Graphical Dictionaries and the Memorable Space of Graphical Passwords *

Julie Thorpe, P.C. van Oorschot
School of Computer Science, Carleton University
{jthorpe, paulv}@scs.carleton.ca

Abstract

In commonplace textual password schemes, users choose passwords that are easy to recall. Since memorable passwords typically exhibit patterns, they are exploitable by brute-force password crackers using attack dictionaries. This leads us to ask what classes of *graphical* passwords users find memorable. We postulate one such class supported by a collection of cognitive studies on visual recall, which can be characterized as mirror symmetric (reflective) passwords. We assume that an attacker would put this class in an attack dictionary for graphical passwords and propose how an attacker might order such a dictionary. We extend the existing analysis of graphical passwords by analyzing the size of the mirror symmetric password space relative to the full password space of the graphical password scheme of Jermyn et al. (1999), and show it to be exponentially smaller (assuming appropriate axes of reflection). This reduction in size can be compensated for by longer passwords: the size of the space of mirror symmetric passwords of length about $L+5$ exceeds that of the full password space for corresponding length $L \leq 14$ on a 5×5 grid. This work could be used to help in formulating password rules for graphical password users and in creating proactive graphical password checkers.

1 Introduction

In ubiquitous textual password schemes, users tend to choose passwords that are easy to remember - this often means passwords which have "meaning" to the user. Unfortunately, these (likely chosen) passwords make up only an insignificant part of the full password space. Furthermore, an attacker may build an attack dictionary of "likely passwords" (roughly equated with those easily remembered) from which to draw candidate guesses. In Klein's 1990 case study [13], 25% of 14 000 user passwords were found in a dictionary of only 3×10^6 words. This suggests that a password scheme's security is linked more closely to the size of its *memorable* password space (for a reasonable definition of "memorable"), than that of the full password space (e.g. for 8-character passwords of digits and mixed-case letters, about 2×10^{14}).

Various psychological studies show that people have significantly better recall for concrete words than abstract words [12, 4]. We expect that passwords from the full password space – such as "x*t1K$h9" – which have no meaning whatsoever, are even less likely to be recalled than abstract words; in general we would not expect users to choose such passwords. Given the success of dictionary attacks, it appears that the security of a text-based password scheme is related to the size of its memorable password space, much of which consists of character strings representing, or derived from, concrete words. Passphrases (passwords based on mnemonic phrases) are one credible solution; however, given the success of dictionary attacks, it seems they are seldom used.

Graphical password schemes (e.g. [11, 2, 6]) have been proposed as an alternative to text-based schemes. One motivation for graphical schemes is that humans have a remarkable capability to remember pictures. Psychological studies support that people recall pictures with higher probability than words, including concrete nouns [14]. This motivates password schemes requiring recall of a picture in lieu of a word. If the number of possible pictures is sufficiently large, and the diversity of picture-based passwords can be captured, it seems reasonable to believe the memorable password space of a graphical password scheme will exceed that of text-based schemes – thus presumably offering better resistance to dictionary attacks. What

*This paper appears in the *Proceedings of the 13th USENIX Security Symposium*, San Deigo, USA, August 9-13, 2004.

remains to be shown is what sort of pictures people are likely to *select* as graphical passwords – corresponding to what we call the memorable password space. We begin to explore this issue in the present paper.

We analyze the memorable password space (defined in §3), motivated by the questions: (1) How might an attacker build a *graphical dictionary*? (i.e. an attack dictionary against a graphical password scheme); and (2) How successful would a brute-force attack using such a dictionary be? As mentioned, the high success rate of brute-force dictionary attacks against textual passwords is believed to be strongly related to the recall capabilities of humans and how this affects password selection: meaningful and thus more easily remembered strings are frequently chosen as passwords. We suggest that a clever attacker would narrow down the password space, and prioritize guesses, to pictures that people are likely to choose as passwords, based on the images they are most likely to recall.

To search for techniques that an attacker might use in building a graphical dictionary, we consult psychological studies on visual memory. We review cognitive studies indicating the types of images people are most likely to recall (and presumably choose as passwords). A collection of studies [1, 7] supports the idea that people recall symmetric images better than asymmetric images. A particularly interesting observation is that mirror symmetry carries a special status in human perception [27]. This motivates us to focus on *mirror symmetric* graphical passwords. An attacker exploiting this property of mirror symmetry (most probably about a vertical or horizontal axis – see §3) might build a graphical dictionary of the encoded representations of graphical passwords, such that each entry represents at least one mirror symmetric image. If such a dictionary, containing some fraction of all possible graphical passwords, allows successful attacks, then the security of graphical password schemes may be significantly less than e.g. if all passwords in the entire space were equiprobable.

We define a class of memorable graphical passwords in general, and specifically how this class would map to a graphical password scheme proposed in 1999 by Jermyn et al. called Draw-A-Secret (DAS) [11]. We chose to analyze the memorable password space of DAS to determine whether these passwords constitute a sufficiently large password space for adequate security. For clarity, we will refer to the length of a textual password as the *t-length*, and the length of a DAS graphical password as *length* (see §4.1). We wish to determine a password-length parameter for DAS such that dictionary attacks are more costly than for text-based schemes, given a fixed t-length. This gives the former a chance to be a more secure alternative. We consider the required graphical password length (see §4.1), so that the mirror symmetric graphical password space outsizes the corresponding space of memorable textual passwords.

We define three subsets of our class of memorable passwords (graphical dictionaries) that we believe would form a basic probability-based ordering of a DAS graphical dictionary. In our analysis of the memorable password space, we found that for DAS passwords of length less than or equal to 8 on a 5×5 grid, even our smallest graphical dictionary (§4.4), which is a subset of what we call memorable graphical passwords, is larger in size than the larger textual password dictionaries of 40 million entries [19] (intended for use with password crackers such as John the Ripper [18]). This implies that DAS passwords of length 8 or larger on a 5×5 grid may be less susceptible to dictionary attack than textual passwords.[1]

Under reasonable assumptions and parameter choices, we show the time to exhaustively try all passwords in the full DAS space is approximately 540 years, in comparison to 6 days for one of our proposed symmetric graphical dictionaries. Thus, if as conjectured, a significant fraction of users choose mirror symmetric passwords, the security of the DAS scheme may be substantially lower than originally believed. However, this reduction in size can be compensated for by longer passwords: the size of the space of mirror symmetric passwords of length about $L + 5$ exceeds that of the full password space for corresponding length $L \leq 14$ on a 5×5 grid.

Our contributions include the definition of a class of memorable graphical passwords, the introduction of graphical dictionaries, an analysis of the memorable password space of the DAS scheme of Jermyn et al. [11], and progress towards understanding the subtleties of DAS. Although we focus our analysis on the DAS scheme, our work has general implications for all graphical passwords. This work could be used to help in formulating password rules for graphical password users and in creating proactive graphical password checkers.

The sequel is organized as follows. §2 briefly discusses related work. §3 presents a proposed class of memorable graphical passwords. §4 analyzes this class for the DAS scheme. §5 discusses additional observations and possible extensions to this work, including further concerns about the size of the DAS password space that might be used in practice. Concluding remarks are made in §6.

2 Related Work

There is a fair amount of literature related to the textual password equivalent of this work. Many password cracking dictionaries and tools are available on the Internet such as *Crack* [17] and *John the Ripper* [18]. Understanding these tools and the dictionaries they use is important to perform effective proactive password checking. Yan [28] discusses some popular proactive password checkers such as *cracklib*. Pinkas et al. [23] discuss human-in-the-loop methods to prevent online dictionary attacks; see also Stubblebine et al. [24]. One defense against offline dictionary attacks is to reduce the probability of cracking through enforcing password policies and proactive password checking.

In the open literature to date, there have been surprisingly few graphical password schemes proposed. One using hash visualization [22] was implemented in a program called Déjà Vu [6], based on psychological findings that people *recognize* pictures better than *recalling* them. Generally, in this scheme a user has a portfolio of pictures of cardinality F that they must be able to distinguish within a group of presented pictures of cardinality T.

Birget et al. [2] recently proposed another scheme employing exactly repeatable passwords, which requires a user to click on several points on a background picture.

The DAS scheme ([11]; see §4.1) uses user-defined drawings as graphical passwords. The main difference from graphical pattern recognition is that DAS passwords must be exactly repeatable (as defined within DAS). Exact repetition allows for the password to be stored as the output of a one-way function, or used to generate cryptographic keys. Given reasonable-length passwords in a 5×5 grid, the full password space of DAS was shown to be larger than that of the full textual password space. In our analysis (see §4), we assume DAS as the underlying scheme for encoding graphical passwords, thus we do not consider passwords that are disallowed within DAS.

Regarding memorability issues for graphical passwords, Davis et al. [5] examine user choice in graphical password schemes. Particular to the DAS scheme, Jermyn et al. [11] argue that the DAS scheme has a large memorable password space by modeling user choice. They examine the size of the password space for combinations of one or two rectangles, and show that this is comparable to the size of many textual password dictionaries.[2] A second approach to characterize memorable passwords was based on the existence of a short program to describe the password, under the assumption that all passwords that can be described by a short program are also memorable (rather than on findings from psychology or user studies). A separate user study on memorability performed by Goldberg et al. [8] showed that people are less likely to recall the order in which they drew a DAS password than the resulting image.

Jermyn et al. [11] suggest that the security of graphical password schemes benefit from the current lack of knowledge of their probability distribution; this motivates our present work.

3 Proposed Class of Memorable Graphical Passwords

Since the entries of textual password dictionaries are based on words people recall better, we are lead to examine what types of images people recall better (and thus presumably choose as graphical passwords). In this section, we appeal to psychological studies and discuss the literature leading us to define mirror symmetric graphical passwords as a class of memorable graphical passwords.

Generally, free recall is ordered along the concreteness continuum: concrete words are recalled more easily than abstract words, pictures more easily than concrete words, and objects better than pictures [14]. Various studies support this result (e.g. [12, 4, 15]). Another [3] found that a series of line drawings is poorly remembered if the subject is unable to interpret the drawings in a meaningful way. The more concrete a drawing, the more meaningful it will be to the viewer.

The literature on visual memory often cites better results for human visual recognition than visual recall. However, it has been noted [20] that the methodologies used in studies that test visual recall are flawed in that they depend on people's skill to recreate the image by drawing and/or a well-defined and well-accepted theory of visual similarity for comparison purposes. Additionally, it is worth noting that most visual recall studies allow at most a few seconds for the test subject to view and memorize the image. Given these flaws, one may question the commonly accepted claim that visual recognition is significantly better than visual recall. Even if visual recognition is better than visual recall, visual recall is better than the recall of words. Thus, findings that visual recognition is better than visual recall do not invalidate the likelihood of an increased memorable password space in recall-based schemes over that of recognition-based schemes.

What may invalidate the likelihood of an increased memorable password space in graphical password schemes is if there are patterns in what *types* of images people recall better than others, creating classes of memorable and thus predictable passwords. If such classes are small enough that a brute-force attack is feasible, then the security of graphical password schemes may be no better in practice, or even worse, than that of the standard textual password scheme.

There appears to be little existing research that examines the *types* of pictures people recall better. However, one cognitive study with interesting implications showed experimentally how visual recall progressively changed over time toward a symmetric version of the image [21]. Given a set of asymmetrical, geometric images, when the test subjects were asked to draw the image from recall, all changes made from the originals were in the direction of some balanced or symmetrical pattern. This change was progressive over time toward a symmetric pattern. That people recall images as increasingly symmetric with time suggests that people prefer images that are symmetric. Thus, the direction in our research changed from finding the specific images people are more likely to recall, to finding evidence that people have better recall for patterns and images that are symmetric.

A representative overview of literature for human symmetry perception [26] notes that many objects in our environment are symmetric. Moreover, most living organisms and plants, as well as almost all forms of human construction are mirror symmetric (reflective). There is mirror symmetry in people, animals, leaves, flower petals, automobiles, planes, trains, art, buildings, tools, furniture, and religious symbols. The objects in the average office or home are another example. There is also significant evidence [27] that mirror symmetry has a special status in human perception over other symmetry types such as repetition, translation or rotational symmetry. While symmetry created by other means such as rotation or translation was found to require scrutiny, mirror symmetry is "effortless, rapid, and spontaneous" [26].

The classical studies mentioned earlier found that people have better recall for pictures than words, and better recall for objects than pictures. If people recall objects best, and most objects are mirror symmetric, this suggests that people may recall mirror symmetric patterns best.

That symmetry is recalled best is supported by an observation by Attneave [1] that when subjects were given random patterns and symmetric patterns of dots, the symmetric ones were more accurately reproduced than random patterns with the same number of dots. Attneave theorized that this may indicate that some perceptual mechanism is capable of organizing or encoding the redundant pattern into a simpler, more compact, less redundant form [1]. In a separate study, French [7] observed that dot patterns that were symmetric were more easily remembered. Intuitively, this is no surprise - in the case of mirror symmetry, a subject must only recall half of the image and its reflection axis in order to reconstruct the entire image.

Mirror symmetry has a special meaning to human's visual perception, particularly when the axis is about the vertical and horizontal planes. Mirror symmetry has been found to be more easily perceived as having meaning when it is about the vertical axis, followed by when it is about the horizontal axis [27].

Supported by these collective studies, we propose the following: since people are more likely to recall symmetric images and patterns, and people perceive mirror symmetry as having a special status, a significant subset of users are likely to choose mirror symmetric patterns as their graphical password. We suggest that the mirror symmetric patterns chosen are more likely to be about vertical or horizontal axes, since mirror symmetry about these axes is more easily perceived. For graphical passwords, we thus define *memorable password* to mean a password that exhibits mirror symmetry about a vertical or horizontal axis in its *components* (i.e. those parts of a drawing that are visually distinct), meaning that each component is either mirror symmetric in its own right, or is part of a mirror symmetric pair of components. More formally, these are *Class I memorable passwords*, leaving the door open for future Classes II, III, etc.

We suggest that a clever attacker may specifically try as candidate passwords, in a brute-force attack, all memorable passwords in a graphical password space; and more specifically, those passwords containing all possible symmetric components first with symmetry about all possible vertical axes, followed by those with symmetry about all possible horizontal axes.

4 Analysis of Class I Memorable Password Space

To contribute towards a security evaluation of DAS, we determine the size of the more probable subsets of the DAS *Class I memorable password space* (recall §3), i.e. the number of DAS password encodings (see §4.1) representing at least one memorable password (recall §3). This is based on the reasoning that the number of entries in a "successful" attack dictionary provides a measure of security.

The DAS graphical password scheme relies on a user's ability to recall their DAS password "exactly" (as defined by the resolution of the encoding scheme). What users must recall can be divided into two parts: the temporal order of the strokes used in making the drawing, and the final appearance of the drawing. The latter is what our Class I memorable passwords capture, as it appeals to people's ability to recall images. Assumptions concerning the temporal order (i.e. the order of the input of cells) are made in §4.2 and §4.3 to perform this analysis, leading us to define a set S.

§4.2 discusses our terminology and general approach. §4.3 discusses additional cases. §4.4 discusses variations of the attack dictionary. §4.5 briefly discusses our resulting method to quantify the DAS memorable password space. §4.6 presents some computational results.

4.1 Review of DAS Scheme

The DAS scheme [11, 16] decouples the position of the input from the temporal order, producing a larger password space than textual password schemes with keyboard input (where the order in which characters are typed predetermines their position).

A DAS password is a simple picture drawn on a $G \times G$ grid. Each grid cell is denoted by two-dimensional coordinates $(x, y) \in [1 \ldots G] \times [1 \ldots G]$. A completed drawing is encoded as a sequence of coordinate pairs by listing the cells through which the drawing passes, in the order in which it passes through them. Each time the pen is lifted from the grid surface, this "pen-up" event is represented by the distinguished coordinate pair $(G + 1, G + 1)$. Two drawings having the same encoding (i.e. crossing the same sequence of grid cells with pen-up events in the same places in the sequence) are considered equivalent. Drawings are divided into equivalence classes in this manner.

DAS disallows passwords considered difficult to repeat exactly (e.g. passwords involving pieces lying close to a grid boundary). The definition of "close to a grid boundary" is unclear [11]; we define it as any part of a stroke that is indiscernible as to which cell it lies within, meaning it lies within the *fuzzy boundary* of a grid line. Any stroke is invalid if it starts or ends on a fuzzy boundary, or if it crosses through the fuzzy boundary near the intersection of grid lines. We reuse the following terminology.

- The *neighbours* $N_{(x,y)}$ of cell (x, y) are $(x - 1, y), (x + 1, y), (x, y - 1)$ and $(x, y + 1)$.
- A *stroke* is a sequence of cells $\{c_i\}$, in which $c_i \in N_{c_{i-1}}$ and which is void of a pen-up.
- A *password* is a sequence of strokes separated by pen-ups.
- 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 strokes (excluding pen-ups).

Jermyn et al. [11] recursively compute the (full) password space size, i.e. the number of distinct representations of graphical passwords in the DAS scheme. This gives an upper bound on the memorable password space and thus on the security of the scheme. It is assumed that all passwords of total length greater than some fixed value have probability zero. They compute the full password space size for passwords of total length at most L_{max}. For $L_{max} = 12$ and a 5×5 grid, this is 2^{58}, exceeding the number of textual passwords of 8 characters or less constructed from the printable ASCII codes ($\sum_{i=1}^{8} 95^i < 2^{53}$).

4.2 Basic Terminology and General Approach

To capture visually mirror symmetric DAS passwords, we first consider which reflection axes to use. We assume that the user references the grid lines for the symmetry in the drawing, since if the reflection axis is a point of reference, the password will be easier to repeat exactly. Therefore, the reflection axes considered are those that cut a set of grid cells (Fig. 1a), or are on a grid line (Fig. 1b). This means that any symmetric password drawn such that its axis is off-center within a set of cells is not considered. For example, the password in Fig. 2a is visually symmetric when the grid is not in place, but we do not consider it part of the DAS Class I set of memorable passwords since its reflection axis is not on a grid line or centered in a set of cells as shown in Fig. 2b. We justify this assumption as follows: it is more difficult for a user to draw an exactly repeatable symmetric password without a visible point of reference on the grid for the reflection axis.

We thus define the set of axes within a $W \times H$ grid (width W, height H): $A = A_h \cup A_v$; $A_h = \{1, 1.5, 2, \ldots, (H - 1).5, H\}$; $A_v = \{1, 1.5, 2, \ldots, (W - 1).5, W\}$. Here $i.5$ is the grid line separating rows i and $i + 1$, or columns i and $i + 1$ respectively.

In addition to the visual appearance of a DAS password, the most likely *ways* in which a visually mirror symmetric DAS password can be drawn must be considered in constructing a DAS Class I graphical dictionary. It turns out to be quite tricky to map the idea of a visually mirror symmetric DAS password onto DAS encodings to enumerate, as we describe in a number of cases (below and in §4.3). DAS Class I memorable passwords are only defined in terms of their visual structure. There is a one-to-many relationship between a given Class I memorable password to the number of ways it can be drawn in the DAS scheme (which are then mapped to possibly less unique DAS encodings). We believe there are some more likely *ways* that users will draw mirror symmetric components in their

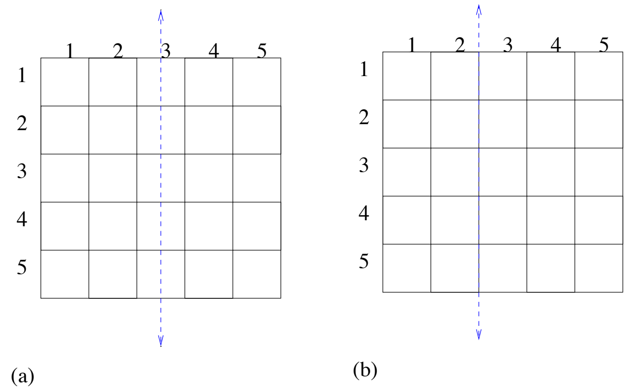


Figure 1: Possible axes can (a) cut a set of cells; or (b) be on a grid line between sets of cells.

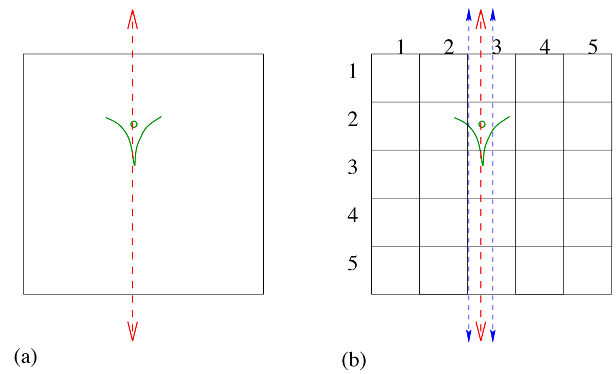


Figure 2: Drawing that is symmetric about a difficult to reference axis. Assuming the v is drawn before the dot, the encoding of (b) is (2,2), (3,2), (3,3), (3,2), pen-up, (3,2), pen-up. If shifted slightly to the right to be symmetric about the vertical axis $x = 3$, it has symmetric encoding (see §4.2): (3,2), (3,3), (3,2), pen-up, (3,2), pen-up.

DAS passwords; we call this "more probable" subset of unique DAS encodings S.

Preliminary user studies have shown that the temporal order has an adverse effect on user's ability to recall a DAS password [8]. If the temporal order is a complicating factor that adds more complexity to what users must recall, it is likely that they will choose DAS passwords with less complexity (e.g. less strokes). We assume the *way* users will draw a DAS Class I password is such that the composite stroke(s) of each mirror symmetric component are drawn in a symmetric manner (as defined in the *disjoint case* as described in this section and the *continuous and enclosed cases* as described in §4.3). We believe the resulting subset S captures the easiest (and thus more likely to be chosen) ways to draw DAS Class I memorable passwords.

We model each symmetric DAS password as a series of strokes (each representing a single component or pair of components) that have *local symmetry* about a set of axes, each such stroke modeled by a virtual start point s and virtual end point e (not necessarily the start and end points of the user-drawn stroke). A stroke has local symmetry if it is symmetric about some axis in a given set of axes. This includes drawings where all or most strokes are symmetric about different axes, which may have no immediately perceivable pattern, as shown in the example password in Fig. 3a. When the strokes are symmetric about axes in the same vicinity, it results in an increasingly symmetric drawing as a whole, which we call *pseudo-symmetry*. An example of a pseudo-symmetric drawing is shown in Fig. 3b. When the strokes are all symmetric about the same axis, it results in a drawing that has *global symmetry* (e.g. the star in Fig. 3c); since all strokes are symmetric about the same axis, the entire drawing is symmetric about the same axis.

We define a *symmetric encoding* to be an encoding that represents an equivalence class of DAS passwords, where at least one password in the equivalence class belongs to S. Using the DAS encoding scheme, a sym-

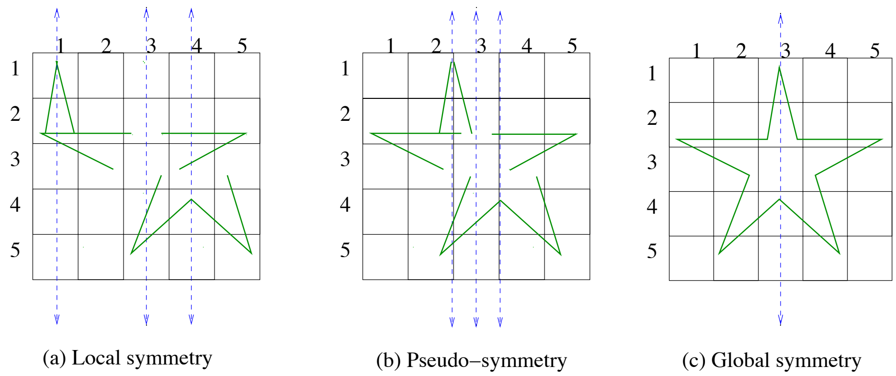


Figure 3: Example Class I memorable DAS passwords (that could be drawn such that they are in S) containing the same components, symmetric about different patterns of axes: (a) 3 different, scattered axes, (b) 3 different, nearby axes, and (c) a single axis.

metric encoding may represent a number of passwords, some of which may not be visually mirror symmetric (e.g. see Fig. 4).

Fig. 4 illustrates different representations of one equivalence class of DAS passwords with the same symmetric encoding. This implies our results count not only mirror symmetric passwords, but also others which are not but belong to an equivalence class in which at least one password is mirror symmetric.

Each stroke within a symmetric encoding is bounded within a *symmetric area*, defined as the area between a given axis and the closest grid boundary parallel to the axis, reflected about the axis (see Fig. 5).

The most obvious way to draw a stroke in a symmetric manner is to draw a stroke within the symmetric area, then draw its reflection about the reflection axis as shown in Fig. 6a. We call the initial stroke from virtual start point s to virtual end point e that the reflection is based upon the *defining stroke*, and the reflection the *reflected stroke*, which can be drawn from s^R (the reflection of s) to e^R (the reflection of e) or vice versa.[3]

Given a defining stroke z, its reflected stroke z^R (relative to an axis a) is said to be an *exact reflection* if z^R is z's mirror image about a and they are separated by a pen-up. Exact reflection is not required to have a stroke that exhibits mirror symmetry (see §4.3). A *symmetric stroke* is the combined result of a defining stroke and a reflected stroke. A *valid point*, relative to an axis a, is any point that is contained within the symmetric area defined by a (see Fig. 5). A *valid defining stroke*, relative to an axis a, is a defining stroke consisting solely of valid points within the symmetric area defined by a. A *valid symmetric stroke* is the composition of a valid defining stroke and its reflected stroke. We define a valid symmetric stroke that holds the property of exact reflection to be the *disjoint case*. As a disjoint case has the property of exact reflection, its length will always be even.

The product of the number of ways to draw a defining stroke and the number of ways to draw its reflected stroke provides the number of ways to draw a symmetric stroke, excluding additional cases (§4.3 discusses the latter, namely the continuous case and the enclosed shape case).

4.3 Continuous and Enclosed Cases

A point in an encoded defining stroke is *potentially continuous* if it lies within a cell that is either cut by the reflection axis a in question, or adjacent to a when a is on a grid line. If a point p is potentially continuous, its reflection p^R is in the same cell as p or in a neighbouring cell, and thus the stroke can be drawn directly from p to p^R without a pen-up. When the start and end points of the defining stroke are potentially continuous, the three most straightforward ways to draw the resulting symmetric stroke are as follows: disjointly (the disjoint case – recall §4.2), as one continuous stroke (the *continuous case*), or as one continuous enclosed stroke (the *enclosed case*).

A symmetric stroke can be drawn as a continuous case when the defining stroke's end point is potentially continuous. We define the continuous case as when the defining stroke continues through a to the reflected stroke, creating a single, *continuous symmetric stroke*. For example, the encoding for Fig. 6b would be: (1,1), (1,2), (1,3), (1,4), (2,4), (3,4), (4,4), (5,4), (5,3), (5,2), (5,1), ending with a pen-up. The stroke could also be drawn in the reverse order. Examples of the same visual representation of a 'U', with one disjoint and the other continuous, are shown in Figures 6a and b.

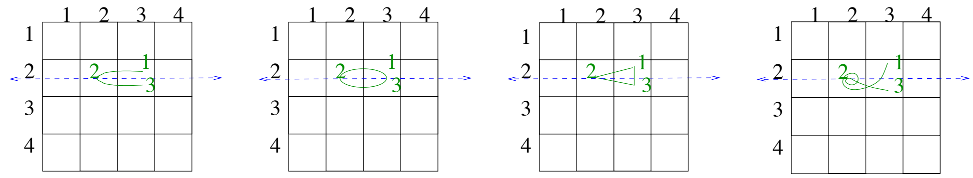


Figure 4: Example DAS passwords in equivalence class with symmetric encoding (3,2), (2,2), (3,2), pen-up.

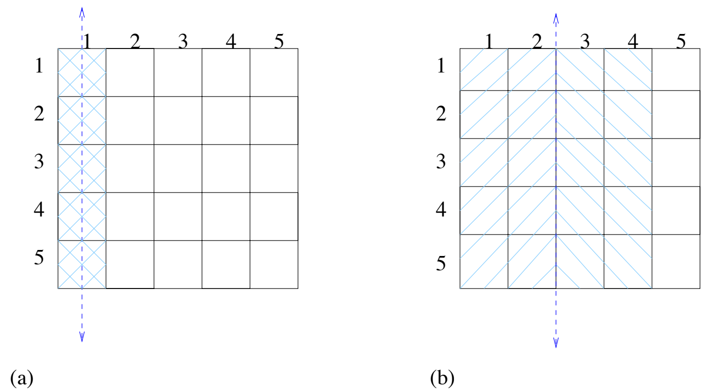


Figure 5: Example symmetric areas for (a) the axis $x = 1$; and (b) $x = 2.5$

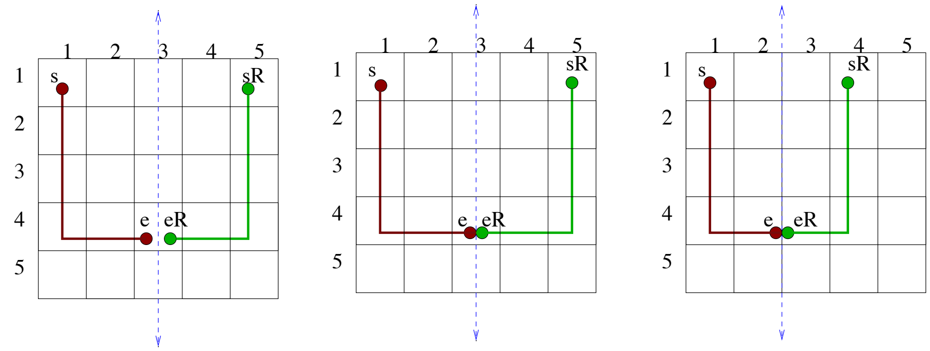
Note that the continuous case's encoding is different, depending on whether a cuts a set of cells or is on a grid line. If a cuts a set of cells as in Fig. 6b, the defining stroke's endpoint e is the same as its reflection e^R. Since there is no pen-up to separate e from e^R, it cannot appear in the encoding twice, thus e^R does not appear in the resulting encoding. If a is on a grid line (Fig. 6c), e and e^R reside in different cells, and e^R does appear in the resulting encoding.

A symmetric stroke can be drawn as an enclosed case when both the defining stroke's start and end points are potentially continuous. We define the enclosed case to be when the defining stroke continues through a to the reflected stroke, and then joins back up with the defining stroke, creating an enclosed shape (e.g. Fig. 7). When a shape is enclosed, the drawing may start and end at any point in the shape and still retain its mirror symmetry. As with the continuous case, the enclosed case's encoding is different, depending on whether a cuts a set of cells or is on a grid line. The continuation of the defining stroke into the reflected stroke will be encoded as in the continuous case; the difference between these two cases is the encoding to join the reflected stroke back into the defining stroke. When a is on a grid line, the start point of the defining stroke is repeated as the last point of the user's stroke (e.g. Fig. 7b). When a cuts a set of cells (e.g. Fig. 7a), it is the same as the continuous case since $s = s^R$, enclosing the shape. Thus, to avoid double-counting, we exclude from the continuous case, the cases where s is potentially continuous.

4.4 Smaller Graphical Dictionaries

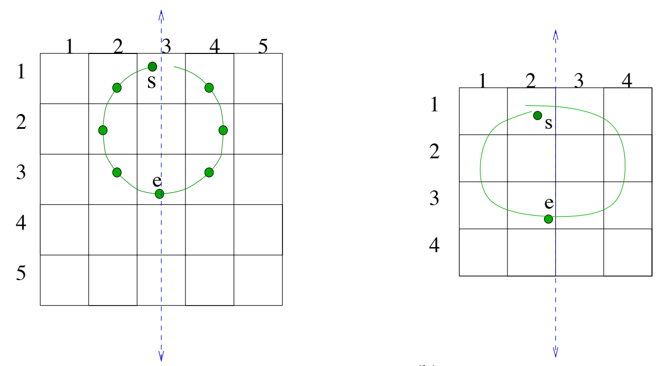
It is in an attacker's best interest to reduce the graphical dictionary size to decrease the attack time and increase probability of success relative to the effort expended. A logical way to attempt to do so is to assume that it is more likely for a user to choose the center-most axes as the reflection axes. We define *Class Ia* as those passwords in Class I whose components (recall §3) are symmetric (in their own right, or pairwise) about the center 3 of each set of axes (i.e. the marked axes in Fig. 8). This produces pseudo-symmetric drawings (recall §4.2, Fig. 3b). This optimization of the graphical dictionary reduces its size as a function of the grid size. We also define *Class Ib* as those in Class I whose components are symmetric about the center vertical and horizontal axes. This produces drawings with global symmetry. The Class Ib dictionary is a subset of the Class Ia dictionary, which is a subset of the Class I dictionary.

If pseudo-symmetry is considered more likely than global symmetry, the attacker may choose to use those passwords that are composed of strokes symmetric about a small set of close axes, such as Class Ia. The



(a) Disjoint case (b) Continuous case (i) (c) Continuous case (ii)

Figure 6: Disjoint and Continuous Cases. Symmetric strokes, consist of a defining stroke (solid line from s to e) and reflected stroke (solid line from s^R to e^R). The last two, visually representing the letter 'U', show continuous cases where: (b) the axis cuts a set of cells; and (c) the axis is on a grid line.



(a) (b)

Figure 7: Different types of the enclosed shape case where the axis in (a) cuts a set of cells and (b) is on a grid line. (a) shows all possible representative start/end points.

size of the dictionary will increase exponentially with each additional axis considered, meaning that the time to exhaust the dictionary is reduced by this method, particularly with higher values of L_{max}. Class Ib captures all passwords that are globally symmetric and centered about the grid (vertically and/or horizontally), plus those that have components symmetric about the center vertical and horizontal axes (e.g. the coffee cup in Fig. 9). If the user subconsciously uses the grid to frame the drawing (i.e. using the grid as part of the drawing's overall symmetry), the resulting drawings would be globally symmetric about either of the center axes.

4.5 Quantifying the Memorable Password Space

Our general approach to quantify $|S|$ (recall §4.2) is to determine how many DAS passwords in S are of length at most a given maximum password length L_{max}. The composite strokes of each password in S have defining strokes that connect a given virtual start and end point in the symmetric area. Counting all passwords of length at most L_{max} and defining passwords in terms of strokes follows Jermyn et al. [11]; however, our method for defining the set of strokes of a given length is entirely different, and only symmetric strokes are included in the set.

The key points of our method of quantifying $|S|$ are discussed in this section (for more details see [25]). Generally, the base formula for defining the set of strokes does the following: for every possible virtual start point $s = (x, y)$, and end point $e = (x, y)$ in a given $W \times H$ grid, we determine the number of ways to draw a symmetric stroke (symmetric about any valid axis in A) of length ℓ based on a defining stroke that joins s to e. The reason for specifying s and e is so we know explicitly whether s and/or e are potentially

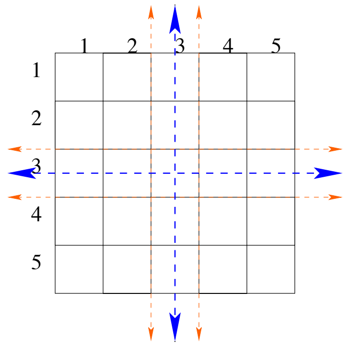


Figure 8: Highest probability reflection axes. The thickest axes are the vertical and horizontal center axes. Adjacent axes are marked in a thinner arrowed line.

continuous (recall §4.3) in order to enumerate the continuous and enclosed cases.

The number of defining strokes from s to e is enumerated by examining the number of permutations of up, down, left, and right movements that join s to e while remaining within the bounds of the symmetric area, for all valid axes in A. The primary considerations in this method are: path *diversions*, and the amount of *room* between the current position and the bounds in every direction within a given symmetric area.

The number of possible diversions for a given s, e, ℓ, and axis $a \in A$ is based on the difference between the desired defining stroke length $\frac{\ell}{2}$ and the minimum length path (stroke with the least number of cells) that joins s to e. The difference between $\frac{\ell}{2}$ and the minimum length path required to join s to e is the number of extra cells that should exist in the stroke from s to e that divert from the minimum length path. In order for the defining stroke with diversions to connect s to e, each diversion must be paired with a cell crossing in the opposite direction to reconnect with the minimum length path. An example of a diversion is provided in Fig. 10.

Room is the number of cell crossings in a given direction that can occur from s, before the defining stroke goes out of the symmetric area bounds in question. If at any point in the defining stroke, the number of left cell crossings exceeds the number of right cell crossings by more than the amount of left room, the defining stroke is invalid. The use of room in other directions is analogously defined. Given a starting point s and the symmetric area, we know the amount of available room in each direction. For example, in Fig. 11, right room = 2, left room = 1, top room = 1, and bottom room = 3.

When ℓ is even, the symmetric strokes enumerated for a given s and e are a combination of the disjoint, continuous, and enclosed cases. When ℓ is odd, the symmetric strokes enumerated for a given s and e are a combination of the continuous and enclosed cases. These sets intersect due to the nature of our counting method. In determining the size of an overall memorable password space, overlaps must be accounted for to avoid double-counting. Fig. 12 gives a representative illustration of how the strokes intersect with one another when ℓ is even (more specifically, $\ell = 12$); when ℓ is odd, the disjoint case is void (recall §4.2).

If a symmetric stroke z has a symmetric defining stroke z^D, and a symmetric reflected stroke z^R, z is the same as two independent symmetric strokes, which can be independently included in S (i.e. they are either continuous symmetric strokes or enclosed symmetric strokes). Thus, we must ensure that all disjoint case symmetric strokes that have symmetric defining strokes are subtracted from the count.

Some enclosed shapes will be double-counted by this method since an enclosed stroke may be symmetric about both a horizontal axis $a \in A_h$ and a vertical axis $a \in A_v$ (e.g. Fig. 13). The double counting is due to counting all possible start/end points of an enclosed shape case. The enclosed shapes that are symmetric about an $a \in A_h$ and an $a \in A_v$ can be identified as those whose defining strokes are symmetric. We identify and subtract those defining strokes that are symmetric continuous cases (including those whose start point is potentially continuous). This involves determining the candidate axis a_c and all candidate midpoints m that will produce a continuous symmetric stroke from s to e. These defining strokes must be identified only once; we identify them when $a \in A_h$. In Fig. 13, the symmetric defining stroke (about the horizontal axis) is indicated as the circular dashed line.

We are aware of other smaller cases of overlap that we have experimentally determined as insignificant to the overall results. One such case is when the reflected stroke is identical regardless of whether it is drawn from s^R to e^R or from e^R to s^R, but is not an enclosed case (e.g. lines that repeat over each other more than once). Additionally, there is a smaller set of defining strokes that result in the same symmetric stroke when reflected about one horizontal and another vertical axis, which occurs when the second half of the defining stroke is a 180 degree rotation of the first half of the stroke. There may be other small cases of overlap that we are presently unaware of, but we believe that the set we used will account for any overlap of significant impact on the overall result.

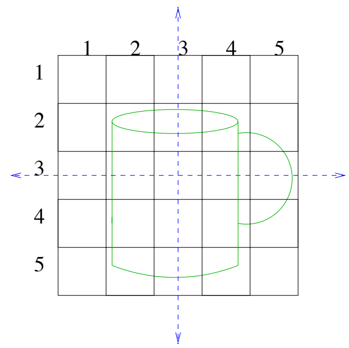


Figure 9: Example Class Ib drawing. One component (the handle) is symmetric about the center horizontal axis, and another (the cup), is symmetric about the center vertical axis.

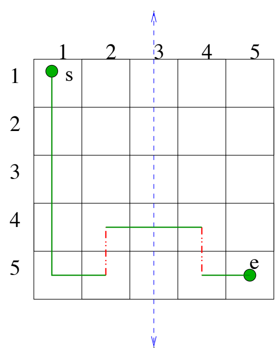


Figure 10: Defining stroke with a diversion (diversion is marked in a dashed line).

4.6 Approximate Size of Class I Memorable Password Space

Table 1 gives sample results computed using the method outlined in §4.5 (for details including equations, see [25]) for S (recall §4.2) and the intersection of S with Class Ia and Class Ib memorable passwords (recall §4.4), respectively denoted S_{Ia} and S_{Ib}. Values given are log_2(number of passwords). S_{Ia} and S_{Ib} both show an exponential reduction from the full DAS space: S_{Ib} grows at an exponential rate of approximately 3.6 bits per unit increase in password length and S_{Ia} grows at a corresponding rate of approximately 4.0, whereas the full DAS space and S grow at a corresponding rate of approximately 4.8. The size of the full DAS password space was double-checked using a variation of our method, and essentially agrees with the results given in [11].

Each of the three subclasses of Class I memorable passwords presented in Table 1 allow perceptually quite distinct classes of drawings (recall Fig. 3 and §4.4). We found the size of S to be surprisingly close to that of the full DAS space; however, upon reflection this is sensible, as the only requirement for a stroke to be symmetric is that it is locally symmetric about any axis in A (e.g. Fig. 3a), which includes the combinatorially large set of all permutations of dots and lines of length two.

The smaller the set of axes used, the smaller the graphical dictionary becomes. It is a reasonable strategy for an attacker to narrow down the graphical dictionary to a small number of axes, or at least prioritize a search such that globally symmetric passwords (e.g. Fig. 3c) are considered first. When a single axis (or two) are considered at a time to produce globally symmetric passwords, each result will never be larger than that for the two center axes, as the latter maximizes the symmetric area in which the passwords can reside. Thus, the maximum dictionary size of such a variation would be at most a small constant factor, proportional to the number of axes considered, of that using only the center axes. We believe that the set of globally symmetric passwords best captures the symmetry discussed in §3, and our intuition suggests that Class Ib (e.g. Fig. 3c) is more likely than Class Ia (e.g. Fig. 3b), which is more likely than Class I (e.g. Fig. 3a).

To provide context for the practical implications of our results, we discuss how long it might take to perform a dictionary attack against the DAS scheme using each of the above graphical dictionaries. The exact method used to perform a dictionary attack depends on the authentication method used by the system. We assume that authentication is performed by hashing the entered password using the MD5 hash algorithm, then comparing the hashed password to the password file entry for the user.[4] In this case, a dictionary attack requires comparing the hashed value of each candidate password to the hashed value of the target password, hoping for a match. Here the attack time is at least the time to hash each candidate password. Thus, we

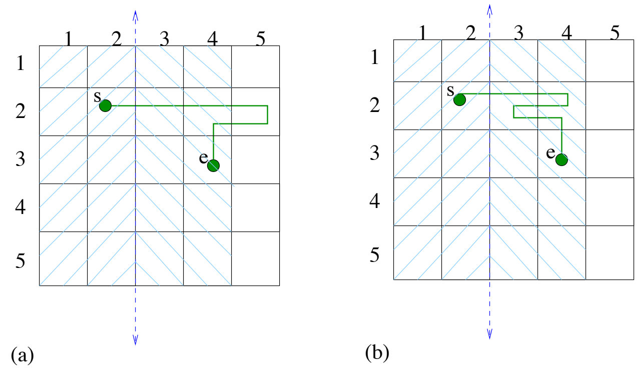


Figure 11: (a) Defining stroke that goes outside of the symmetric area when right room = 2, number of right cell crossings = 3. (b) Defining stroke with the same amount of right room and cell crossings that remains within the symmetric area.

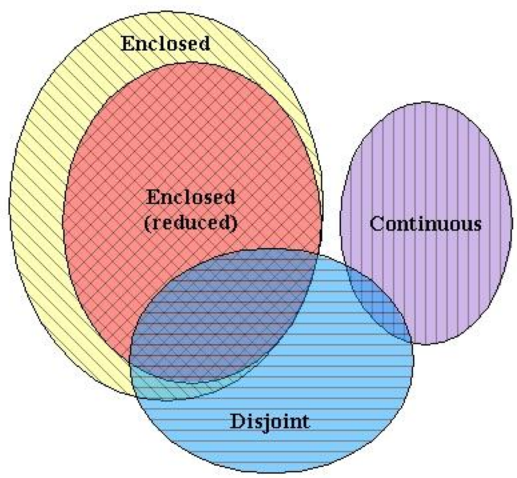


Figure 12: Relationship between different cases of symmetric strokes of length 12 on a 5×5 grid. Enclosed (reduced) refers to the enclosed case, after removing double-counting. (Note: a stroke of length 12 implies a password of length at least 12.)

tabulate the time required to hash all passwords in each password set for comparison.

We calculate two sets of times: one where we assume the attacker has one *Pentium 4* 3.2GHz machine, and another where we assume the attacker has one thousand such machines, with which linear speed-up is achieved. It is reasonable to consider that a determined attacker could exploit one thousand, or even one hundred thousand machines using a worm, to distribute the password-cracking load. Using an MD5 performance result of 3.66 cycles/byte for a *Pentium 3* 800MHz machine [10] (scaled to 3.2GHz), and a 512 bit block size, approximately 1.37×10^7 hashes can be performed per second per machine. Given the assumed resources, the estimated time to generate the password hashes is given in Table 2.

The times provided in Table 2 highlight the implications of the graphical dictionary size. Assuming that we want an attacker to require an average of 10 years to exhaust these dictionaries with 1000 computers at 3.2GHz, the dictionary size must be approximately 2^{63}. Referring to Table 1, our Class Ib dictionary (global symmetry) is above this size when $L_{max} = 18$. This implies that for this level of security (and a 5×5 grid), DAS users should choose passwords of length at least 18.

Note that an attacker may achieve success substantially faster than the times given in Table 2 if dictionary entries are ordered according to their probability of occurring. For example, if the entire Class I password dictionary was used, it would be reasonable to order it such that all those that also fall into Class

L_{max}	1	2	3	4	5	6	7	8	9	10
Full DAS space	4.7	9.5	14.3	19.2	24.0	28.8	33.6	38.4	43.2	48.1
(i) S	4.7	9.5	14.3	19.1	23.9	28.7	33.6	38.4	43.2	48.0
(ii) S_{Ia}	3.3	7.7	11.6	15.7	19.8	23.8	27.9	31.9	36.0	40.0
(iii) S_{Ib}	3.3	6.9	10.5	14.1	17.7	21.2	24.8	28.4	32.0	35.6
L_{max}	11	12	13	14	15	16	17	18	19	20
Full DAS space	52.9	57.7	62.5	67.3	72.2	77.0	81.8	86.6	91.4	96.2
(i) S	52.8	57.6	62.4	67.2	72.0	76.8	81.7	86.5	91.3	96.1
(ii) S_{Ia}	44.1	48.1	52.1	56.2	60.2	64.3	68.3	72.4	76.4	80.4
(iii) S_{Ib}	39.1	42.7	46.3	49.9	53.4	57.0	60.6	64.2	67.8	71.4

Table 1: Bit-size of graphical password space, for total length at most L_{max} on a 5×5 grid.

Dictionary ($L_{max} = 12$)	Time to exhaust (1 machine)	Time to exhaust (1000 machines)
Full DAS Space	541.8 years	197.8 days
S	505.6 years	184.5 days
S_{Ia}	255 days	6.1 hours
S_{Ib}	6 days	8.7 minutes

Table 2: Time to exhaust various dictionaries (3.2GHz machines, 5×5 grid).

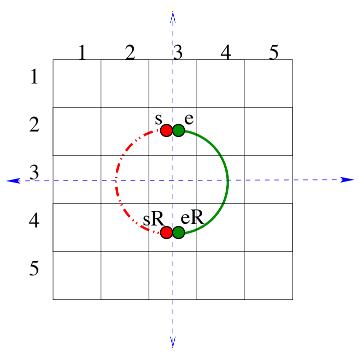


Figure 13: An enclosed shape symmetric about a horizontal and vertical axis; it would be double-counted by our approach, without explicit subtraction.

Ib are first, followed by those remaining that fall into Class Ia, etc. Note that if the target passwords are not in any of the above dictionaries, the attack will fail.

Some of the larger textual password dictionaries contain approximately 4×10^7 entries [19]. Our smallest graphical dictionary exceeds this number of entries for $L_{max} \geq 8$. This implies that even if users choose the globally mirror symmetric passwords we have defined, provided the password length is at least 8, the DAS scheme may still offer greater security than textual passwords against dictionary attacks.

5 Additional Observations and Future Work

One may question the likelihood of users choosing symmetric graphical passwords, based solely on cognitive studies on visual recall. It is interesting to note that out of the 8 example passwords in the original DAS paper [11], 5 fall under our definition of globally symmetric and 7 fall under our definition of locally symmetric. We believe it is difficult to conjure many visually pleasing patterns that do not exhibit symmetry.

The graphical dictionaries discussed earlier do not include repetition symmetry when the components are asymmetric (e.g. Fig. 15) or rotational symmetry. These two forms of symmetry could be classified as Class II and III memorable passwords. These symmetries were not addressed in this analysis as cognitive studies report that they do not hold the same special status as mirror (reflective) symmetry in human perception. It is unknown whether people are as likely to recall repetitive or rotational symmetry more or less efficiently as mirror symmetry. It would be interesting to explore the effect of adding these two forms of symmetry on our graphical dictionaries.

Another interesting direction would be to determine the effect on a dictionary of limiting the number of strokes in DAS passwords to e.g. 3 or 4. One psychological study [7] has shown that people optimally recall

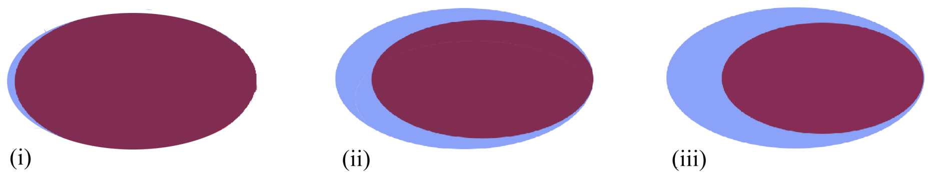


Figure 14: Representative Venn diagrams (log_2) illustrating the size relationships of each set in Table 1 to the full DAS space ($L_{max} = 12$, $H = 5$, $W = 5$). Each outer ellipse represents the full DAS space; the darker inner areas represent (i) S, (ii) S_{Ia}, and (iii) S_{Ib}.

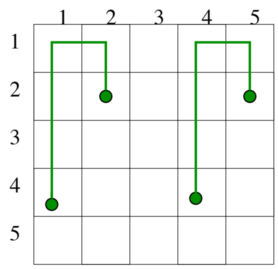


Figure 15: DAS password with repetitive symmetry and without mirror symmetry.

6 to 8 dots in a pattern when given 0.5 seconds to memorize each. Another study [9] found that the number of dots recalled in different grid sizes decreases drastically after 3 or 4 dots. Note that a user must recall two points for each stroke: the start and end points. A conservative analogy of how these studies relate to our dictionaries is to assume users naturally recall at most 4 strokes. An attacker could use this knowledge to further prioritize a dictionary and/or reduce its size. We note that all permutations of dots that lie on a cell that is cut by a reflection axis are counted in these graphical dictionaries, as each is considered an enclosed case. All permutations of dots form a significant part of the set of enclosed cases (and the full DAS password space), as the number of dot permutations for a given L_{max} is $\sum_{i=1}^{L_{max}}(W \times H)^i$. The summation counts all passwords up to length L_{max}; $(W \times H)^i$ counts all possible dot permutations of length i, as each dot is of length 1 and there are $(W \times H)$ cells that may be chosen for each dot. When all axes are used, $L_{max} = 12$, $H = 5$, and $W = 5$, the number of dot permutations is approximately 2^{56}. This is because when a password's strokes are longer for a password of fixed length, there are fewer strokes and thus fewer permutations of its composite strokes. This limitation would not restrict the overall length of the password – it could still be very long. We expect that if one models 4 as the maximum number of strokes per password, the size of the Class I memorable password space will be significantly less than our results for $L_{max} > 4$. The implication of this would be that DAS passwords may be less secure than otherwise believed.

One way to increase the password space without increasing the required password lengths would be to increase the grid size. However, this may have a negative effect on the memorability of DAS passwords, since it has been found that the recall performance of subjects decreases as a function of the grid size [9]. Alternatively, the DAS password space could be increased by adding user-selected characteristics to the drawing such as colour, backgrounds, and textures.

Although the focus of our work is the hypothetical application of a mirror symmetric graphical dictionary on the DAS scheme, this method of analysis could be applied to a variety of other graphical password schemes. For example, Birget et al. [2] propose the users be provided an image, and asked to choose a given number of click points. One could assume that a user would be more likely to choose symmetric objects in an image as click points. The same assumption might be valid for the Déjà Vu scheme [6], where the attacker would presume the user's portfolio is more likely to contain symmetric random art images.

6 Concluding Remarks

Our results suggest that a user's tendency to recall certain types of images may aid an attacker in creating a graphical dictionary for dictionary attacks against the DAS scheme. If or when graphical passwords become commonly used, this information could be used (as is textual dictionary information) in recommending password lengths and properties for graphical password users, and in performing proactive graphical password checking [28]. Studies on how users actually do use graphical password schemes would result in even more specific recommendations.

Although this analysis examines the memorability of DAS passwords from the view of the visual and temporal structure of the drawing, it does not consider other factors of DAS passwords that may affect memorability. One such factor is the number of coordinates and strokes that people can recall when given enough time (recall §5). It is unknown whether the numbers cited for the number of coordinates people recall are a function of the time given to examine the pattern. Based on our class of memorable graphical passwords, we can guess what sort of images people are likely to draw; the complexity of these images in terms of password length or number of strokes is a separate issue.

Another factor one may expect to affect memorability of a password is the temporal order of the drawing. It is still unclear as to whether the memorability benefits of pictures would be distorted due to the need to not only recall the visual image associated with the picture, but the order in which it must be input. If the temporal order is a complicating factor that adds significant complexity to what users must recall, they may be more likely to choose single-stroke (or fewer-stroke) passwords. This could also be used to an attacker's advantage, providing an improvement to the graphical dictionary of mirror symmetric graphical passwords. A conservative variation of this concept was used in our graphical dictionaries: we assumed that users would use symmetry in both a local and global scope, local being the actual stroke drawn, global being the relationship between the strokes to be a symmetric password when viewed as a whole.

We believe that this work provides a significant extension to the analysis of graphical passwords – it shows promise for the security of graphical passwords and gives incentive for their further study. This work has also raised many new and interesting questions for how to pursue research in this area (see §5), suggesting there is much room for future work, in graphical password security and in related psychological studies. Psychological studies that allow a subject unlimited or a reasonably bounded time to memorize a dot sequence or grid drawing would be useful. The results could be examined for an upper bound on how the number of dots or complexity of the drawing could affect the memorability of the pattern, and thus what password lengths people are likely to choose. Similarly, psychological studies on how temporal order affects memorability of dot patterns or grid drawings would be useful in determining the type and length of strokes people will use within their password. Studies to show how grid size affects the memorability of drawings and what sort of graphical passwords users choose in practice would be helpful. Finally, extensions or alternatives to the DAS encoding scheme may improve security by increasing the size of the resulting password space.

7 Acknowledgements

The first author acknowledges Canada's National Sciences and Engineering Research Council (NSERC) for funding her PGS A scholarship. The second author acknowledges NSERC for funding an NSERC Discovery Grant and his Canada Research Chair in Network and Software Security.

Notes

[1]Increasing the grid height and/or width will increase the dictionary size for any given length. A length of 8 is a quite simple DAS password; see example passwords in [11].

[2]Note that rectangles are a subclass of our class of memorable passwords.

[3]Note that when the defining stroke is drawn from e to s, it is considered a different defining stroke.

[4]An alternative is to use the hashed password as a cryptographic key for decrypting a check-word for authentication; this key might also be used to encrypt files.

References

[1] F. Attneave. Symmetry, Information and Memory for Patterns. *American Journal of Psychology*, 68:209–222, 1955.

[2] J.-C. Birget, D. Hong, and N. Memon. Robust Discretization, With an Application to Graphical Passwords. Cryptology ePrint Archive, Report 2003/168, 2003. `http://eprint.iacr.org/`, site accessed Jan. 12, 2004.

[3] G. H. Bower, M. B. Karlin, and A. Dueck. Comprehension and Memory For Pictures. *Memory and Cognition*, 3:216–220, 1975.

[4] M.W. Calkins. Short Studies in Memory and Association from the Wellesley College Laboratory. *Psychological Review*, 5:451–462, 1898.

[5] D. Davis, F. Monrose, and M.K. Reiter. On User Choice in Graphical Password Schemes. In *13th USENIX Security Symposium*, 2004.

[6] R. Dhamija and A. Perrig. Déjà Vu: A User Study Using Images for Authentication. In *9th USENIX Security Symposium*, 2000.

[7] R.-S. French. Identification of Dot Patterns From Memory as a Function of Complexity. *Journal of Experimental Psychology*, 47:22–26, 1954.

[8] J. Goldberg, J. Hagman, and V. Sazawal. Doodling Our Way to Better Authentication, 2002. CHI '02 extended abstracts on Human Factors in Computer Systems.

[9] S.-I. Ichikawa. Measurement of Visual Memory Span by Means of the Recall of Dot-in-Matrix Patterns. *Behavior Research Methods and Instrumentation*, 14(3):309–313, 1982.

[10] J. Nakajima and M. Matsui. Performance Analysis and Parallel Implementation of Dedicated Hash Functions. In *Advances in Cryptology – Proceedings of EUROCRYPT 2002*, pages 165–180, 2002.

[11] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin. The Design and Analysis of Graphical Passwords. *8th USENIX Security Symposium*, 1999.

[12] E. A. Kirkpatrick. An Experimental Study of Memory. *Psychological Review*, 1:602–609, 1894.

[13] D. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In *The 2nd USENIX Security Workshop*, pages 5–14, 1990.

[14] S. Madigan. Picture Memory. In John C. Yuille, editor, *Imagery, Memory and Cognition*, pages 65–89. Lawrence Erlbaum Associates Inc., N.J., U.S.A., 1983.

[15] S. Madigan and V. Lawrence. Factors Affecting Item Recovery and Hypermnesia in Free Recall. *American Journal of Psychology*, 93:489–504, 1980.

[16] F. Monrose. *Towards Stronger User Authentication*. PhD thesis, NY University, 1999. `http://www.cs.nyu.edu/csweb/Research/Theses/monrose_fabian.pdf`, site accessed January 12, 2004.

[17] A. Muffett. Crack password cracker. `http://ciac.llnl.gov/ciac/ToolsUnixAuth.html`, site accessed Jan. 12, 2004.

[18] Openwall Project. John the Ripper password cracker. `http://www.openwall.com/john/`, site accessed Jan.7, 2004.

[19] Openwall Project. Wordlists. `http://www.openwall.com/passwords/wordlists/`, site accessed Jan.7 2004.

[20] S.E. Palmer. *Vision Science: Photons to Phenomenology.* MIT Press, Cambridge, Mass., 1999.

[21] F.T. Perkins. Symmetry in Visual Recall. *American Journal of Psychology*, 44:473–490, 1932.

[22] A. Perrig and D. Song. Hash Visualization: a New Technique to Improve Real-World Security. In *International Workshop on Cryptographic Techniques and E-Commerce*, pages 131–138, 1999.

[23] B. Pinkas and T. Sander. Securing Passwords Against Dictionary Attacks. In *9th ACM Conference on Computer and Communications Security*, pages 161–170. ACM Press, 2002.

[24] S. Stubblebine and P.C. van Oorschot. Addressing Online Dictionary Attacks with Login Histories and Humans-in-the-Loop. In *Financial Cryptography'04*. Springer-Verlag LNCS (to appear), 2004.

[25] J. Thorpe and P.C. van Oorschot. Graphical Dictionaries and the Memorable Space of Graphical Passwords. Extended version (in progress): `http://www.scs.carleton.ca/~jthorpe`.

[26] C.W. Tyler. Human Symmetry Perception. In C.W. Tyler, editor, *Human Symmetry Perception and its Computational Analysis*, pages 3–22. VSP, The Netherlands, 1996.

[27] J. Wagemans. Detection of Visual Symmetries. In C.W. Tyler, editor, *Human Symmetry Perception and its Computational Analysis*, pages 25–48. VSP, The Netherlands, 1996.

[28] J. Yan. A Note on Proactive Password Checking. ACM New Security Paradigms Workshop, New Mexico, USA, 2001. `http://citeseer.nj.nec.com/yan01note.html`, site accessed Jan. 12, 2004.