Pass-Go: A Proposal to Improve the Usability of Graphical Passwords

Hai Tao and Carlisle Adams
(Corresponding author: Carlisle Adams)

School of Information Technology and Engineering, University of Ottawa
800 King Edward Avenue, P. O. Box 450, Station A, Ottawa, Ontario, Canada K1N 6N5
(Email: htao@wlu.ca, cadams@site.uottawa.ca)
(Received Nov. 28, 2006; revised and accepted Apr. 30, 2007)

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 full 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.

Keywords: Dictionary attack, graphical password, Pass-Go, memorable, password space

1 Introduction

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 [8, 15, 32, 37], 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 [17, 18]. This "small dictionary" attack is so successful that in Klein's case study [13], 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). 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 [4, 19, 25]. 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. [12] 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}[1] =12) is 2.3×10^{17}, larger than that of textual passwords composed of 8 printable ASCII characters ($958 = 6.6 \times 10^{15}$).

Thorpe and Van Oorschot [30, 31, 35] 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

[1] L_{max}, Maximum password length (i.e., passwords longer than Lmax are considered to have a probability of zero to be chosen)

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?

In this paper we propose a new grid-based graphical password scheme, Pass-Go, in which users select intersections on a grid to authenticate to a system. Our scheme was inspired by an old Chinese game, Go, which is famous with its inexhaustible variety and simple rules, and played by more than 100 million people around world [34] (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 Subsection 3.2.

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. The size of the graphical dictionary can be reduced to a minimum of 43 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.

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

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.

- **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.

  Blonder [3] 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") 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. Similar schemes were suggested by [21, 23, 24].

  Birget et al. [2] released the restriction 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 the 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) [36]. Their result shows that it is easier to create Passpoints passwords than alphanumeric passwords.

The common problems of single-image schemes include:

1) The security has not been proven by quantifying the memorable password space;
2) 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;
3) 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;
4) Looking for small spots in a rich picture might be tiresome and unpleasant for users with weak vision.

- **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 [10]. This class of passwords was shown to be remembered by users for a long period after short perception [7].

  Passfaces^{TM}, a commercial product, requires a user to select previously seen human face pictures as a password [20]. Brostoff and Sasse [5] conducted a user study (34 subjects involved) and their result suggests that Passfaces^{TM} is easier to remember than textual passwords. Davis et al. [6] 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. 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^{TM} the user's choice is highly affected by race, the gender of the user, and the attractiveness of the faces. For example, 10% of the passwords of males could be cracked by merely two guesses. Compared to that, in the story scheme, the worst 10% of passwords for the most predictable population still required twenty guesses to break.

  By exploiting hash visualization techniques [22], another scheme called Déjà Vu [7] was designed with non-describable abstract images, 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.

  Other similar schemes and discussion can be found in [11, 29]. In general, multiple-image schemes all suffer from following common shortcomings:

  1) Considerably large display space is needed to hold multiple images;
  2) 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;
  3) Passwords have to be stored in the clear, therefore the authentication server is required to be strongly protected;
  4) 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. [12], DAS (Draw-A-Secret) 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 Subsection 3.1.

A common problem for graphical passwords is shoulder surfing: 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.

Stubblefield and Simon [26] 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. However, we don't categorize it into the scope of graphical passwords, but rather as a mnemonic strategy for textual passwords, such as Passphrases [38].

One might notice that our categorization of graphical passwords differs from most literature [7, 28, 35, 36], where a boundary line was drawn between recognition and recall. Monrose and Reiter [14] 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.

3 Design and Analysis of a New Grid-based Scheme

In this section, 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

In DAS, a user draws lines on a grid and the display shows the actual trace, as shown in Figure 1. 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 1 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;

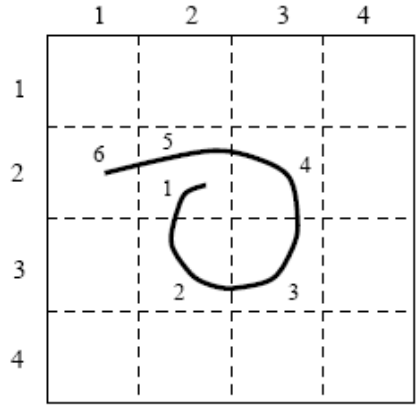


Figure 1: DAS password

- 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);
- L_{max}: passwords longer than L_{max} are considered to have a probability of zero to be chosen.

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 $95^8 \approx 25^3$).

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 2.56×10^6 passwords, which is approximately the size of a textual dictionary [13]. 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. [9] conducted a small scale user study on a similar scheme (Passdoodle) and 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 [30] studied the memorable password space of DAS, and introduced the concept of a symmetric graphical dictionary, based on psychological theories [1] 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".

S_{1b} is a subset of S1a. 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^{TM}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 S1a or S1b respectively.

Nali and Thorpe [16] conducted a small scale user study, in which 16 users were instructed to draw DAS passwords on paper, rather than in a real system. Their result shows that approximately 45% of users chose symmetric passwords, 2/3 of which were mirror symmetric. 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.

Thorpe and Van Oorschot [31] 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. 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).

Van Oorschot and Thorpe [35] suggested that an attacker might prioritize his 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 =12 on a 55 grid, which could be effectively exhausted by one 3.2GHz Pentium^{TM}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

3.2.1 Select Intersections Instead of Cells

As the name implies, 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 2. The shape and size of the sensitive area can be predefined. In our implementation, sensitive areas are round circles with a radius of$\frac{1}{4} \times d$(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 2. 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 [12] (i.e., a grid comprised of $G+1$ horizontal and vertical lines).

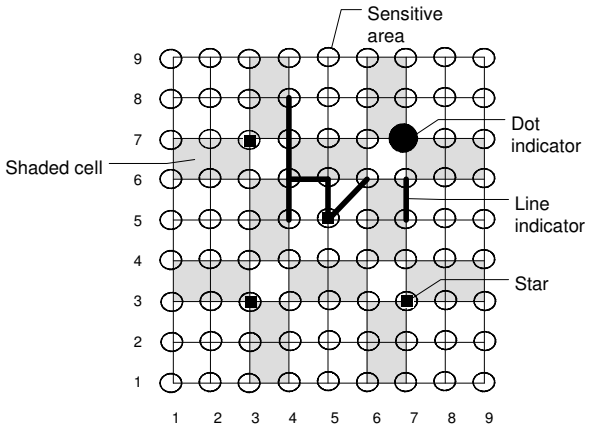


Figure 2: Pass-Go design

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 2. 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 will repetitively impact a user's brain and thus accelerate the process of memorization.

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 2 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;
- 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;
- L_{max}: passwords longer than L_{max} are considered to have a probability of zero to be chosen.

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 2, 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 Section 4). 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).

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 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. [12], 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 $n(x, y, l, G)$[2] 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

$$\begin{aligned} n(x, y, l, G) = {} & n(x-1, y-1, l-1, G) + n(x-1, y, l-1, G) \\ & + n(x-1, y+1, l-1, G) + n(x, y-1, l-1, G) \\ & + n(x, y+1, l-1, G) + n(x+1, y-1, l-1, G) \\ & + n(x+1, y, l-1, G) + n(x+1, y+1, l-1, G). \end{aligned}$$

Let us define $NumberOfColors$ as the number of colors available in a Pass-Go implementation (throughout this paper, we assume $NumberOfColors = 8$). To compute the full password space for colored Pass-Go, we further modify the function $N(l, G)$[3] 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 Pass-Go, which is a subset of that of colored Pass-Go[3]. The bit-sizes of full password spaces for Pass-Go-9 and colored Pass-Go-9 are given in Table 1, and the comparison with that of Pass-Go-5 and DAS-5 is shown in Figure 3. Values given are log_2 (number of passwords).

In Table 1, we consider the maximum value of L_{max} as 40, as justified by our user study (see details in Subsection 4.4.1), 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 3 shows 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 Subsection 4.4.5.

[2] $n(x, y, l, G)$ 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.

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

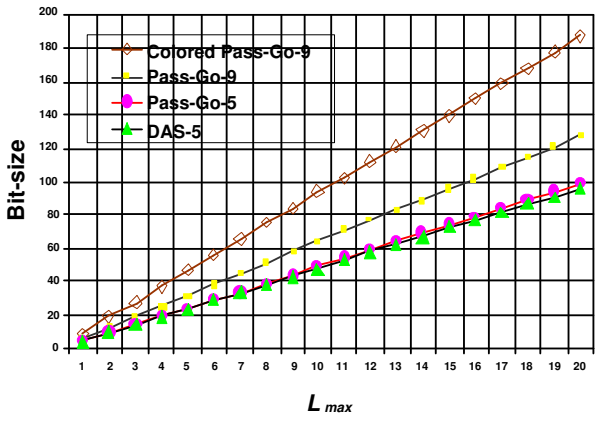


Figure 3: Comparison of full password spaces between colored Pass-Go-9, Pass-Go-9, Pass-Go-5, and DAS-5

Table 2: Direction codes

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

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.

Color codes (if applicable) can be encoded by 01-08. The password in Figure 2, 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.

An alternative for movement encoding is to repeat the direction codes multiple times based on the number of in-

Table 1: Full password space in bit-size for Pass-Go-9 and colored Pass-Go-9

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	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.

tersections subsequently passed. 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 "$\rightarrow\nearrow\uparrow\nwarrow\leftarrow\swarrow\downarrow\searrow$", and penup with ",", to improve the readability. Then the previous password can be encoded as:

$$Black, (4,8)\downarrow_3, red, (4,6)\rightarrow_1\downarrow_1\nearrow_1, (7,7), blue, (7,6)\downarrow_1 .$$

3.5 Keyboard Input and Textual Display Support

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. $\downarrow$) 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.

3.6 Solutions to Shoulder Surfing

In this section, we suggest several 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 Subsection 4.3.3. The other is to use disguising indicators (see Figure 4). 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. Users can also trigger disguising indicators by using the right button of a mouse. In addition, textual passwords can be combined with graphical passwords. For example, a user can press one or more keys on the keyboard (between strokes, or even during a stroke), to add extra encodings to the password, thus increase the difficulty for onlookers to learn the password (and the security of the password).

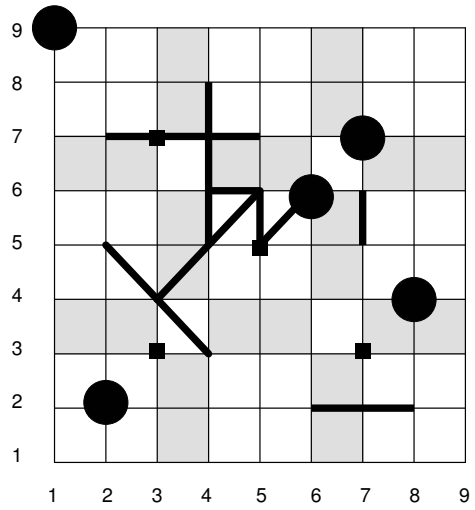


Figure 4: Disguising indicators

3.7 Dynamic Password Policy

Conventional password policies make rules to disallow certain kinds of weak passwords [27, 39]. 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

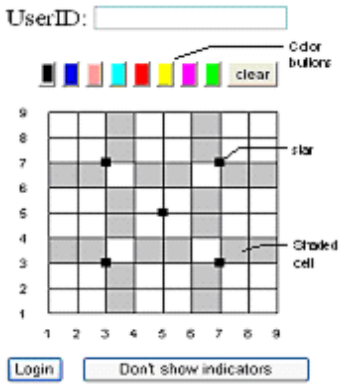


Figure 5: Login interface

be excluded from the attacker's dictionary, whose size is thus reduced.

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 Subsection 4.4.7.

4 User Study

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 Subsection 3.7 work in the way we expect?

4.2 Outline of User Study

Colored Pass-Go-9 along with a simple teaching management system was implemented on a website, which is accessible through the Internet by Java enabled browsers. The login interface is shown in Figure 5. 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 Subsection 3.6 to alleviate the "shoulder surfing" problem. Passwords were stored in the clear in the server to facilitate our access and 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 first author of this paper. 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. 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. They were informed that a FAQ page 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.

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. 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

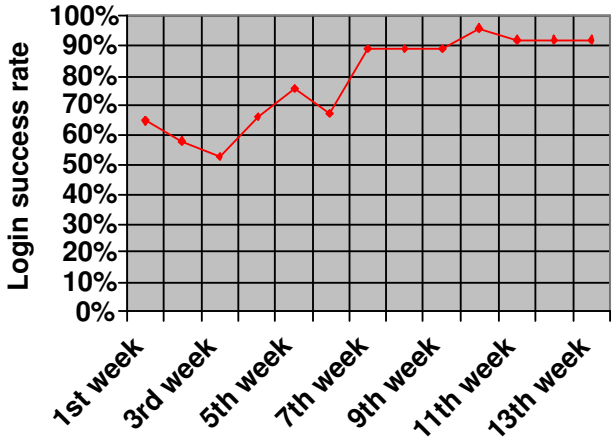


Figure 6: Weekly login success rate

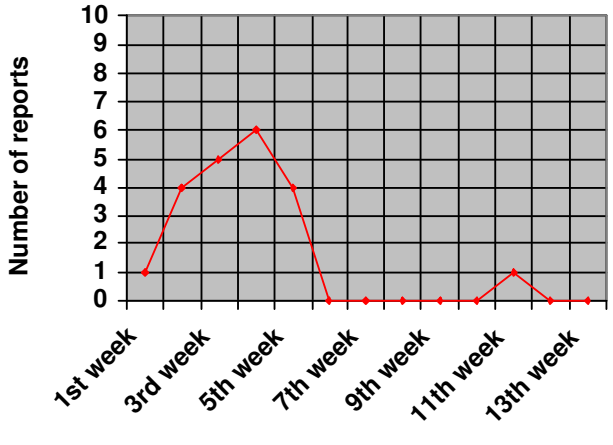


Figure 7: Password-forgotten reports

under different password policies (see detail in Subsection 4.3.2). Because the stroke-count was shown to have the largest impact on password space [35], 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 Subsection 3.7, with a probability of 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 first author of the paper, who then reset the password after authenticating the user's identity.

While we did suggest that users change their passwords 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 second 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.

4.3 Analysis on Usability

4.3.1 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 6.

Figure 6 shows that the login success rate was low in the first three weeks, but kept going up until the 7th week, when it became stable at around 90%.The opposite trend was observed for the password-forgotten report rate, as shown in Figure 7. There were in total 21 password-forgotten reports, 95% of these were reported in the first

Table 3: The success rate for creating 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

5 weeks, with only 1 report in the subsequent 2 months. 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.

4.3.2 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.

4.3.3 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.

4.4 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

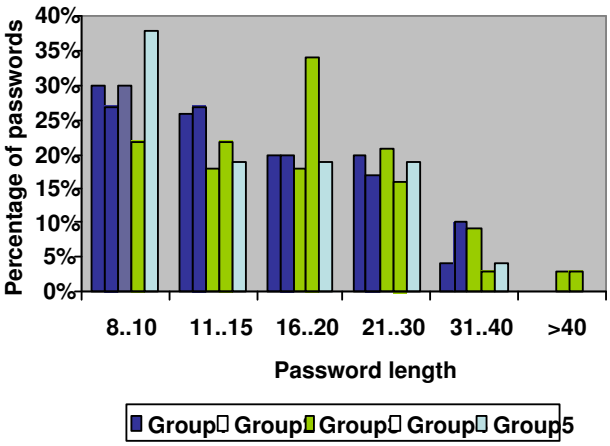


Figure 8: Password length distribution

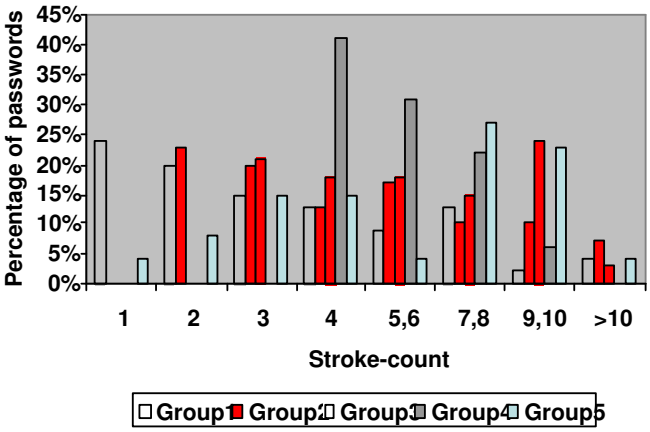


Figure 9: Password stroke-count distribution

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, and more secure passwords (longer, more strokes, more dots, more colors, or more abstract drawings) can be expected 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.

4.4.1 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 Subsection 3.3, where we set $L_{max} = 40$ and obtained an extremely large full password space.

From Table 4 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 Subsection 4.3.3).

Figure 8 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.

4.4.2 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,

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%					

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 [31] 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, the attacker has to restrict stroke-count to 6. Figure 9 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.

4.4.3 Dot

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 dictionary, as suggested by [31] 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 [33]. Go players might choose a drawing solely composed of dots, to simulate a possible situation in a game. 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.

4.4.4 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 $8 \times (3 \times 7) = 168$ (considering that a 2-color-password must start with a color code, the second color code occurs before one of the subsequent strokes, and the average stroke-count in group 1 is $3.80 \approx 4$).This means that even without a color policy, the security of 15% passwords can be increased by about 7.4 bits ($log_2 168$).

However, the color distribution was quiet uneven, as shown in Figure 10. We see that red and blue are the most often used, 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.

4.4.5 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

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%		
	37%		12%		29%		19%	3%

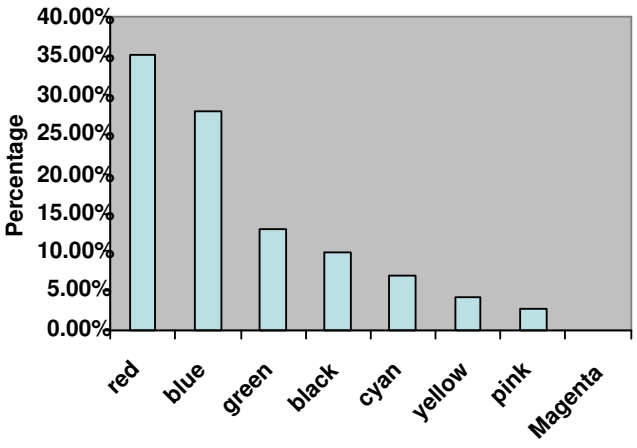


Figure 10: Color distribution for colored passwords

they were drawn also reflect various individual habits and interests. Twelve percent of the passwords can be categorized into well-known symbols, such as $\equiv, \neq, \div, \$$ 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. 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 be classified or associated with concrete meaning by us.

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 [38]. The reason is simple: a mnemonic strategy is also a perfect guide for an attacker to build a corresponding dictionary.

4.4.6 Symmetric

By ignoring color, out of 167 passwords, 67 (40%) fall into S_{1b}(recall Subsection 3.1). 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 [13] (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 [30] that the security of DAS had been overestimated originally.

It is surprising to see that 68 passwords (41%) fall into S_{1a} (recall Subsection 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 [30]), 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

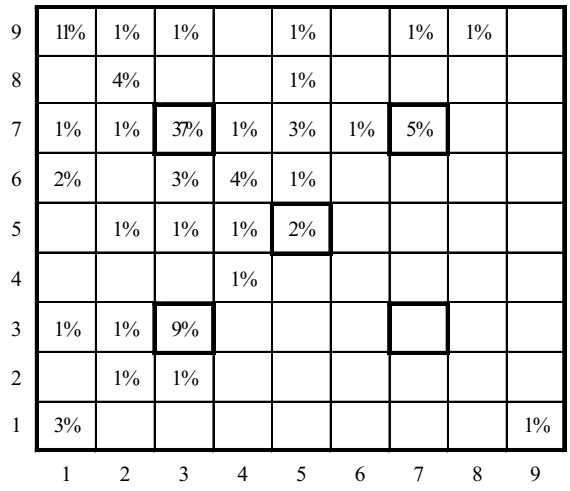


Figure 11: Distribution of starting points in groups 1-4

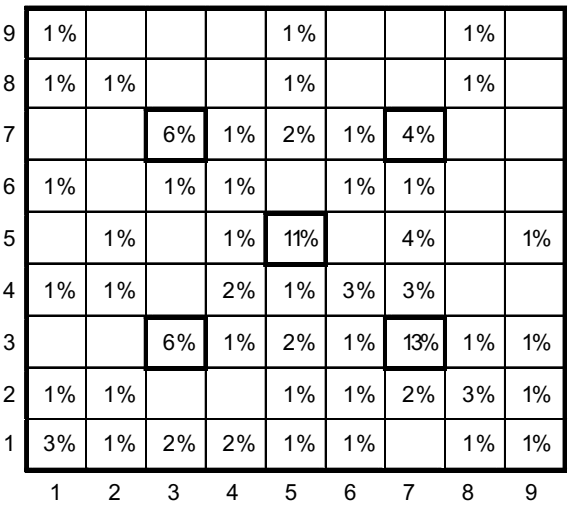


Figure 12: Distribution of ending points in groups 1-4

average length in our user study) to capture a fraction of symmetric passwords. By excluding passwords longer than 16, there are 38 (23%) passwords left in S_{1a} and 37 (22%) left in S_{1b}.

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 Subsection 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^{TM}4 machine (following the same method used in [30]) 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 (1.2×1013), which however is still 3.3 times the password space of 7 alphanumeric character passwords ($627 = 3.5 \times 1012$). One 3.2GHz Pentium^{TM}4 machine will take 9.7 days to exhaust such a dictionary; however, this dictionary will 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.

4.4.7 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 6 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 11 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 12), and cover most parts of the grid, 48% of the passwords still end at stars or corners in groups 1-4.

This uneven distribution of starting and ending points may be taken advantage of by an attacker to prioritize his 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 dictionary by restricting the starting point to (3,7), the most selected starting point, the size of his dictionary can be reduced by 6.3 bits ($log_2 81$), but with the cost of giving up 63% of the passwords.

Figures 13 and 14 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 Subsection 4.2. 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

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 (percentage)	
					Group 1	whole
Full Space[4]	57.7	541.8 yrs	102*	1.2×1012 yrs	28(61%)	96(57%)
S_{1a}	48.1	255 days	79.2*	1.6×109 yrs	8(17%)	38(23%)
S_{1b}	42.7	6 days	71.9*	1×107 yrs	8(17%)	37(22%)
S_2	40.2	1.1 days	57*	334 yrs	21(46%)	
$S_{1b} \cap S_2$	30.7	2.1 mins	43.4*	9.7 days	7(15%)	NA**

[4] 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.

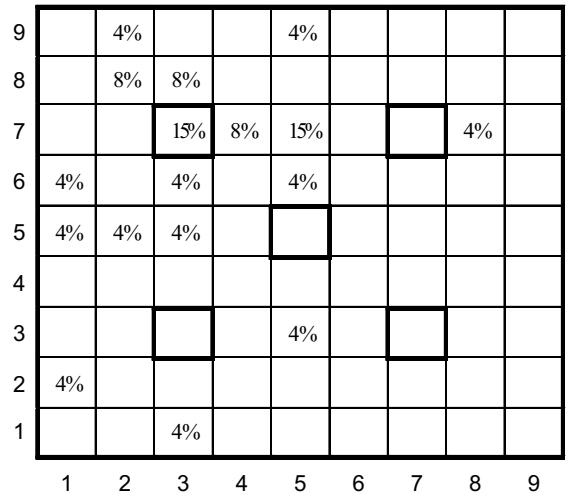


Figure 13: Distribution of starting points in group 5

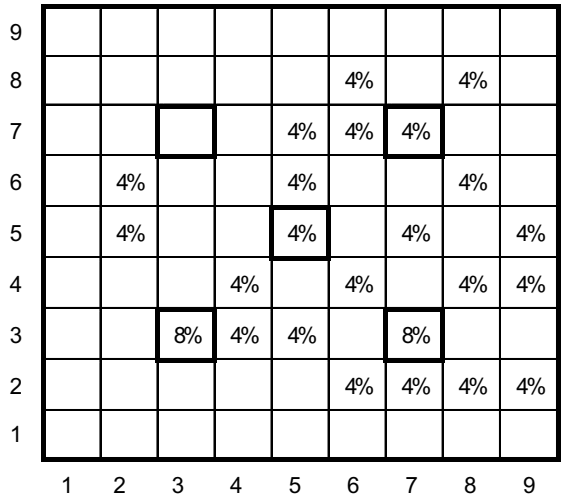


Figure 14: Distribution of ending points in group 5

passwords starting from stars or corners are more likely to end at stars or corners.

5 Variations Based on Pass-Go

In this section, 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.

5.1 PassCells

In PassCells, instead of showing a grid, a matrix of cells is presented on the display, as illustrated in Figure 15. The matrix of cells works in the same way as the sensitive areas in Pass-Go. This change makes the boundary of sensitive areas visible to users, and might be preferred by certain users.

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 small matrix size, such as 5×5 or 7×7.

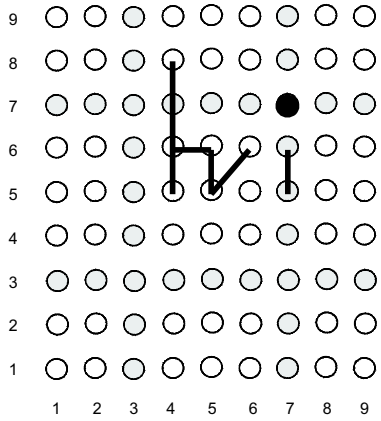


Figure 15: PassCells

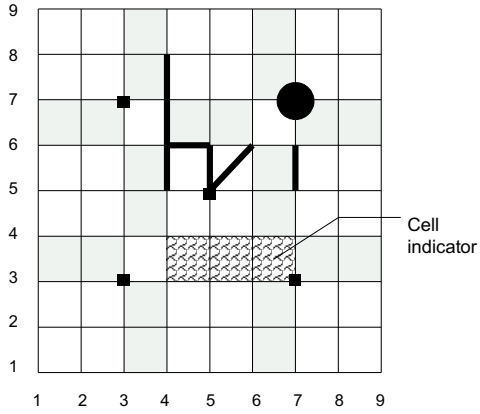


Figure 16: Cell indicators

5.2 Cell Indicator

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 16.

5.3 Curved Line Indicator

We define a cell center as an area surrounding the center of each cell in a grid, as shown in Figure 17. 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, $\frac{1}{4} \times d$ (where d is the side length of a grid cell). Similar to sensitive areas, the cell centers are invisible to users.

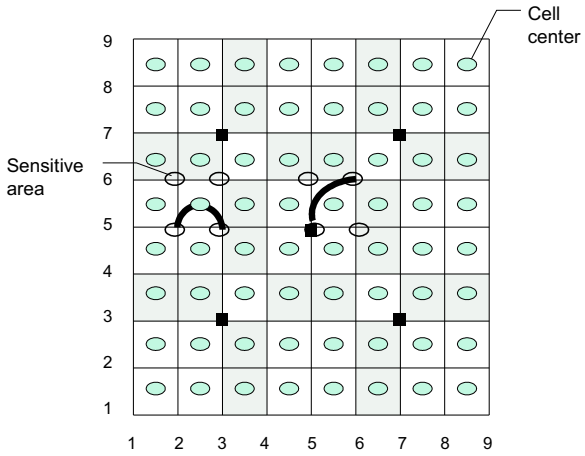


Figure 17: Curved line indicators

We define a cell center as an area surrounding the center of each cell in a grid, as shown in Figure 17. 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, $\frac{1}{4}d$ (where d is the side length of a grid cell). Similar to sensitive areas, the cell centers are invisible to users.

6 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. 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; several solutions for the shoulder surfing problem; a dynamic password checking method; and three variations on the basic scheme.

We conducted an informal user study 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.

Future work should be directed toward optimizing the

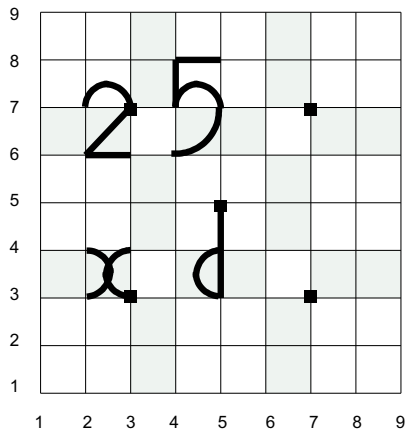


Figure 18: Example password with curved lines

Pass-Go scheme: designing more helpful referencing aids; finding the optimal 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 devices, in a user group representative of the general population, would be helpful to compare the security and usability between them.

Acknowledgments

We would like to thank Julie Thorpe for providing reference data and Paul Van Oorschot for valuable feedback. Many thanks as well to Guy-Vincent Jourdan for his assistance in conducting the user study.

References

[1] F. Attneave, "Symmetry, Information and memory patterns," *American Journal of Psychology*, vol. 68, pp. 209-222, 1955.

[2] J. Birget, D. Hong, and N. Memon, "Robust discretization, with an application to graphical passwords," *Cryptology ePrint Archive*, Report 2003/168, http://eprint.iacr.org/2003/168, Jan. 29, 2006.

[3] G. Blonder, *Graphical Passwords*, United States Patent 5559961, 1996.

[4] G. H. Bower, M. B. Karlin, and A. Dueck, "Comprehension and memory for pictures," *Memory and Cognition*, vol. 3, pp. 216-220, 1975.

[5] S. Brostoff, and M. A. Sasse, 2000, "Are Passfaces^{TM} more usable than passwords? A field trial investigation," *Proceedings of Human Computer Interaction*, pp. 405-424, 2000.

[6] D. Davis, F. Monrose, and M. K. Reiter, "On user choice in graphical password schemes," *Proceedings of the 13th USENIX Security Symposium*, pp. 151-164, 2004

[7] R. Dhamija, and A. Perrig, "D′ej‘a Vu: A User study using images for authentication," *Proceedings of the 9th USENIX Security Symposium*, pp. 45–58 , 2000.

[8] D. Feldmeier, and P. Karn, "UNIX password security-ten years later," *Proceedings of the 19th International Conference on Advances in Cryptology (Crypto '89)*, LNCS435, pp. 44–63, Springer-Verlag, 1989.

[9] J. Goldberg, J. Hagman, and V. Sazawal, "Doodling our way to better authentication," in *Conference on Human Factors and Computing Systems, Poster Session: Student Posters*, pp. 868-869, 2002.

[10] H. Intraub, "Presentation rate and the representation of briefly glimpsed pictures in memory," *Journal of Experimental Psychology: Human Learning and Memory*, vol. 6, no. 1, pp. 1-12, 1980.

[11] W. Jansen, S. Gavrila, V. Korolev, R. Ayers, and R. Swanstrom, *Picture Password: A Visual Login Technique for Mobile Devices*, NIST Report-NISTIR7030, 2003.

[12] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," *Proceedings of the 8th USENIX Security Symposium*, pp. 1–14, 1999.

[13] D. Klein, "Foiling the cracker: A survey of, and improvements to, password security," *Proceedings of the 2nd USENIX Security Workshop*, pp. 5-14, 1990.

[14] F. Monrose, and M. K. Reiter, Graphical passwords, Security and Usability, L. Cranor and S. Garfinkel Edit, O'Reilly, Ch. 9, pp. 147-164, 2005.

[15] R. Morris, and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 11, pp. 594-597, 1979.

[16] D. Nali, and J. Thorpe, *Analyzing User Choice In Graphical Passwords*, Technical Report TR-04-01, Carleton University, Canada, 2004.

[17] *Openwall Project, John The Ripper Password Cracker*, Jan. 29, 2006, http://www.openwall.com/john/.

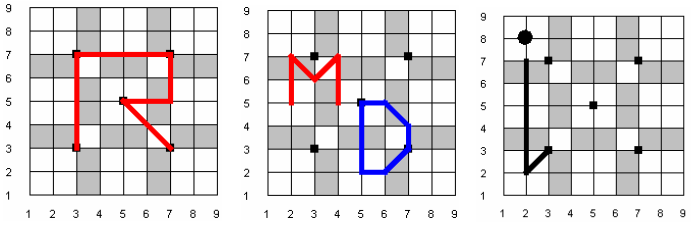
[18] *Openwall Project, Wordlist*, Jan. 29, 2006, http://www.openwall.com/passwords/wordlists.

[19] A. Paivio, T. B. Rogers, and P. C. Smythe, "Why are pictures easier to recall than words?" *Psychonomic Science*, vol. 11, pp. 137-138, 1968.

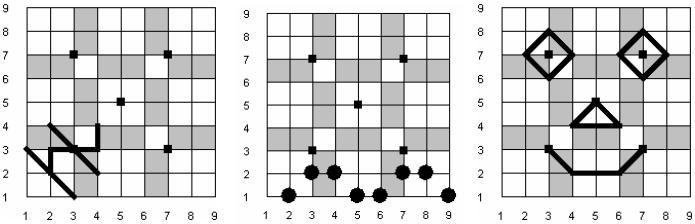
[20] *Passfaces, The science behind passfacesTM for windows*, site accessed on Jan. 29, 2006, http://www.realuser.com/resources/white%20papers.htm.
[21] *Passlogix*, site accessed on Jan. 29, 2006, http://www.passlogix.com.
[22] A. Perrig, and D. Song, "Hash visualization: A new technique to improve real-world security," *International Workshop on Cryptographic Techniques and E-Commerce*, pp. 131-138, 1999.
[23] *SFR*, site accessed on Jan. 29, 2006, http://www.viskey.com/tech.html.
[24] L. Sobrado, and J. Birget, *Graphical Passwords, The Rutgers Scholar*, Rutgers University, Camden New Jersey 081024, 2002.
[25] L. Standing, "Learning 10,000 pictures," *Quarterly Journal of Experimental Psychology*, vol. 25, pp. 207-222, 1973.
[26] A. Stubblefield, and D. R. Simon, *Inkblot Authentication*, Microsoft Technical Report MSR-TR-2004-85, 2004.
[27] W. C. Summers, and E. Bosworth, "Password policy: The good, the bad, and the ugly," *Proceedings of the winter international synposium on Information and communication technologies*, pp. 1–6,2004.
[28] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: A survey," *21st Annual Computer Security Applications Conference (ACSAC)*, pp. 463–472, Dec. 5-9, 2005.
[29] T. Takada, and H. Koike, "Awase-E: Image-based authentication for mobile phones using user's favorite images," *Human-Computer Interaction with Mobile Devices and Services*, vol. 2795, pp. 347-351, Springer-Verlag, GmbH, 2003.
[30] J. Thorpe, and P. C. Van Oorschot, "Graphical dictionaries and the memorable space of graphical passwords," *Proceedings of the 13th USENIX Security Symposium*, pp. 135-150, 2004.
[31] T. J. Thorpe, and P. C. Van Oorschot, "Towards secure design choices for implementing graphical passwords," *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC)*, pp. 50–60, Tucson, USA. Dec. 2004.
[32] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and key issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101-115, Sep. 2006.
[33] USGO, *A Very Brief History of Go*, site accessed on Jan 20, 2006, http://www.usgo.org/resources/gohistory.asp.
[34] USGO, *Top Ten Reasons to Play Go*, site accessed on Jan. 20, 2006, http://www.usgo.org/resources/topten.asp.
[35] P. C. V. Oorschot, and J. Thorpe, *On The Security of Graphical Password Schemes*, Technical Report TR-05-12, Carleton University, Canada, 2005.
[36] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies (Special Issue on HCI Research in Privacy and Security)*, vol. 63, pp. 102-127, 2005.
[37] T. Wu, "A real-world analysis of kerebros password security," *Proceedings of the 1999ISOC Symposium on Network and Distributed System Security*, vol. 8, pp. 723-736, 1990.
[38] J. Yan, A. Blackwell, R. Aanderson, and A.Grant, *The Memorability and Security of Passwords - Some Empirical Results*, Technical Report, no. 500, Computer Laboratory, University of Cambridge, 2000.
[39] J. Yan, "A note on proactive password checking," *ACM New Security Paradigms Workshop*, pp. 127–135, New Mexico, USA, 2001.

Appendix A: Sample user-chosen passwords

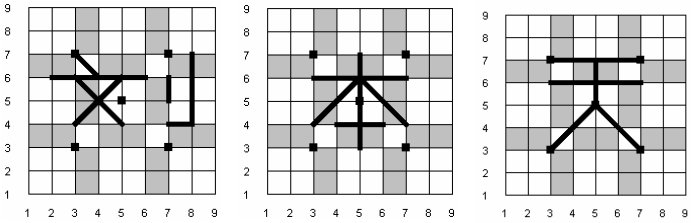
Alphanumeric:



Abstract drawing:



Chinese character:



Appendix B: A method to approximate the full password space

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

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

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

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

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 ($8G^2$) 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

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

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

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

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.

Hai Tao received the M.A.Sc. in Electrical Engineering from University of Ottawa, Canada. He is currently a Systems Analyst at the Wilfrid Laurier University, Waterloo, Canada. His research interests are in the area of cryptography and network security.

Carlisle Adams is a Professor in the School of Information Technology and Engineering (SITE) at the University of Ottawa. Prior to his academic appointment in 2003, he worked for 13 years in industry in the design and standardization of a variety of cryptographic and security technologies for the Internet. His research and technical contributions include the CAST family of symmetric ciphers, protocols for authentication and management in PKI environments, and an architecture and policy language for access control in electronic networks. Dr. Adams is coauthor of Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition (Addison Wesley, 2003) and has published over 70 refereed papers and technical specifications in the areas of cryptography, computer & network security, access control, and privacy.