reminds us that our estimate on the security of Pass-Go will be conservative; it is possible that more secure passwords (longer, more strokes, more dots, more colors, or more abstract drawings) may be chosen in a system which protects more sensitive information.

The result for each group and for the whole set of participants is given in Table 4.

### 4.5.1 Length

Our user study shows that users tend to choose very long passwords voluntarily in Pass-Go. If we consider all passwords in the 5 groups as a whole, out of the 167 passwords, the lengths of passwords range from 8 to 41, excluding an extremely long password of 76 (it is interesting that this extremely long password was used 20 times, and 90% of these logins were successful), with an average of 16.88. In group 1 (without any password policy applied), out of 46 passwords, the lengths of passwords range from 8 to 39, with an average of 16.06. This sufficiently justifies our discussion in §3.3, where we set $L_{max}$=40 and obtained an extremely large full password space.

Table 4 Comparison of characteristics of user-chosen passwords between 5 groups

| Group | Group1 | Group2 | Group3 | Group4 | Group5 | whole |
|---|---|---|---|---|---|---|
| Policy | | Stroke-count ≥2 | Stroke-count ≥3 | Stroke-count ≥4 | Dynamic policy on starting point | |
| Number of subjects | 46 | 30 | 33 | 32 | 26 | 167 |
| Avg. length | 16.06 | 17.00 | 18.00 | 17.96 | 15.46 | 16.88 |
| Avg. stroke-count | 3.80 | 4.93 | 6.27 | 5.56 | 6.15 | 5.19 |
| Avg. dot-count | 1.54 | 2.73 | 3.03 | 2.28 | 3.07 | 2.43 |
| Avg. color-count | 1.17 | 1.13 | 1.15 | 1.06 | 1.19 | 1.14 |
| Percentage of passwords starting from stars or corners | 67% | 83% | 60% | 62% | 15% | N/A |
| | 68% | | | | | |
| Percentage of passwords ending at stars or corners | 50% | 50% | 45% | 46% | 23% | 44% |
| | 48% | | | | | |

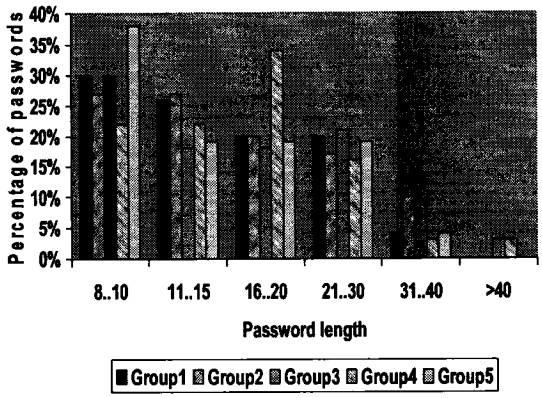



**Figure 29 Password length distribution**

From Table 5 we see that the stroke-count policies did not actually affect the user choice of password length noticeably. The average length in group 4 is 17.96, which is greater than that of Group 1 by only 1.9 units. Thus, the long passwords were unlikely to have been produced by our stroke-count policies. These long passwords strongly suggest (though we cannot show directly), that most users did not encounter much difficulty when inputting Pass-Go passwords. This substantiates the usability of Pass-Go through another means.

We speculate that the long passwords mainly arise from following reasons: 1) drawing along a visible line is easier; 2) a larger size grid encourages users to draw bigger pictures; 3) selecting intersections in Pass-Go is much easier and less error prone than selecting cells in DAS.

We expect longer passwords if the input device is a light pen or touch screen, because a mouse (or touch pad on a laptop) does not work well for drawing, as our users made clear (recall §4.4.3).

Figure 29 shows that the distribution of password length is uneven. For example, 30% of the passwords in Group 1 fall into the range 8-10, while only 5% fall into the range 31-40. It is thus reasonable that an attacker prioritizes the dictionary by length or, sets

$L_{max}$=16, to capture a normal password with an average length. However, when $L_{max}$=16, the full password space is 102 bits (ignoring color), which is still too large to be exhausted.

### 4.5.2 Stroke-count

Out of the total of 167 passwords, the stroke-count ranges from 1 to 12, (excluding an extreme case of 17, which contains 8 lines and 9 dots), with an average of 5.19. In group 1, out of 46 passwords, the stroke-count ranges from 1 to 11, with an average of 3.80.

By comparing each group, we see that the average stroke-count of Group 3 has the largest value of 6.27, which is even greater than that of Group 4. It is also interesting to note that in group 5, where we did not apply a stroke-count policy, the average stroke-count is 6.15, much greater than that of group 1.

If an attacker adopts Thorpe and Van Oorschot's method [Thorpe and Van Oorschot, 2004b] by restricting stroke-count to 4, 72% of the passwords in group 1 can be captured, but only 42% in group 5. In group 4, to capture the same fraction of passwords as in group 1, they have to restrict stroke-count to 6. Figure 30 shows that the percentage of passwords decreases in general along with the growth of stroke-count, meaning that it would be also effective to prioritize a dictionary in the order of stroke-count.

In general, applying stroke-count policies results in larger average stroke-count and excludes small stroke-count passwords; therefore this can be an effective method to improve the security of Pass-Go.

### 4.5.3 Dot

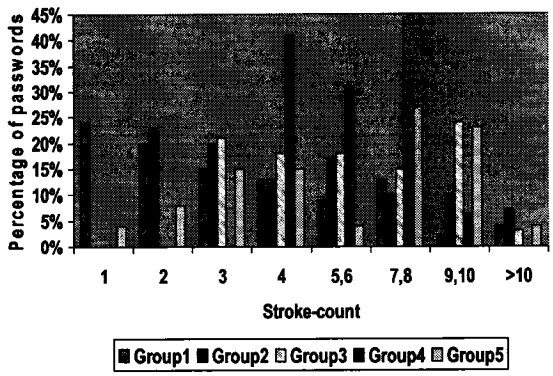



**Figure 30 Password stroke-count distribution**

The average dot-count in each group varies from 1.54 to 3.07, with an average of 2.43 over the whole set of participants, meaning that we can expect 2.43 dots in every Pass-Go password. Out of 167 passwords, 81 of them (48.5%) contain at least one dot. More precisely, 32 passwords (19.2%) are solely composed of dots, and 49 (29.3%) are mixed with lines.

Such a frequent occurrence of dots implies that nearly half of all the passwords in our user study will escape attention if an attacker excludes passwords with at least one dot from his/her dictionary, as suggested by [Thorpe and Van Oorschot, 2004b] for DAS. We believe that our users chose dots so frequently because the design of our scheme optimizes the operation to draw a dot (only one click is needed).

We also expect a higher frequency of dot occurrence if Pass-Go is deployed in China, Korea, and Japan, where Go is one of the most popular games. It is estimated that from 5 to 10 percent of the Korean population plays Go regularly [Usgo 2006b]. Go players might choose a drawing solely composed of dots, to simulate a possible situation in a game (see Figure 31). The special movements like "jump", "short fly", and "long fly"

could be further exploited to make a sequence of dots meaningful and memorable. The dot is also a basic stroke for oriental characters.

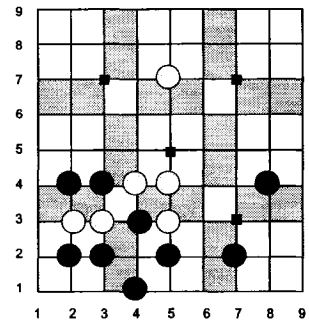



**Figure 31 An example password derived from a situation of Go game**

To ease the input of a password derived from a Go game situation, we suggest a "Go" button, which can switch the grid to a "Go game mode". In the "Go game mode", the color of the input device will switch automatically between black and a light color once a dot is drawn, simulating the rule of Go game that stones are put on the Go board alternatively (once for a player).

### 4.5.4 Color

Although choosing colors was optional, out of the total of 167 passwords, 49 users (29.3%) chose colored passwords voluntarily. We believe this arise from the human perceptual system's tendency to favor color. Among these colored passwords, the average color-count is 1.45, meaning that we can expect one password containing 2 colors out of every two colored passwords. Theoretically the security of such a 2-color-password is increased by $8\times(3\times7)=168$ (considering that a 2-color-password must start with one (out of the eight) color code; as the average stroke-count in group 1 is $3.80\approx4$, for a colored password with the average stroke-count, i.e. 4, the second color codes must be one (out of the remaining seven) color code, and occurs right after one of the first 3

penups). This means that even without a color policy, the security of 15% passwords can be increased by about 7.4 bits ($log_2 168$).

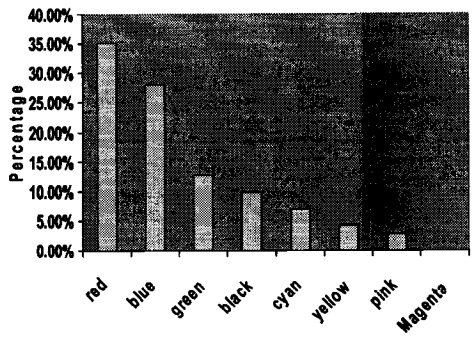



**Figure 32 Color distribution for colored passwords**

However, the color distribution was quiet uneven, as shown in Figure 32. We see that red and blue are the most often used in colored passwords, compared to pink and magenta, with a percentage of merely 2.8%, implying that our above estimate on the improvement of security by color is somewhat optimistic. Striking colors are more likely to be selected, suggesting that a better color combination strategy (such as deploying colors with similar striking index) is highly needed for optimization purposes. We leave this as an open problem.

### 4.5.5 Pattern

Passwords collected in our user study exhibit diversified patterns, as shown in Table 5. Thirty-seven percent of the passwords can be recognized or described as alphanumeric, some of which are mixed with dots. The size of the alphanumeric varies from one single cell to a 6×6 block of cells. In addition, the location and the order in which they were drawn also reflect various individual habits and interests. Twelve percent of the passwords can be categorized into well-known symbols, such as ≡, ≠, ÷, $ and so on, some of which are also mixed with dots.

It is interesting that 5 passwords (3%) were derived from Chinese characters, some of which are shown in Appendix A. We expect more passwords derived from oriental characters if Pass-Go is deployed in Korea, where most characters are composed of short straight lines and would be easy to draw on our grid. Some examples of Chinese characters are given here.

日 木 目 长 人 大 米 火 天 田　　开 白 旧 久 口 才 干 寸 土 士
斤 巾 中 央 羊 小 虫 王 于 己　　于 与 余 也 子 工 厂 生 归 不
仆 百 刁 刀 品 北 力 力 里 月　　出 明 历 呆 夫 早 上 下 左 右
手 毛 半 个 本 牛 正 化 画 时　　来 电 分 见 点 出 江 巨 用 元
旦 方 非 丰 已 古 今 向 北 立　　页 母 亩 产 又 由 友 专 会 合
可 克 舌 丌 几 只 之 内 乙 艾　　少 勺 共 公 女 问 文 风 凤 用

Our emphasis in this paper, however, is not to quantify the number of oriental characters (or symbols from other languages on the earth of which we might be unaware) that can lead to Pass-Go passwords, and the ways to draw them. We leave this as an open problem. Nineteen percent of the passwords are solely composed of dots and the rest (29%) are abstract drawings, which are difficult to for us to classify or associate with concrete meaning.

We speculate that the diversified patterns in our user-chosen passwords arise from the fact that users can draw passwords more freely than in DAS, where diagonal lines are difficult to draw. Out of 167 passwords, 52 passwords (31.1%) contain diagonal lines. If we exclude 32 passwords solely composed of dots, 38.5% of the passwords contain at least one diagonal line, implying that the diagonal line plays a significant role in memorable Pass-Go passwords.

Although there are many ways that can be suggested to create a secure password in Pass-Go, we are not attempting to give any mnemonic strategy, such as Passphrases [Yan et al. 2000]. The reason is simple: a mnemonic strategy is also a perfect guide for an attacker to build a corresponding dictionary.

Table 5 Distribution of password patterns

| | Alphanumeric | | Well known symbols | | Abstract drawings | | Dots only | Chinese characters |
|---|---|---|---|---|---|---|---|---|
| | Lines only | Mixed with dots | Lines only | Mixed with dots | Lines only | Mixed with dots | | |
| Number of passwords | 40 | 21 | 6 | 14 | 35 | 14 | 32 | 5 |
| Percentage | 24% | 13% | 4% | 8% | 21% | 8% | 19% | 3% |
| | 37% | | 12% | | 29% | | | |

### 4.5.6 Symmetric

By ignoring color, out of 167 passwords, 67 (40%) fall into $S_{1b}$ (recall §3.1), similar to the result of [Nali and Thorpe, 2004]. Such a portion of passwords could be considered as "significant", in the multi-account attack model, in which the target of an attacker is anyone out of multiple accounts, such as in the case study by [Klein 1990] (25% in that case). Therefore, assuming that DAS users draw passwords in a similar way as in Pass-Go in terms of symmetry and $L_{max}$=12, our result actually supports Thorpe and Van Oorschot's analysis [Thorpe and Van Oorschot, 2004a] that the security of DAS had been overestimated originally.

It is surprising to see that 68 passwords (41%) fall into $S_{1a}$ (recall §3.1), only one more than the number in $S_{1b}$. It appears that our users were more likely to refer to the center visible lines, rather than the other nearby axes, when drawing a symmetric password. Therefore, it would not help the attackers to consider axes other than the center ones in Pass-Go.

However, $L_{max}$ in Pass-Go has been shown by our user study to be much larger than that conjectured for DAS (40 vs. 12), implying the graphical dictionaries based on $L_{max}$=12 can only capture very short symmetric Pass-Go passwords with length 12 of less. To achieve a reasonable success as in DAS, an attacker has to extend the symmetric

graphical dictionaries significantly in Pass-Go. However, when $L_{max}$=40, the sizes of the graphical dictionaries for Pass-Go will be very large, because even for DAS-5, the size of $S_{1b}$ will grow to approximately 143 bits (the bit-size of $S_{1b}$ for DAS-5 grows with the $L_{max}$ at a rate of approximately 3.6 [Thorpe and Van Oorschot, 2004a]), which is currently too large to exhaust.

Nevertheless, a clever attacker might reduce the size of graphical dictionaries by restricting the value of $L_{max}$ to a smaller value, for example 16 (a value close to the average length in our user study) to capture a fraction of symmetric passwords. By excluding passwords longer than 16, out of 167 passwords, there are 38 (23%) passwords left in $S_{1a}$ and 37 (22%) left in $S_{1b}$.

Table 6 Bit-sizes of graphical dictionaries for DAS-5 ($L_{max}$ = 12) and Pass-Go-9 ($L_{max}$ = 16), illustrative times to exhaust, and numbers (percentages) of Pass-Go-9 user-chosen passwords captured

| Dictionary | DAS-5 ($L_{max}$ = 12) | | Pass-Go-9 ($L_{max}$ = 16) | | | |
|---|---|---|---|---|---|---|
| | Bit-size | Time to exhaust (one machine) | Bit-size | Time to exhaust (one machine) | Number of passwords captured (percentage) | |
| | | | | | Group 1 | whole |
| Full Space[3] | 57.7 | 541.8 yrs | 102* | $1.2 \times 10^{12}$ yrs | 28(61%) | 96(57%) |
| $S_{1a}$ | 48.1 | 255 days | 79.2* | $1.6 \times 10^{9}$ yrs | 8(17%) | 38(23%) |
| $S_{1b}$ | 42.7 | 6 days | 71.9* | $1 \times 10^{7}$ yrs | 8(17%) | 37(22%) |
| $S_2$ | 40.2 | 1.1 days | 57* | 334 yrs | 21(46%) | NA** |
| $S_{1b} \cap S_2$ | 30.7 | 2.1 mins | 43.4* | 9.7 days | 7(15%) | |

In order to make a reasonable comparison, here we use the sizes of graphical dictionaries for DAS-9 ($L_{max}$ = 16) as a reference, to approximate their corresponding supersets (recall §3.3) for Pass-Go-9 ($L_{max}$ = 16). Table 6 gives a comparison of the sizes of graphical dictionaries and exhaust times by one 3.2GHz Pentium™4 machine

[3] The full password space for the specific setting of $L_{max}$, as specified in each column. Therefore, in the last column, only 57% of all passwords collected in our user study fall into the full password space (when $L_{max}$=16), as the remaining passwords (43%) are longer than 16 and thus do not fall into this password space.

* Values were approximated by that of DAS-9 ($L_{max}$=16), which were provided by the second author of [Van Oorschot and Thorpe, 2005] through personal communication.

** Values are not applicable as stroke-count policies were applied in groups 1-4.

(following the same method used in [Thorpe and Van Oorschot, 2004a]) for DAS-5 ($L_{max}$=12) and Pass-Go-9 ($L_{max}$=16). Also the numbers of Pass-Go passwords which fall into the corresponding dictionaries are given.

From Table 6, we see that the sizes of graphical dictionaries for Pass-Go-9 ($L_{max}$ = 16) are much larger than that of DAS-5 ($L_{max}$ = 12), and most of them cannot be effectively exhausted currently. The smallest graphical dictionary for Pass-Go-9 ($L_{max}$=16) is $S_{1b} \cap S_2$ ($S_2$ is the subset of passwords with stroke-count $\leq$4). The size of such a dictionary is 43.4 bits ($1.2\times10^{13}$), which however is still 3.3 times the password space of 7 alphanumeric character passwords ($62^7=3.5\times10^{12}$). One 3.2GHz Pentium$^{TM}$4 machine will take 9.7 days to exhaust such a dictionary; however, this dictionary will only capture a small fraction (15%) of Pass-Go passwords in group 1.

Note that our analysis on the security of Pass-Go is conservative, because:

- 43% out of the 15% captured passwords in group 1 contain diagonal lines, which were not considered in DAS graphical dictionaries;
- We neglected color.

### 4.5.7 Starting and ending point distribution

First let us examine the distribution of starting and ending points in groups 1-4, where users can start and end their passwords arbitrarily. From Table 4 we see that 68% of the passwords in groups 1-4 start from stars or corners, much higher than the ideal percentage 9/81≈11% (if starting points are distributed in an absolutely even manner). Figure 33 shows that users tend to draw their passwords from the top left half of the grid, especially from the top left star, corner and nearby intersections. Although the ending points are distributed more uniformly than starting points (see Figure 34), and cover most parts of the grid, 48% of the passwords still end at stars or corners in groups 1-4.

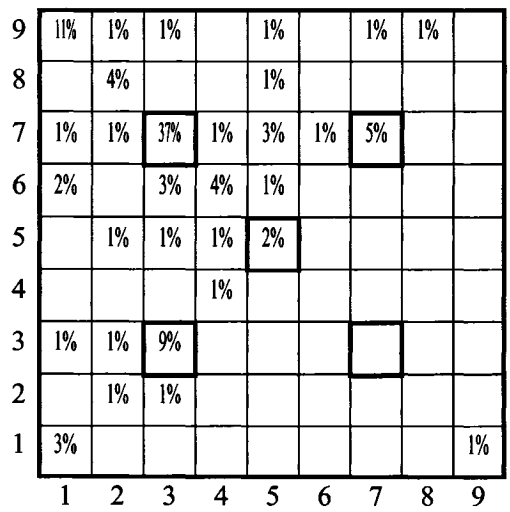



**Figure 33 Distribution of starting points in groups 1-4**

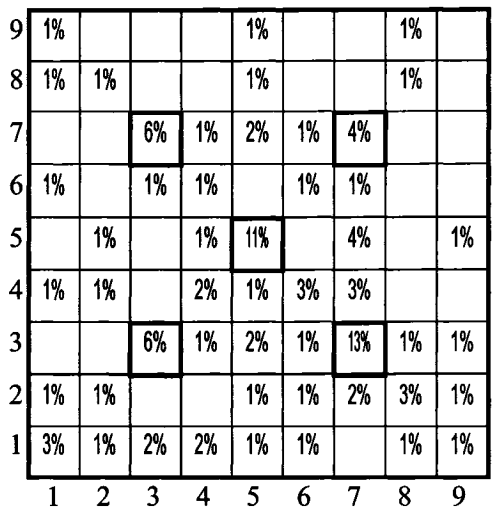



**Figure 34 Distribution of ending points in groups 1-4**

This uneven distribution of starting and ending points may be taken advantage of by an attacker to prioritize his/her dictionary. This should not be surprising, however; in an English dictionary, one will find more words starting with r, s and t than q, x and z. In Pass-Go, if an attacker narrows down his/her dictionary by restricting the starting point to (3,7), the most selected starting point, the size of his/her dictionary can be reduced by 6.3 bits ($log_2 81$), but with the cost of giving up 63% of the passwords.

Figures 34 and 35 show the distribution of starting and ending points in group 5, where we applied a dynamic password policy to disallow passwords starting from stars or corners with a probability of 89%, as stated in §4.3. Fifteen percent of the passwords start from stars or corners, much closer to the ideal percentage, implying that the distribution of starting points was controlled in a way we desired. However, we still see that most passwords start from the top left half of the grid. Although we did not apply a similar policy on the ending point in group 5, 23% of the passwords end at stars or corners, significantly lower than that of groups 1-4, implying that passwords starting from stars or corners are more likely to end at stars or corners.

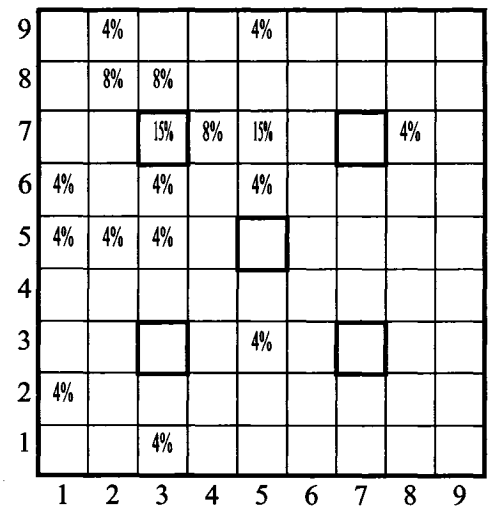



**Figure 35 Distribution of starting points in group 5**

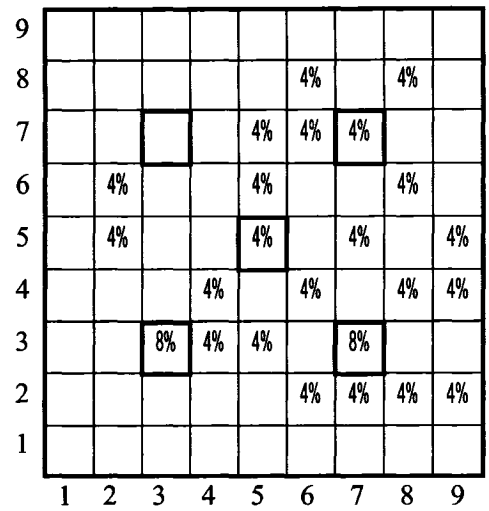



**Figure 36 Distribution of ending points in group 5**

## *4.6 Analysis on the passwords collected from the practice page*

As stated in §4.2, we set up a practice page on the user study website and encouraged Internet users who visited our website to create an account on a Pass-Go-9 in which "stars" were removed. For these visitors, we did not apply any password policy or even the requirement for the minimum password length, as we did in groups 1-5. In other words, visitors were allowed to choose any password desired, even one single dot. The FAQ page (see Appendix C) on the website was the only available source for these visitors to learn the scheme.

Over a five month period, from late August, 2005 (when the website was made available on the Internet), to late January, 2006, 57 passwords (which we refer to below as practice passwords) were collected. However, it is unknown who created these passwords, from where, at what particular time, and for what purposes. The age, race, education level, and profession of our visitors are also unknown. Nevertheless, we find that the statistics of these practice passwords are informative and very useful, and can be used as a reference for our evaluation of the security and usability of Pass-Go. In this

section, we therefore compare this group of passwords with those collected in groups 1-5, and analyze the difference between them.

There were in total 165 login attempts using practice passwords, and 99 of these login attempts were successful. The login success rate was 60%, lower than that of groups 1-5 (78%), implying the FAQ page is not sufficient for user training and confirming the effect of the tutorial given to the subjects in groups 1-5. On average there were 2.89 login attempts per user, implying these practice passwords were seldom used. It is not surprising because the website only provided very limited function (i.e., displaying the encoding of the user's Pass-Go password). Nevertheless, nine users made more than five login attempts, with a maximum of 29 times, implying that there was a small proportion of visitors who were persistent in practicing our scheme.

The lengths of the practice passwords range from 1 to 42, with an average of 11.24. This further justifies our analysis in §3.3, where we set $L_{max}$=40 and obtained an extremely large full password space. The distribution of practice password length is given in Figure 37. We see that the distribution of password length is uneven, similar to the result of groups 1-5. Thirty-six percent of the practice passwords fall into the range 1 to 5, almost the same as the percentage of the practice passwords of length of 10 or more. This confirms again that it will be effective to prioritize a graphical dictionary by the order of password length and search short passwords first. As the full password space for Pass-Go-9 when $L_{max}$=5 is only 32 bits (recall §3.3), quite exhaustible by a fast computer [Thorpe and Van Oorschot 2004a], it is reasonable that an attacker builds a simple graphical dictionary by restricting the password length to 5, if users are allowed to create passwords with arbitrary length. Therefore, the basic requirement (i.e., password length≥8) used in groups 1-5 is necessary, unless resistance to off-line attacks is not needed or users are so well educated that nobody will choose such short passwords.

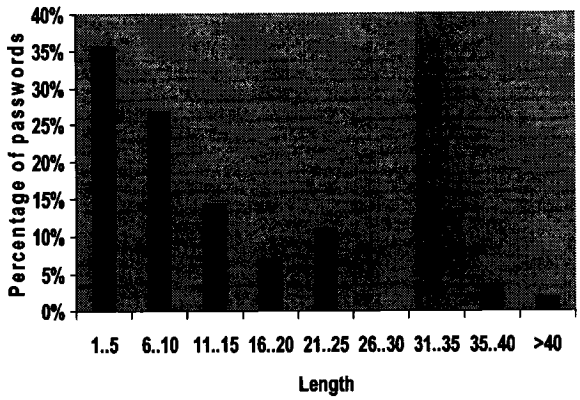



**Figure 37 Distribution of password length for practice passwords**

The stroke-counts of the practice Pass-Go passwords range from 1 to 14, with an average of 4.01, which is close to that of group 1 (3.80), but lower than that of group 5 (6.15). This suggests that the number of subjects in group 1 and 5 might be too small to provide a representative and convincing value for the average stroke-count. This shows the deficiency of our user study and suggests that a larger user study with more subjects involved is highly needed to provide a closer estimate about the average stroke-count. The distribution of stroke-count is shown in Figure 38. Twenty-six percent of the passwords contain only one stroke, similar to that of group 1 (24%), confirming that a significant fraction of users prefers to choose passwords as simple as possible when no constraint is applied or security is not important. This is also consistent with the result of our questionnaire at the end of the user study, where 80% of our users indicated that they picked a password as easy or as simple as possible. For this reason, we believe that stroke-count policies are effective measures to improve the security of Pass-Go.

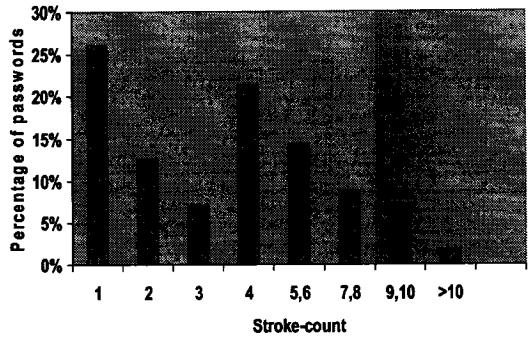



**Figure 38 Distribution of stroke-count for practice passwords**

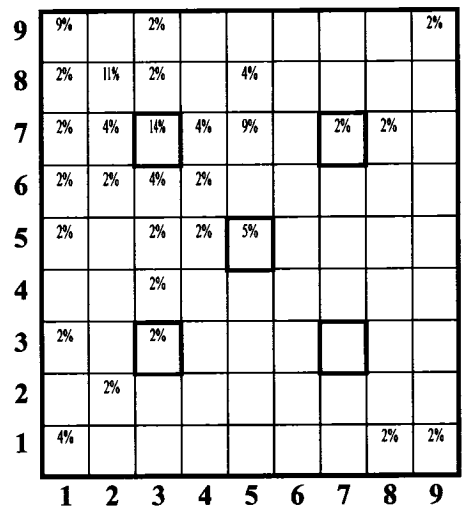



Figure 39 Distribution of starting points of practice passwords

Out of 57 practice passwords, 19 passwords (33.3%) were colored passwords (see §3.2.5), similar to that of groups 1-5 (29.3%). This confirms that a significant fraction of users will choose colored passwords voluntarily even without any requirement. Among these colored passwords, the average color-count is 1.84, higher than that of groups 1-5 (1.45). We believe this arises from the fact that users of groups 1-5 used their passwords much more frequently than the visitors; they had to consider the ease of remembering and

inputting their passwords. As clicking on a color button requires moving the mouse pointer out of the grid, and then moving back after the clicking, a colored password may take a considerably longer time to input. Blue and red were also the most selected colors, confirming the need for a better color combination strategy.

The average dot-count is 2.50, and 34 passwords (60%) contain at least one dot, higher than that of groups 1-5 (48.5%). This confirms that many Pass-Go passwords will escape attention if an attacker excludes all the passwords which contain at least one dot.

Passwords collected from the practice page also exhibit diversified patterns, as shown in Table 7. Sixteen percent of the passwords can be recognized or described as alphanumeric, some of which are mixed with dots. The size of the alphanumeric varies from one single cell to an 8×8 block of cells. In addition, the location and the order in which they were drawn also exhibit various patterns. Eighteen percent of the passwords can be categorized into well-known symbols, such as ┐, ], ◊, and so on, some of which are also mixed with dots. Forty percent of the passwords are solely composed of dots and the rest (26%) are abstract drawings, which are difficult for us to classify or associate with concrete meaning.

Out of 57 passwords, 14 passwords (24.6%) contain diagonal lines, similar to the result of groups 1-5 (31.1%). If we exclude 23 passwords solely composed of dots, 41.2% of the passwords contain at least one diagonal line, also close to the result of groups 1-5 (38.5%). This confirms our analysis in §4.5.5 that the diagonal line plays a significant role in memorable Pass-Go passwords.

Out of 57 practice passwords, 21 passwords (37.5%) fall into $S_{1b}$, similar to the result of groups 1-5 (40%). The same 21 passwords fall into $S_{1a}$, confirming that Pass-Go users are more likely to refer to the center visible grid lines, rather than the other nearby axes when drawing a symmetric password. Therefore, it would not help the attackers to consider axes other than the center ones in Pass-Go.

Table 7 Distribution of practice password patterns

| | Alphanumeric | | Well known symbols | | Abstract drawings | | Dots only |
|---|---|---|---|---|---|---|---|
| | Lines only | Mixed with dots | Lines only | Mixed with dots | Lines only | Mixed with dots | |
| Number of passwords | 6 | 3 | 7 | 3 | 10 | 5 | 23 |
| Percentage | 11% | 5% | 13% | 5% | 18% | 9% | 40% |
| | 16% | | 18% | | 26% | | |

Out of 57 practice passwords, 22 passwords (39%) start from stars or corners, much lower than that of groups 1-4 (68%), where no starting point policy was applied as well. More precisely, 13 practice passwords (23%) start from stars, substantially lower than that of groups 1-4 (51%). It appears that removing stars from the grid effectively results in the decrease of practice passwords starting from stars. In other words, "stars" do attract users to start their passwords from there. Nine practice passwords (16%) start from corners, close to that of groups 1-4 (15%), implying that removing the stars has no impact on the percentage of the passwords which start from corners.

Meanwhile, 16 passwords (28%) end at stars or corners, lower than that of groups 1-5 (44%). It appears that users tend to end their passwords at stars when they are available. To make the distribution of starting and ending points more even, it would be effective to remove the stars from the grid, with the cost of losing some degree of memorability.

# Chapter 5 Memorable password space analysis

We have shown in §4.5.6 that Pass-Go offers significant resistance over the DAS scheme to symmetric graphical dictionary attacks suggested by Thorpe and Van Oorschot [Thorpe and Van Oorschot 2004a; 2004b; Van Oorschot and Thorpe 2005]. This leads us to ask if a clever attacker is able to build a smaller graphical dictionary and achieve comparable success.

According to the definition of $S_{1b}$ (recall §3.1), any dot on the center (vertical and/or horizontal) grid lines is considered as a mirror symmetric stroke. Therefore, a password composed of 12 such dots will fall into $S_{1b}$. If these dots are clicked randomly on the center grid lines, apparently the resulting passwords are not memorable for most people. For this reason, we believe that the size of the graphical dictionaries can be further reduced.

On the other hand, we find that there are many memorable passwords which cannot be captured by the graphical dictionaries suggested by Thorpe and Van Oorschot. For instance, Pass-Go passwords derived from alphanumeric or well-known symbols make up 49% of the user-chosen passwords in our user study. Although some of them were drawn in unpredictable ways or mixed with dots, many of these passwords may still be effectively guessed. For example, a password derived from the English letter "S" is not a symmetric password, but obviously it is predictable. In other words, the graphical dictionaries should be expanded.

To claim that Pass-Go offers stronger resistance to off-line dictionary attacks than textual passwords, the memorable password space should be quantified. Our following discussion in this section is to quantify two small subsets of the memorable password space of Pass-Go, and show that they are significantly larger than that of the textual passwords.

The methodology we use is borrowed from [Jermyn et al. 1999], where user choices were modeled and then the sizes of subsets of the memorable password space were quantified.

First, following the same calculation used in [Jermyn et al. 1999], drawing merely two rectangles in Pass-Go-9 can produce approximately $4.3\times10^8$ different passwords, which is much larger than that for DAS-5 ($2.6\times10^6$). The cardinality of such a simple graphical dictionary is also significantly larger than that of the textual dictionary used in Klein's case study [Klein 1990], in which about 25% of 14,000 passwords were cracked by a dictionary with only 3 million entries. If such a graphical dictionary is used, by ignoring color as well, 4 user-chosen passwords (2.4%) in groups 1-5 will be captured, which is far lower than that of textual passwords.

Second, we assume that passwords composed of one or two alphanumeric or well-known symbols and drawn in predictable ways are memorable. We notice that different symbols can result in the same drawing (i.e. the same password encoding). For example, Pass-Go passwords derived from the letter "S" and the number "5" can be actually the same. Therefore, we only consider one of these overlapping symbols. By excluding overlapping symbols, 7 numbers, 39 English upper or lower-case letters, and 64 well-known symbols are selected as the prototypes of our memorable passwords.

We define $N_r$ as the number of rectangles in which an alphanumeric or well-known symbol can be held. The value of $N_r$ varies for different alphanumeric or well-known symbols. We also define the width of a rectangle as $w$, and the height as $h$. For example, for a single cell, both its width and height are 1, namely $w=1$ and $h=1$.

As any rectangle on the grid can hold a letter "o" or "c", $N_r$ =1296 ( $C_9^2 \times C_9^2$ ) for "o" and "c"; only squares ($h=w$) can hold letter "X" or "N", due to the diagonal lines contained in such letters, resulting in $N_r$=204. In order to hold a letter "A", the height of the rectangle should be greater than one ($h>1$), and "I" must be held in a rectangle whose width is even ($w=2\times n$ and $n$ is an integer greater than zero). According to the different

values of $N_r$, we could classify the alphanumeric and well-known symbols into roughly 10 groups, as shown in Table 8. The symbols in each group have the same requirements for the rectangles to hold them. In other words, the symbols in each group have the same value of $N_r$.

For each group, we count the number of the corresponding alphanumeric or well-known symbols as $N_m$. For example, there are four symbols in the second group, resulting in $N_m$=4. We also define $N_w$ as the number of ways in which the symbols could be drawn. For example, $N_w$=2 for "|", as "|" can be drawn in two different ways (from top to bottom or from bottom to top), resulting in two different password encodings[4]. Also there are four ways[5] to draw the character "+", but we estimate 2 as a reasonable value, as most people write "-" from left to right and "|" from top to bottom. Similarly, there are 12 ways to draw "÷", but we only estimate 4 as a reasonable value. The reason we estimate in this conservative way is to guarantee that every password we count is reasonably memorable (i.e., they are likely to be chosen by users in practice), and to avoid overestimating the size of the memorable password space of Pass-Go.

In order to prevent the size of this subset of Pass-Go passwords from becoming too large, we only consider the predictable ways in which the symbols are likely to be drawn. For example, the passwords shown in Figure 40 are not considered as predictable, and are excluded from this subset of memorable passwords, although they have the shape of letter "H" and "A" and might be memorable for some people. For the letter "H", we think a predictable way is to draw the stroke "—" in the exact middle of the two vertical lines; for the letter "A", we expect the stroke "—" to touch the two vertical lines. Otherwise, as any character can be drawn theoretically in a large number of ways, we cannot obtain an effective dictionary.

[4] Actually the number "1" can be drawn in many other ways. For example, a user can break the password into two or more strokes.
[5] There might be many more ways if the password is broken into many strokes.

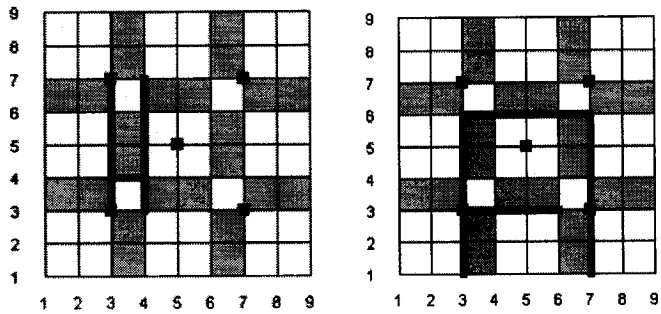


**Figure 40 Pass-Go passwords drawn in unpredictable ways**

Then we sum the number of memorable passwords in each group, and the result is shown in Table 8. We see that the size of the subset of passwords, which contain one or two alphanumeric or well-known symbols and are drawn in predictable ways, is $6.2\times10^{10}$ (35.9 bits), smaller than the size of $S_{1b} \cap S_2$ (recall §4.5.6) when $L_{max}$=16 ($2^{43.4}=1.2\times10^{13}$), but is significantly larger than the size of the dictionary used in Klein's survey [Klein 1990] for textual passwords ($3\times10^{6}$). If an attacker uses such a graphical dictionary, by ignoring colors, 32 passwords (19%) in groups 1-5 can be captured, more effectively than using $S_{1b} \cap S_2$. Therefore, from an attacker's point of view, this method is more efficient than the methods suggested by Thorpe and Van Oorschot [Thorpe and Van Oorschot 2004a; 2004b; Van Oorschot and Thorpe 2005].

We believe the size of this subset of passwords can be further reduced by excluding some symbols which might be less likely to be chosen than others, or restricting the size of rectangles and ways in which passwords are drawn, with the cost of losing some number of passwords. This should be done by conducting larger scale user studies on the patterns of Pass-Go passwords. We leave this as an open problem.

Table 8 Memorable passwords calculation

| Requirements of rectangles | Alphanumeric or well-known symbols | $N_m$(Number of Alphanumeric or well-known symbols) | $N_w$ (Number of ways to draw) | $N_r$ (Number of rectangles can be held) | Number of passwords |
|---|---|---|---|---|---|
| | . | 1 | 1 | 81 | 81 |
| *w≥1 or h≥1* | \| - : .. | 4 | 2 | 324 | 2592 |
| *w≥1 and h≥1* | a c o n U u L Q = ] 7 ⌐ ⌟ ‖ | 14 | 4 | 1296 | 72576 |
| *w=2h or h=2w* | M W V ∑ ^ < > ; ▽ △ ◁ ▷ | 12 | 4 | 114 | 5472 |
| *h=w* | X Z N y , % * ↖ ↗ ↘ \ ↙ « » t f ± + / ↘ ↗ ∠ ⌥ ◸ ◺ ◺ ◿ ◹ ◸ ◸ ◹ | 31 | 4 | 204 | 25296 |
| *w=2n* | I T m | 3 | 4 | 576 | 6912 |
| *h>1* | ! i | 2 | 4 | 252 | 2016 |
| *(h>1and w≥1) or (w>1 and h≥1)* | A B E F G H J K P R S b d e g h j k 2 3 4 6 8 ɳ ≡ ↑ → ↓ ← | 29 | 4 | 1008 | 116928 |
| *h>1 and w=2n* | Y ⌂ ÷ ∴ ∵ | 5 | 4 | 448 | 8960 |
| *(h>1 and w=1) or (w>1 and h=1)* | R l ≤ ≥ ↿ ↾ ⇂ ⇀ ↼ ⍐ | 10 | 4 | 224 | 8960 |
| | | | | | |
| Total number of memorable passwords which contain one symbol | | | | | $2.5\times10^5$ |
| Total number of memorable passwords which contain two symbols | | | | | $6.2\times10^{10}$ |

# Chapter 6 Variations based on Pass-Go

In this chapter, we propose some variations on Pass-Go, which further explore the precious space resource on the grid, and offer either better usability or stronger security. Some of the variations are more complicated than the basic Pass-Go scheme, and might lead to increased user training and support cost.

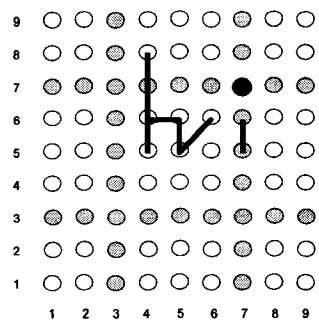



**Figure 41 PassCells**

## *6.1 PassCells*

One feature of the Pass-Go scheme is that sensitive areas are invisible; therefore a user will not know if he/she has successfully selected an intersection or not until the dot or line indicator appears. Users might have to spend a period of time practicing before they can draw lines comfortably without making unintentional errors (e.g., touching neighbor intersections).

We propose PassCells to address this problem. Instead of showing a grid, a matrix of cells is presented on the display, as illustrated in Figure 41. The shape and size of the cells can be predefined. The matrix of cells works in the same way as the sensitive areas in Pass-Go. A user is required to select one or more cells to authenticate a system. A dot indicator appears when one cell is touched separately, and line indicators display when

two or more cells are touched continuously. A line indicator will be drawn from the center of the first selected intersection to the center of the second selected intersection. The dot indicator can have the same shape and size as the cells. From the users' point of view, when a cell is clicked, it is filled with the color of the input device.

This change makes the boundary of sensitive areas visible to users, and a user can see exactly where he/she should touch; this might ease the task of inputting a password for some users. Based on the feedback from family, friends, and classmates, I found that different people had different preferences. Some people indicated that they liked Pass-Go more as the visible grid lines make drawing a line easier and less error prone; however, others expressed that they preferred PassCells because it is clearer where they should start or stop a stroke. We believe the preference can also be influenced by the cultural backgrounds of users. To the best of my knowledge, the intersections of a grid are used to place chessmen or "stones" in most Chinese games; while in the west, the cells are more often used. This somehow reflects the difference of thinking between the east and west. Therefore, it might be possible that more users in China prefer Pass-Go, while PassCells is more welcome in North America. This question can only be answered based on a large user study in which multiple user groups representing different cultural backgrounds participate. We leave this as an open problem.

The major drawback of PassCells is that it is not scalable, as reference aids are difficult to deploy. For this reason, we only suggest it to be used for a small matrix size, such as 5×5 or 7×7. Another drawback is that users cannot draw lines along the visible grid lines; this might reduce the usability of the scheme to some extent. In addition, variations like curved lines (see §6.3) are difficult to be combined into PassCells. Moreover, this scheme has nothing to do with the Go game; therefore deriving a password based on a Go game situation is not possible.

## 6.2 Cell indicators

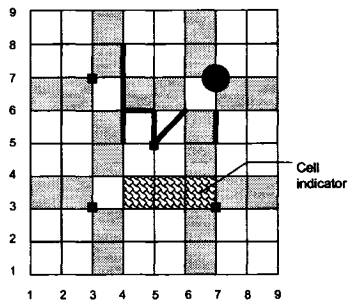



**Figure 42 Cell indicators**

Besides dot and line indicators, we introduce cell indicators. The right button of a mouse may be used to choose cells in the grid. If the input device is a light pen, we can define that pressing the shift key (or space key) changes the light pen to a "right button mode". The corresponding indicators could be patterned cell indicators as shown in Figure 42.

One drawback of cell indicators is that the login interface might look messy, as the shaded cells and cell indicators might overlap each other. One solution for this problem is to remove shaded cells. However, cell indicators might still overlap with dot indicators. Therefore, users have to be educated to avoid this kind of password. Using only stars as reference aids will also reduce the scalability of the scheme. Therefore, we do not recommend cell indicators to be used for a large grid size.

The Cell indicator itself can actually be thought of as a variation of the DAS scheme. Instead of showing the trace of the movement of the input device, cell indicators show the cells that correspond most closely with the input trace.

## 6.3 Curved line indicator

We define a cell center as an area surrounding the center of each cell in a grid, as shown in Figure 43. The size of the cell center can be adjusted. In our implementation the radius of the cell center is the same as that of the sensitive area, ¼×*d* (where *d* is the side length of a grid cell). Similar to sensitive areas, the cell centers are invisible to users.

Curved line indicators can be displayed when the input device passes through or around the cell centers. For example in Figure 43, to draw the curve on the left, we could press the left button of the mouse on intersection (2,5), then keep holding it while dragging to (3,5) through the corresponding cell center. To draw a curve on the right, we could press the left button of the mouse on intersection (5,5), then keep holding it while dragging to (6,6) without passing the corresponding cell center. In this way, we can derive more passwords, which contain curved line indicators, as shown in Figure 44. By losing some degree of ease in inputting straight lines, this feature might be preferable in eastern Asian countries, where simple curved lines are the basic strokes in their characters.

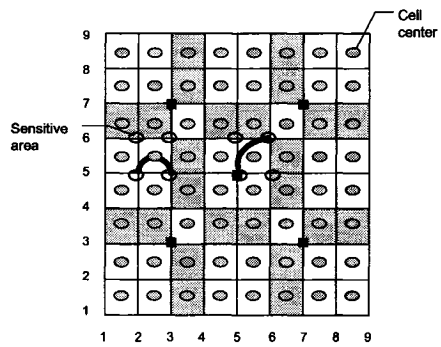



**Figure 43 Curved line indicators**

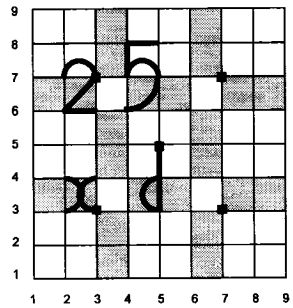



**Figure 44 Example Password with curved lines**

## *6.4 Short line indicator*

We assumed before that all the pendowns (pressing down the left button of a mouse or putting a light pen down onto the surface of the display) and penups occur inside a sensitive area. What if they occur somewhere outside the sensitive areas?

We introduce short line indicators at either the beginning or the end of a stroke, which can only be 1/3 the length of a normal line indicator unit, as shown in Figure 45. The short line indicators can be drawn horizontally, vertically, or diagonally, depending on the direction in which the stroke starts or ends. For example, to draw the last stroke for "h" in Figure 45, one needs to move the input device out of the sensitive area of the intersection (4,4) diagonally towards top-right, and release the left button of a mouse or lift the light pen without touching any other neighbor intersections (to differentiate a line indicator).

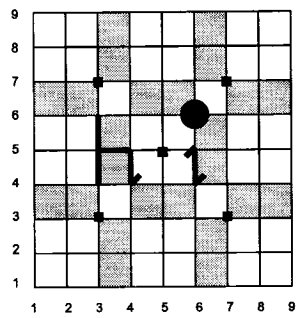



**Figure 45 Short line indicators**

The advantage of short line indicators is that they expand the password space; therefore this variation of Pass-Go can provide stronger security. The drawback of short line indicators is that they require the grid cells to be sufficiently large. If the grid cells are too small, it will be difficult for a user to draw short line indicator without touching neighbor sensitive areas.

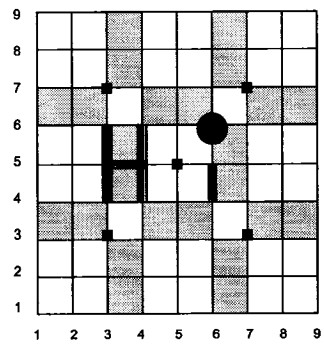



**Figure 46 Patterned line indicators**

## 6.5 Patterned indicator

To strengthen the security of Pass-Go, we introduce patterned line indicators, as shown in Figure 46. Before drawing a line, the user can choose the pattern (or style) of the line indicator. Line patterns can be switched by clicking pattern buttons (like color buttons), which can be deployed surrounding the grid. The default can be set as the regular thick line (see Figure 19). Based on the same idea, patterned dot indicator can be designed to further expand the password space of Pass-Go.

## 6.6 Penup gap

One problem of Pass-Go is that the final shape of a password cannot provide sufficient information about how the password is actually drawn. For example, a straight line can be drawn in one stroke or in multiple strokes, but their final shapes can be exactly the same. Similarly, the direction in which a stroke is drawn can also result in different password encodings. A user has to remember this information (the order and direction) completely by heart. The complexity of remembering a password is high, making (especially new) users confused, as discovered by our user study (recall §4.4.1).

In order to make the final shape of a password more informative, we suggest to use penup gap. Penup gap is a small gap which is displayed at the end of a stroke (i.e., at the place where a penup occurs). For example in Figure 47, the encoding of the password is

(4,8), (4,7), (0,0), (4,7), (4,6), (0,0), (4,6), (4,5), (0,0), (4,6), (5,6), (5,5), (6,6), (0,0), (7,7), (0,0), (7,6), (7,5),(0,0).

The long vertical line for the letter "h" is actually drawn in 3 strokes, as indicated by the penup gaps in Figure 47. As a penup gap also indicates the end of a stroke, the direction in which a line is drawn can thus be deduced by users.

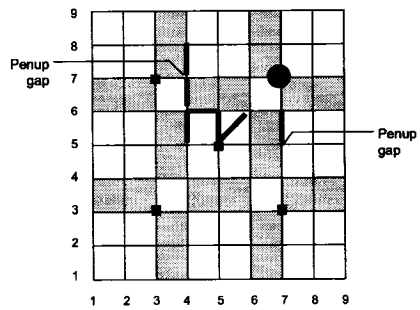



**Figure 47 Penup gap**

## *6.7 Directional indicator*

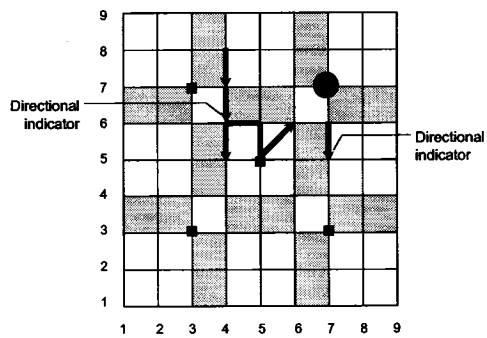



**Figure 48 Directional indicator**

An alternative for penup gap is directional indicator, as shown in Figure 48. Instead of a small gap, an arrow (or gradually narrowed line) is shown to indicate the direction in which the stroke is drawn and where penups occur.

## 6.8 Sequence number

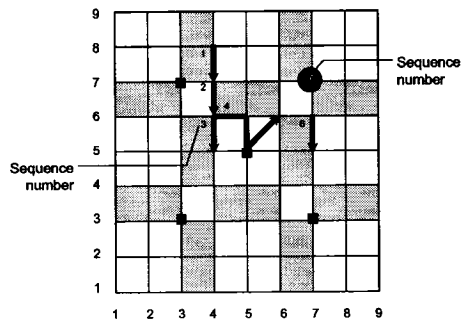



**Figure 49 Sequence number**

To further facilitate the reading of the final shape of a Pass-Go password, we introduce sequence numbers to indicate the order in which the password strokes are drawn. Sequence numbers can be placed near the starting points of line indicators or inside dot indicators, as shown in Figure 49. By combining sequence numbers and directional indicators (or penup gaps), the final shape of a password can provide a clear view of how the password is actually drawn.

Sequence numbers and directional indicators (or penup gaps) can work as options or tools, which can be enabled or disabled by a user according to his/her particular need at run time.

# Chapter 7 Conclusions and future work

We have presented a new graphical password scheme and shown that it keeps most of the advantages of the DAS scheme and offers stronger security and better usability. We conducted an informal user study on Pass-Go and provided detailed statistics about the characteristics of user-chosen passwords. The most important among them is that users tend to choose very long passwords in our scheme, leading to an extremely large password space. We applied current available techniques to reduce the size of the small graphical dictionaries and see that even with the strictest conditions, the size of the graphical dictionary is still 3.3 times the password space of 7 alphanumeric character passwords, and can at most capture a small fraction (15%) of Pass-Go passwords. We compared the impact of stroke-count policies on the user choices, and tested our dynamic password checking method. We quantified the memorable password space by modeling user's choice. The size of the memorable password space of Pass-Go was shown to be much larger than that of textual passwords.

Our contributions also include the following: a new categorization of graphical password schemes; the introduction of reference aids; an efficient and human readable encoding scheme; identification of the need and a solution for keyboard input support; two solutions for the shoulder surfing problem; a dynamic password checking method; and several variations on the basic scheme.

Future work should be directed toward optimizing the Pass-Go scheme: exploring the feasibility and usability of Pass-Go-11, 13 or even 15; designing more helpful referencing aids; finding the best size for sensitive areas or cell centers (for curved line indicators); setting up better color combination strategies; looking for better solutions for the shoulder surfing problem, and so on. Memorable passwords derived from oriental or other characters should be particularly studied and quantified. A user study in which users make more effort to create secure passwords would be helpful to give a closer estimate of the security of Pass-Go, rather than the conservative estimate given in this paper. An extensive user study based on the DAS scheme will provide a better comparison between

these two schemes. Moreover, user studies on the various Pass-Go variations, based on different grid sizes, with various input or display devices, in a user group representative of the general population, would be helpful to compare the security and usability between them.

# References


ATTNEAVE, F. 1955. Symmetry, Information and Memory Patterns. *American Journal of Psychology 68*, 209-222.

BIRGET, J., HONG, D., AND MEMON, N. 2003. Robust discretization, with an application to graphical passwords. *Cryptology ePrint Archive*, Report 2003/168. http://eprint.iacr.org/2003/168, last accessed on Jan 29, 2006.

BLONDER, G. 1996. Graphical passwords. United States Patent 5559961.

BOWER, G. H., KARLIN, M. B., AND DUECK, A. 1975. Comprehension and memory for pictures. *Memory and Cognition 3,* 216-220.

BROSTOFF, S. AND SASSE, M. A. 2000. Are Passfaces™ more usable than passwords? A field trial investigation. In *Proceedings of Human Computer Interaction*, pages 405–424, 2000.

DAVIS, D., MONROSE, F., AND REITER, M. K. 2004. On User Choice in Graphical Password Schemes. In *Proceedings of the 13th USENIX Security Symposium.* 151-164.

DHAMIJA, R. AND PERRIG, A. 2000. Déjà Vu: A User Study Using Images for Authentication. In *Proceedings of the 9th USENIX Security Symposium.*

FELDMEIER, D. AND KARN, P. 1989. UNIX password security-Ten years later. In *Proceedings of the 19th International Conference on Advances in Cryptology (CRYPTO '89).* Lecture Notes in Computer Science, vol. 435. Springer Verlag.

GOLDBERG, J., HAGMAN, J., AND SAZAWAL, V. 2002. Doodling our way to better authentication. *Conference on Human Factors and Computing Systems, Poster Session: Student Posters,* 868-869.

HONG, D., MAN, S., HAWES, B., AND MATHEWS, M. "A password scheme strongly resistant to spyware," In *Proceedings of International conference on security and management.* Las Vergas, NV, 2004.

HONG, L., AND JAIN, A. 1997. Integrating faces and fingerprints for personal identification. *Lecture Notes in Computer Science,* (1351), 1997.

INTRAUB, H. 1980. Presentation rate and the representation of briefly glimpsed pictures in memory. *Journal of Experimental Psychology: Human Learning and Memory,* 6(1):1–12.

JANSEN, W., GAVRILA, S., KOROLEV, V., AYERS, R., AND SWANSTROM, R. 2003. Picture Password: A Visual Login Technique for Mobile Devices. *NIST Report - NISTIR7030.*

JERMYN, I., MAYER, A., MONROSE, F., REITER, M. K., AND RUBIN, A. D. 1999. The Design and Analysis of Graphical Passwords. In *Proceedings of the 8th USENIX Security Symposium.*

KLEIN, D. 1990. Foiling the cracker: A survey of, and improvements to, password security. In *Proceedings of the 2nd USENIX Security Workshop.* 5-14.

MAN, S., HONG, D., AND MATHEWS, M. 2003. "A shoulder surfing resistant graphical password scheme," in *Proceedings of International conference on security and management.* Las Vegas, NV, 2003.

MONROSE, F. AND REITER, M. K. 2005. Graphical passwords. *Security and Usability,* L. Cranor and S. Garfinkel, Eds.O'Reilly, Chapter 9, 147–164.

MORRIS, R. AND THOMPSON, K. 1979. Password security: A case history. *Communications of the ACM (11),* 594-597.

NALI, D. AND THORPE, J. 2004. Analyzing user choice in graphical passwords. *Technical Report TR-04-01, Carleton University, Canada.*

OPENWALL PROJECT. 2006a. John the ripper password cracker. http://www.openwall.com/john/, site accessed on Jan 29, 2006.

OPENWALL PROJECT. 2006b. Wordlist. http://www.openwall.com/passwords/wordlists, site accessed on Jan 29, 2006.

PAIVIO A., ROGERS, T. B., AND SMYTHE, P. C. 1968. Why are pictures easier to recall than words? *Psychonomic Science*,11:137-138, 1968.

PASSFACES. 2006. The science behind passfaces™ for windows. http://www.realuser.com/resources/white%20papers.htm, site accessed on Jan 29, 2006.

PASSLOGIX. 2006. www.passlogix.com, site accessed on Jan 29, 2006.

PAULSON, L.D., Taking a graphical approach to the password, *Computer* 35 No.7 19-19, IEEE Computer Society. 2002

PERRIG, A. AND SONG, D. 1999. Hash Visualization: a New Technique to Improve Real-World Security. In *International Workshop on Cryptographic Techniques and E-Commerce*, pages 131–138, 1999.

SFR. 2006. www.viskey.com/tech.html, site accessed on Jan 29, 2006.

SOBRADO, L. AND BIRGET, J. 2002. Graphical Passwords. *The Rutgers Scholar, Rutgers University, Camden New Jersey 08102 4.*

STANDING, L. 1973. Learning 10,000 pictures. *Quarterly Journal of Experimental Psychology,* 25:207-222, 1973.

STUBBLEFIELD, A. AND SIMON, D. R. 2004. Inkblot Authentication. *Microsoft Technical Report MSR-TR-2004-85.*

SUMMERS, W. C. AND BOSWORTH, E. 2004. Password policy: the good, the bad, and the ugly. In *Proceedings of the winter international synposium on Information and communication technologies.*

SUO, X., ZHU, Y., AND OWEN, G. S. 2005. Graphical Passwords: A Survey. In *21st Annual Computer Security Applications Conference (ACSAC)* (December 5-9).

TAKADA, T. AND KOIKE, H. 2003. Awase-E: Image-based Authentication for Mobile Phones using User's Favorite Images. In *Human-Computer Interaction with Mobile Devices and Services*, vol. 2795 / 2003: Springer-Verlag GmbH, 2003, pp. 347 - 351.

THORPE, J. AND VAN OORSCHOT, P. C. 2004a. Graphical Dictionaries and the Memorable Space of Graphical Passwords. In *Proceedings of the 13th USENIX Security Symposium.* 135-150.

THORPE, J. AND VAN OORSCHOT, P. C. 2004b. Towards Secure Design Choices For Implementing Graphical Passwords. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC),* Tucson, USA. December 2004.

USGO. 2006a. Top Ten Reasons to Play Go. http://www.usgo.org/resources/topten.asp, site accessed on Jan 20, 2006.

USGO. 2006b. A Very Brief History of Go. http://www.usgo.org/resources/gohistory.asp, site accessed on Jan 20, 2006.

VAN OORSCHOT, P. C. AND THORPE, J. 2005. On the Security of Graphical Password Schemes. *Technical Report TR-05-12, Carleton University, Canada.*

WIEDENBECK, S., WATERS, J., BIRGET, J., BRODSKIY, A., AND MEMON, N. 2005. PassPoints: Design and longitudinal evaluation of a graphical password system. *International J. of Human-Computer Studies (Special Issue on HCI Research in Privacy and Security)* 63, 102–127.

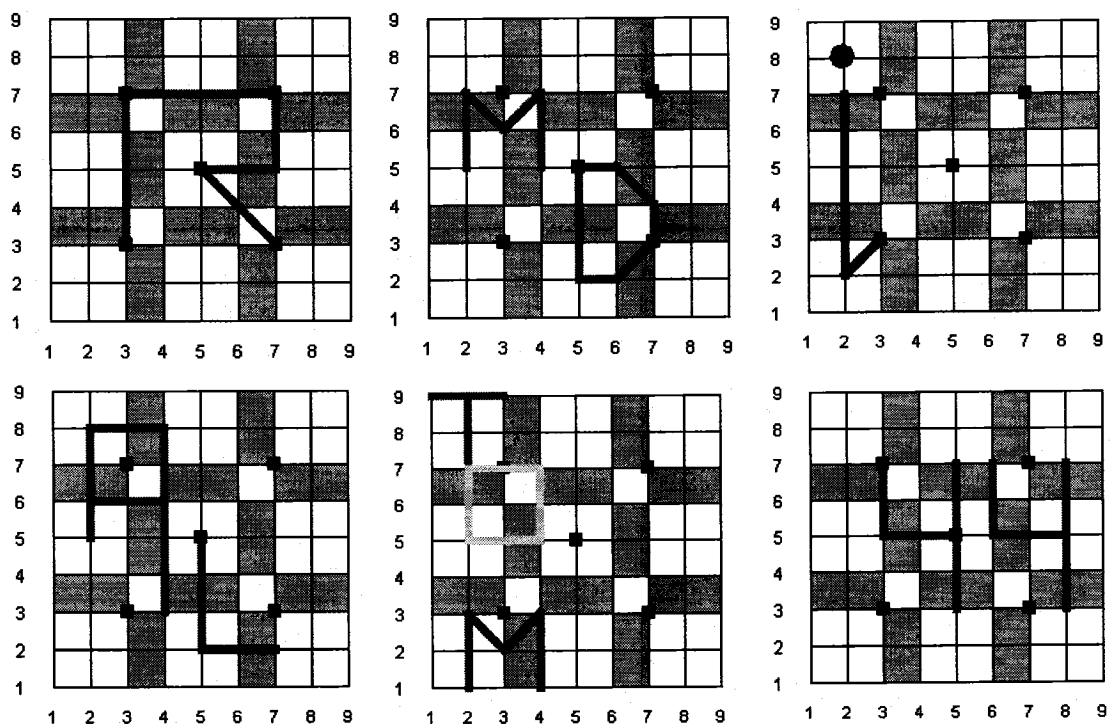
WU, T. 1990. A real-world analysis of Kerebros password security. In *Proceedings of the 1999ISOC Symposium on Network and Distributed System Security*. 9, vol. (8). 723-736.

YAN, J., BLACKWELL, A., ANDERSON, R., AND GRANT, A. 2000. "The Memorability and Security of Passwords – Some Empirical Results", *Technical Report No. 500, Computer Laboratory, University of Cambridge.*

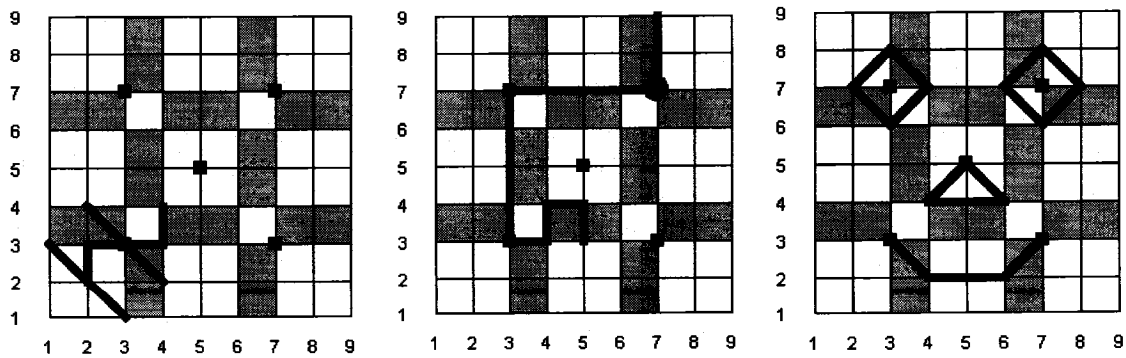
YAN, J. 2001. A Note on Proactive Password Checking. *ACM New Security Paradigms Workshop*, New Mexico, USA.

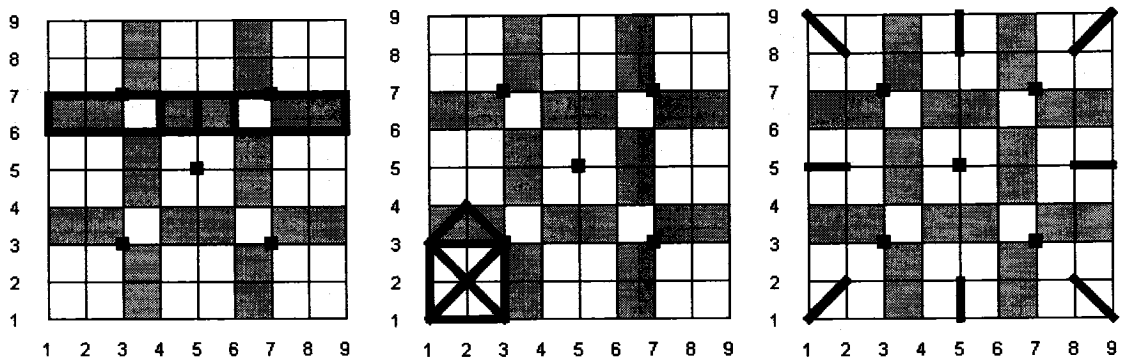
# Appendix A Sample user-chosen passwords

Alphanumeric:

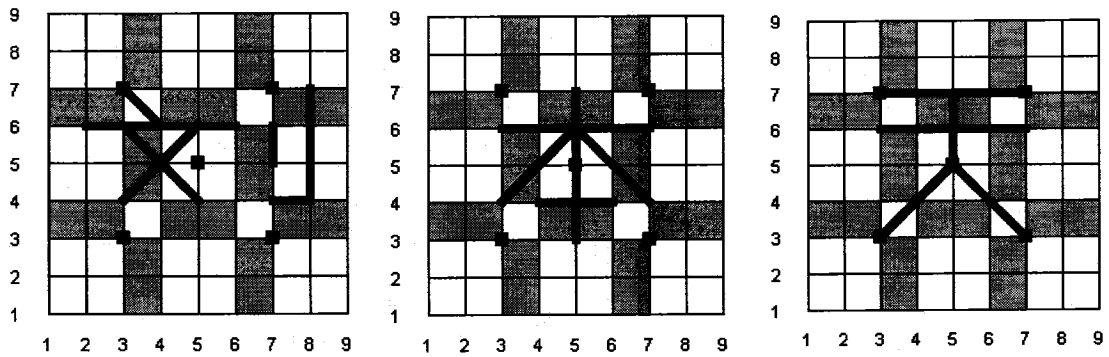


Abstract drawing:

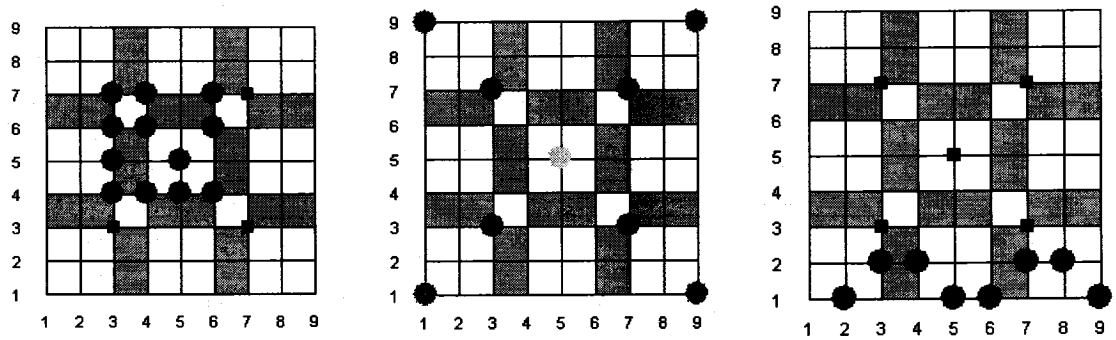




Chinese character:



Solely dots:



# Appendix B Improvements over multiple-image schemes

As discussed in §2.1.2, multiple-image schemes exploit human's significant capability of recognizing visual images. However, we find that the current multiple-image schemes can be further improved and optimized. In this section, we discuss several improvements over the multiple-image schemes. (Note that formal user studies, conducted using significant numbers of subjects, would need to be done to confirm whether the proposed schemes are indeed improvements, and to quantify how effective they are.)

- ***Shaped image scheme***

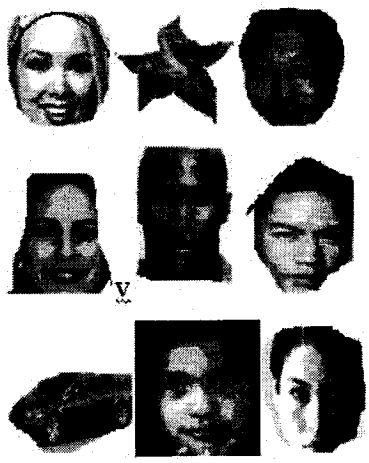


**Figure 50 Shaped image scheme**

Similar to Passfaces™ [Passfaces 2006] and the Story scheme [Davis et al. 2004], one or more previously seen images are required to be recognized and identified among other decoy images. The difference is that the images are designed in different shapes, as shown in Figure 50. For example, some images have a square shape, and some have a circle shape. Other shapes (e.g., triangle, diamond, oval, or parallelogram) can also be used. These shapes provide additional information to users. When a password needs to be input, more information can be used by a user to differentiate images.

- ***Colored image scheme***

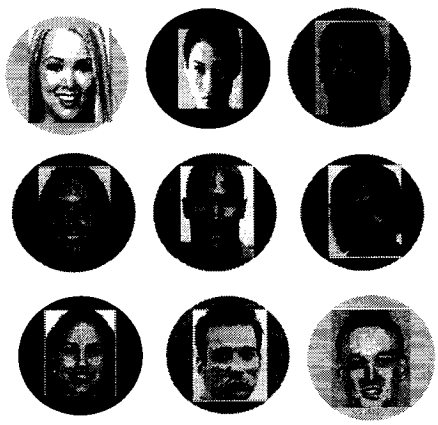


**Figure 51 Colored image scheme (a)**

Images are presented with different color backgrounds, as shown in Figure 51. For example, some images have a blue background, and some have a red background. These colored backgrounds can also provide useful information to users. A slight variation is to replace colored backgrounds with colored rings, as shown in Figure 52.

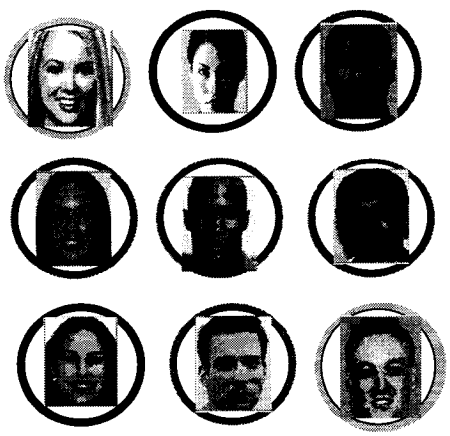


**Figure 52 Colored image scheme (b)**

- ***Grouped image scheme***

In this scheme, we suggest classify images into different theme groups. In the password initialization stage, the user chooses his/her favorite theme. Then the user selects one or more images from the scheme group, and makes a story to remember the password. When the user authenticates a system, pass-images and decoy images from the chosen theme group are presented. Because the user is familiar with this theme, making a story and differentiating images in this topic is easier.

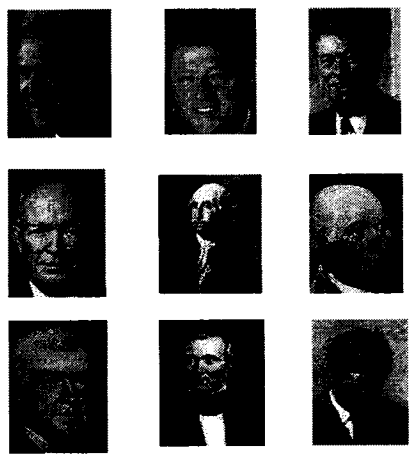


**Figure 53 Grouped image scheme (American presidents)**

For example, one user who is interested in the history of the United States might choose a theme of "American presidents". He can make a story based on his knowledge in history, and the login interface will look like Figure 53. Another user might be a big movie fan, and he/she will probably choose "Oscar stars" as his/her theme, and the login interface will look like Figure 54.

**Figure 54 Grouped image scheme (Oscar stars)**

This scheme takes advantage of people's existing knowledge in different fields. A sequence of American presidents might look random for some people, but might be meaningful for someone who chooses this theme. Another advantage is that this scheme is more pleasant, as users only see the images they like.

# Appendix C A method to approximate the full password space of Pass-Go

For Pass-Go-$G$ and colored Pass-Go-$G$, we give a method to approximate the full password space.

## For Pass-Go-G

Our approximation method is based on the following observations:

- Based on any password with length of $L$, adding one dot ($G^2$) or extending the last stroke by one unit in each direction available (the least 3 and the most 8), could derive a new password with length of $L$+1.
- There are $G^2$ passwords when $L_{max}$=1.

Therefore, the lower bound of the full password space for Pass-Go-$G$ will be

$$\sum_{i=1}^{L_{max}} G^2 \times (G^2+3)^{i-1}$$

and the upper bound of the full password space for Pass-Go-$G$ will be

$$\sum_{i=1}^{L_{max}} G^2 \times (G^2+8)^{i-1}$$

When $L_{max}$=40, the difference between the lower bound and the upper bound is only 3.25, therefore we use the lower bound of the password space to approximate the actual password space.

## For colored Pass-Go-G

Our approximation method is based on the following observations:

- Based on any password with length of $L$, adding one dot ($8 \times G^2$) or extending the last stroke by one unit in each direction available (the least 3 and the most 8), could derive a new password with length of $L$+1.
- There are $8 \times G^2$ passwords when $L_{max}$=1.

For colored Pass-Go-$G$, the lower bound of the full password space will be

$$\sum_{i=1}^{L_{max}} 8 \times G^2 \times (8 \times G^2 + 3)^{i-1}$$

and the upper bound of the full password space for colored Pass-Go-$G$ will be

$$\sum_{i=1}^{L_{max}} 8 \times G^2 \times (8 \times G^2 + 8)^{i-1}$$

When $L_{max}$=40, the difference between the lower bound and the upper bound is only 0.43 bits, therefore we use the lower bound of the password space to approximate the actual password space.

# Appendix D The content of the FAQ page on the user study website

1. What is Pass-Go?

   Pass-Go is a new graphical password scheme, in which a user draws dots and/or lines on a grid as a way of inputting password, as shown in Figure A.

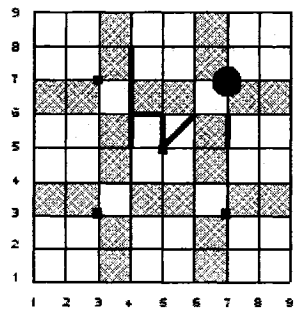


Figure A

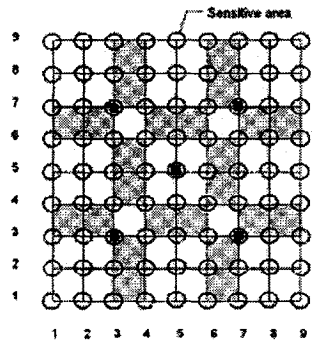



Figure B

2. How can I draw a dot on the grid?

To draw a dot, you can simply CLICK your mouse on an intersection.

3. How can I draw a line on the grid?

To draw a line, you can DRAG your mouse through two or more intersections.

4. Do I need to click or drag through the exact intersection point in order to input my password?

No. Actually there is a sensitive area surrounding each intersection, as shown in Figure B. Touching the sensitive area has the same effect as touching the exact intersection point.

5. Do the sequence and direction I draw matter?

Yes. Pass-Go is a precise recall based scheme, so you have to draw in the exactly same order, direction and color every time.

6. Does a dot count as a separate stroke?

Yes.

7. What is the password length in Pass-Go?

In Pass-Go, the length of a password is the total number of times your password passes through intersection.

8. How can I draw in colors other than black?

In Pass-Go, you can draw in eight different colors. The default is black. Clicking on the color button will change the color of your pen. When you click on a button, it will become "disable" right away, but clicking on another button will enable it again.

9. Do I have to choose colors?

No. Choosing colors is an optional feature, but not required.

10. What is the system requirement?

This web page contains a Java Applet, which requires Java enabled and jre (java running environment) installed on your computer. If you can not see a grid on your screen, you can download this free software by clicking here.

11. Why is this scheme called Pass-Go?

Pass-Go was inspired by an old Chinese game, Go Game, and that is where the name came from.