Towards Secure Design Choices for Implementing Graphical Passwords *

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

Abstract

We study the impact of selected parameters on the size of the password space for "Draw-A-Secret" (DAS) graphical passwords. We examine the role of and relationships between the number of composite strokes, grid dimensions, and password length in the DAS password space. We show that a very significant proportion of the DAS password space depends on the assumption that users will choose long passwords with many composite strokes. If users choose passwords having 4 *or fewer strokes, with passwords of length* 12 *or less on a* 5×5 *grid, instead of up to the maximum* 12 *possible strokes, the size of the DAS password space is reduced from* 58 *to* 40 *bits. Additionally, we found a similar reduction when users choose no strokes of length* 1*. To strengthen security, we propose a technique and describe a representative system that may gain up to* 16 *more bits of security with an expected negligible increase in input time. Our results can be directly applied to determine secure design choices, graphical password parameter guidelines, and in deciding which parameters deserve focus in graphical password user studies.*

1. Introduction

The ubiquitous use of textual passwords for user authentication has a known weakness: users choose passwords with predictable characteristics. This is due to a user tendency to choose passwords that are easy to remember – this often means passwords which have "meaning" to the user. Unfortunately, these (likely chosen) passwords, which we will refer to as the *probable password space*, make up only an insignificant subset of the full password space. It is desirable to have users choose a wide variety of passwords, as this increases the computational expense for the known threat of the *dictionary attack*. A dictionary attack is a brute-force guessing attack where an attacker draws candidate guesses from a dictionary of "likely passwords" (often those easily remembered). If a password scheme's probability distribution is known to be non-uniform, the entropy of the password scheme is reduced. In Klein's 1990 case study [10], 25% of 14 000 user passwords were found in a dictionary of only 3×10^6 words. Additionally, the Morris Worm [24] used a dictionary, consisting of 432 words and the 1988 UNIX online dictionary, with remarkable success (some sites reported that 50% of their passwords were correctly guessed using this dictionary). This suggests that a password scheme's security is linked more closely to the size of the probable password space than that of the full password space (e.g. for 8-character passwords of digits and mixed-case letters, about 2×10^{14}).

Graphical password schemes (e.g. [9, 2, 4]) have been proposed as a plausible alternative to text-based schemes, motivated in part by the fact that humans have a remarkable capability to remember pictures. Psychological studies support that people recall pictures with higher probability than words, including those most easily interpreted to have meaning (concrete nouns) [11]. 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, the probable password space of a graphical password scheme may exceed that of text-based schemes and thus presumably offer better resistance to dictionary attacks.

The "Draw-A-Secret" (DAS) scheme [9] (reviewed in §3.1) is of particular interest as it boasts a large password space, indicating a potentially larger probable password space. Understanding how some possibly predictable characteristics affect this password space is an important step in understanding the security DAS provides. If we assume that a password *complexity property* is a property that affects password memorability (and thus the chance of selection), we would like to identify such properties and know how a high probability of users choosing such passwords would affect the size of the probable password space. We introduce a set of DAS password complexity properties based on pattern complexity factors from Attneave [1]: password length, number of composite strokes (the *stroke-*

*This paper appears in the *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC)*, Tucson, USA. December 6-10, 2004.

count), symmetry, or number of turns in each stroke. We aim to determine if these properties would reduce the size of the probable password space such that it is computationally feasible for an attacker to perform a brute-force guessing attack using a *graphical dictionary* (i.e. an attack dictionary against a graphical password scheme) [26]. In this case, the graphical dictionary would be composed of DAS passwords in the probable password space, ordered from most to least probable.

We examine subsets of the DAS scheme's password space defined by the length of the password (see §3.1) and its stroke-count. The number of turns in a stroke is a complexity property that appears to warrant its own study. The symmetry complexity property has been examined [26]; that analysis also observed that many passwords are the result of permuting fixed sets of stroke combinations. For example, the size of the DAS password space includes all permutations of very short strokes such as dots (single-celled strokes) and 2-cell lines, a subtlety not immediately apparent from the original DAS paper [9]. There are 2^{56} possible passwords consisting solely of dots when $L_{max} = 12$ on a 5×5 grid (see §3.2) – a surprisingly large proportion. Our examination is motivated in large part by this observation and curiosity regarding the effect on the password space of limiting the number of strokes in DAS passwords to e.g. 3 or 4. We show that the size of the DAS password space decreases significantly with fewer strokes (for a fixed password length).

An attacker could use this knowledge to prioritize and/or reduce the size of a graphical dictionary. We determine the security impact by estimating the amount of time required to exhaust such a dictionary and find that such a dictionary (under reasonable parameter choices) could be exhausted using one 3.2GHz machine in just over 1 day. We consider the required graphical password length and stroke-count so that the graphical password space outsizes the corresponding space of textual passwords. We believe these results are significantly more important than those we recently presented [26] related to "memorable" DAS passwords, where memorable is taken to correspond with visual symmetry.

To counter the reduction of the DAS password space that results from user-selected passwords with a small stroke-count, we examine the effect of the seemingly obvious compensatory measure of increasing the grid size. We show that increasing the size of the grid that DAS passwords are drawn in has low security pay-back, assuming that users choose DAS passwords with a small stroke-count. In response, we propose a technique with potentially high security pay-back to increase the password space and discuss how it could increase the uncertainty of the password space by approximately 16 bits.

Our contributions include the identification of graphical password complexity properties and their relationship to the DAS password space, an analysis and better understanding of the DAS password space [9], the creation of graphical dictionaries that differs significantly from the existing literature [26], an understanding of how much security can be gained by increasing grid size, and the proposal of an enhancement to the DAS scheme from which we expect higher security pay-back. Our work on graphical password complexity properties and their relationship to the DAS password space, and on graphical dictionaries could be used to help in formulating password rules for DAS graphical password users and in creating proactive graphical password checkers. Our work on measuring how much benefit can be achieved by increasing grid size and by our proposed enhancement to DAS can be used to increase the security of DAS implementations.

The sequel is organized as follows. §2 discusses related work. §3 explores the relationship between selected complexity properties and the DAS password space, and discusses their security implications. §4 analyzes how increasing the grid size affects the DAS password space and proposes a technique to increase the password space. §5 provides observations and further discussion. Concluding remarks are made in §6.

2. Related Work

The security for a password scheme can be measured in terms of its resistance to dictionary attack. To prevent online dictionary attacks, Pinkas and Sander discuss human-in-the-loop methods [21]; see also Stubblebine et al. [25]. One defence against off-line dictionary attacks is to reduce the probability of cracking through enforcing password policies and proactive password checking. Yan discusses some popular proactive textual password checkers [28] such as *cracklib*. To perform effective proactive textual password checking, it is important to understand available textual password cracking dictionaries and tools (e.g. *Crack* [16] and *John the Ripper* [18]).

The graphical password schemes proposed to date can be generally categorized as recognition-based or recall-based. One recognition based scheme using hash visualization [20] was implemented in a program called Déjà Vu [4]. 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. Another recognition-based scheme called $Passfaces$ [22] requires that a user select a set of human faces as their password. Similar to Déjà Vu, the user is expected to correctly select each of the faces in their password from a set (or sets) of presented faces. A scheme similar to *Passfaces*, called the "Story" scheme [3], requires a user to select a sequence of images (of e.g. food, animals, sports, automobiles, scenic locations, and people) that depict a story.

Recall-based schemes include that proposed by Birget et al. [2], which requires a user to click on several points on a background picture, and the DAS scheme ([9]; see §3.1), which uses user-defined drawings. Both schemes are exactly repeatable (as defined within each scheme), allowing 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 [9] to be larger than that of the full textual password space. In our analysis (see §3), 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 and user choice for graphical passwords, Davis et al. [3] examine user choice in 2 recognition-based graphical password schemes. Particular to the DAS scheme, Jermyn et al. argue that the DAS scheme has a large memorable password space by modelling user choice using short programs to describe passwords, and combinations of 1 or 2 rectangles. They show that the number of rectangle combinations is comparable to the size of many textual password dictionaries. A separate user study on memorability performed by Goldberg et al. [6] showed that people are less likely to recall the order in which they drew a DAS password than the resulting image.

Our recent analysis of graphical passwords [26] suggested that people are likely to choose graphical passwords that are easy to recall, as appears to be the case with textual passwords. This work postulated a class of memorable graphical passwords based on visual mirror symmetry, supported by a collection of cognitive studies, and used this class to analyze DAS. This analysis observed that the number of permutations of short strokes (of length 1) compose a large number of DAS passwords, motivating the current paper in large part.

3. Analysis of Complexity Properties in DAS

To contribute towards a security evaluation of DAS, we determine how the DAS graphical password space is affected by the maximum stroke-count X, in addition to the maximum password length L_{max}. The aim of this analysis (§3.2) is to better understand where DAS gains its large password space. Our results (§3.3) show that the majority of its strength lies in passwords with a significant stroke-count. The security implications of this are discussed in §3.4. A review of DAS is provided in §3.1.

3.1. Review of DAS

In the DAS graphical password scheme [9, 15], a user enters a password by creating a simple drawing on a $G \times G$ grid (e.g. Fig. 1). DAS benefits from decoupling the position of the input from the temporal order, creating a larger number of passwords than when they are coupled. This gives DAS an advantage over textual password schemes with keyboard input (where the temporal order in which characters are typed predetermines their position).

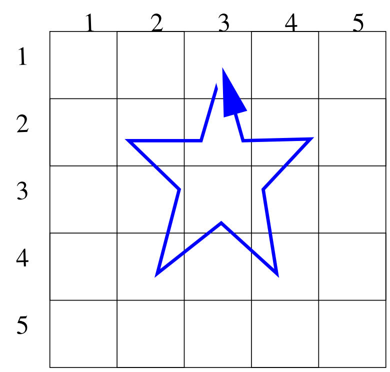

Figure 1. Example DAS password.

Each grid cell is referenced by two-dimensional coordinates $(x, y) \in [1 \ldots G] \times [1 \ldots G]$. As the input device passes through the grid, the sequence of coordinate pairs through which it passes are added to the DAS password encoding. A "pen-up" event (when the input device, e.g. a Tablet PC pen, is lifted from the grid surface) is represented by the distinguished coordinate pair $(G+1, G+1)$. If two drawings have the same encoding (i.e. they crossed the same sequence of grid cells with pen-up events in the same places in the sequence) they are considered equivalent. All drawings that have the same encoding belong to the same *equivalence class*. The encoding of the example password in Fig. 1 is: (3,1), (3,2), (2,2), (2,3), (2,4), (3,4), (3,3), (3,4), (4,4), (4,3), (4,2), (3,2), (3,1), pen-up.

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. Finally, the *length of a password* is the sum of the lengths of its strokes (excluding pen-ups). The example DAS password in Fig. 1 shows 1 stroke of length 13, thus the entire password is of length 13.

DAS disallows passwords considered difficult to repeat exactly (e.g. passwords involving pieces lying close to a grid boundary). Any stroke is invalid if any part of a stroke is indiscernible as to which cell it lies within.

Jermyn et al. [9] compute the size of the full password space, i.e. the number of distinct DAS graphical password encodings. It is assumed that all passwords of total length greater than a fixed value L_{max} have probability zero. They recursively compute the size of the full password space for passwords of total length $\leq L_{max}$. For $L_{max} = 12$ and a 5×5 grid, this is 2^{58}, surpassing the number of textual passwords of 8 characters or less constructed from the printable ASCII codes (2^{53}).

3.2. Quantifying DAS Subsets

We quantify the relationship between DAS password space and the length of passwords and their composite strokes. Our motivation is based on an observation [26] that all permutations of dots are counted in the DAS password space (e.g. see Fig. 2). This forms a large number of passwords, as the number of dot permutations for a given L_{max} on a $W \times H$ grid is $\sum_{i=1}^{L_{max}} (W \times H)^i$. The summation counts all dot permutations for passwords less than or equal to L_{max}, and $(W \times H)^i$ counts all possible dot permutations of length i (each dot is of length 1 and there are $(W \times H)$ cells that may be chosen for each dot). When $L_{max} = 12$, $H = 5$, and $W = 5$, the number of dot permutations is approximately 2^{56} (compared to a full password space of $2^{57.7}$). Intuitively, this result is sensible since if a password of fixed length has longer composite strokes, it must have a smaller stroke-count, and thus fewer permutations of its composite strokes.

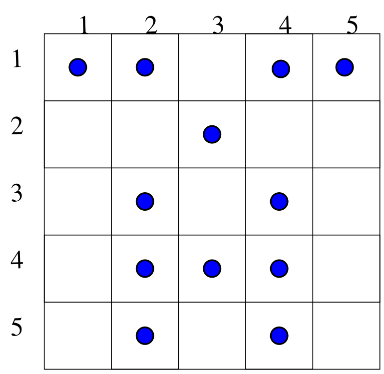

Figure 2. Example DAS password consisting entirely of dots (single-celled strokes).

Our general approach to quantify the relationship between DAS and the stroke-count is to determine how many DAS passwords are of length at most a given maximum password length L_{max}, with a maximum stroke-count of X. Counting all passwords of length at most L_{max} follows [9]; we add the restriction of the maximum stroke-count X within the function *P* (see below).

We modify the function *P(L,G)* [9] that counts the number of passwords of length $\leq L$ (where $1 \leq L \leq L_{max}$ is the password length and G is the side dimension of a square grid), to limit the stroke-count in each password to at most X.

$$P(L, G, X) = \begin{cases} 0 & \text{if } X = 0 \text{ and } L > 0 \\ 1 & \text{if } X \geq 0 \text{ and } L = 0 \\ \sum_{\ell=1}^{L} N(\ell) \cdot P(L - \ell, G, X - 1) & \text{otherwise} \end{cases} \tag{1}$$

P defines the cardinality of the set of passwords with X or fewer strokes, of total password length at most L. *P* is defined recursively in terms of *N(ℓ)* (see [9]), which gives the number of *strokes* of length ℓ. In what follows, we use (1) to determine the relevant number of passwords.

3.3. Subset Size Results

We focus our discussion on a set of results for a 5×5 grid size, giving the bit-size of the password space for passwords of length less than or equal to L_{max} (from 1 to 20) and selected maximum stroke-counts X. The full tabulated set is provided in the extended version of this paper [27].

Fig. 3 shows the effect (log_2) of increasing L_{max} for selected increments of X: the password space's size increases exponentially, illustrating the roles of both L_{max} and X in the DAS password space. Note that the left ends of all but the line representing the full password space ($X = L_{max}$) have been omitted for simplicity – we know that the maximum stroke-count for a password of length L_{max} is L_{max}, thus any line where $X > L_{max}$ will have the same value as when $X = L_{max}$.

Increasing X accounts for at least one half of the bit-size (see the difference between the $X = 1$ line and the full space line, when $L_{max} \geq 5$). The top line, where $X = L_{max}$, in Fig. 3 shows what one would likely expect from reading the original DAS paper [9] (e.g. 58 bits when $L_{max} = 12$). The other thick line, where $X = 4$, is probably (see §3.4) more representative of user choice in DAS passwords, assuming all passwords composed of 4 or fewer strokes are equiprobable. This graph illustrates the role of strokes in the DAS password space; the size of the password space is significantly smaller (40 bits as opposed to 58 bits for the full space) if users choose a password of length at most 12, composed of 4 or less strokes. The password space size still increases with longer password lengths (as shown by the rise in each curve), but the amount of increase is less for smaller stroke-counts (as shown by the

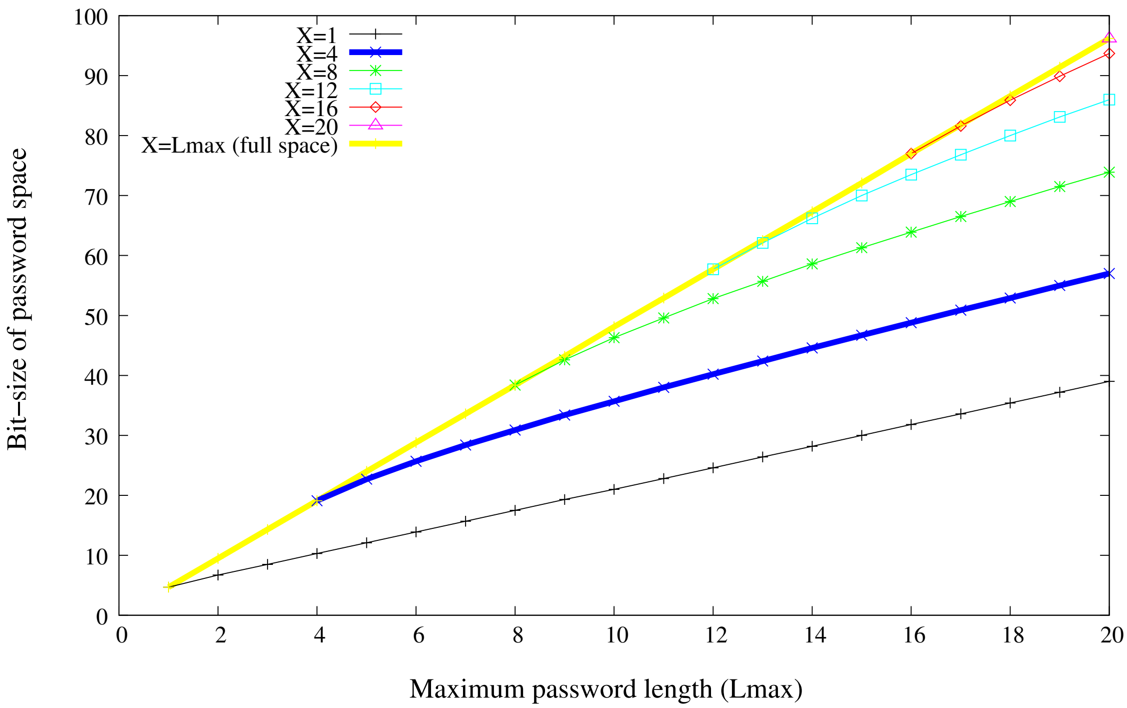

Figure 3. Size of graphical password space for passwords of at most X strokes (for a 5 by 5 grid and a fixed maximum password length Lmax).

more gradual slopes for lines with smaller values of X). Note that for a fixed L_{max}, a smaller maximum stroke-count X (lower lines in Fig. 3) implies a longer average stroke length.

Fig. 4 illustrates the true proportion of passwords (non-logarithmic) when $L_{max} = 12$ with a given stroke-count, showing the large number of permutations that occur when the stroke-count is greater than 6. The proportion of passwords with a stroke-count of 6 or less is so small in comparison that it is not visible on the pie graph. The reason that there are more passwords composed of 11 strokes than 12 is that 12-stroke passwords are restricted to all permutations of 12 dots, versus all permutations of 10 dots and a single stroke of length 2. The number of permutations of 12 dots is $25^{12} = 5.96 \times 10^{16}$, whereas the number of permutations of 10 dots times the number of strokes of length 2 (80 for a 5×5 grid) and temporal order within the password for the stroke of length 2 (11) is $25^{10} \times 880 = 8.4 \times 10^{16}$.

The strength of DAS is gained from taking into account temporal order in terms of the direction of the strokes, and more importantly, the order in which these strokes are drawn. This explains why increasing the stroke-count achieves such gains in the size of the password space: there are many more permutations of these strokes.

We note that all of the visible password space shown in Fig. 4 is accounted for by passwords for which $X > \lfloor \frac{L_{max}}{2} \rfloor$, the point when there must be at least one dot (or similar single-celled stroke) in the password. When $X = \lfloor \frac{L_{max}}{2} \rfloor$, the only combination of strokes that does not include a dot is when all strokes are of length 2 (with 1 of length 3 when L_{max} is odd); thus when $X > \lfloor \frac{L_{max}}{2} \rfloor$, at least 1 of these strokes of length 2 must be broken into 2 of length 1 (or if L_{max} is odd, the stroke of length 3 could be broken into 1 of length 2, and another of length 1).

This leads us to ask: how much of the total password space consists of passwords resulting from seemingly unlikely combinations of very short strokes, i.e. entirely of length 1 and/or 2? We examined this by restricting the formula for counting the number of possible strokes such that strokes larger than length 1 (and another set of results for strokes larger than 2) returned 0, ensuring they were not counted. The results were interesting: passwords composed entirely of strokes of length 1 comprise approximately $\frac{1}{4}$ of the total password space, and passwords composed of only strokes of length ≤ 2 comprise approximately $\frac{1}{2}$ of the password space. This might be examined from another angle: how much of the total password space consists of passwords *without* any strokes of length 1? We found that if users do not draw any strokes of length 1 in their DAS password, the size of the password space when $L_{max} = 12$ on a 5×5 grid is effectively reduced from 58 to 40 bits (i.e. a few millionths of the full password space).

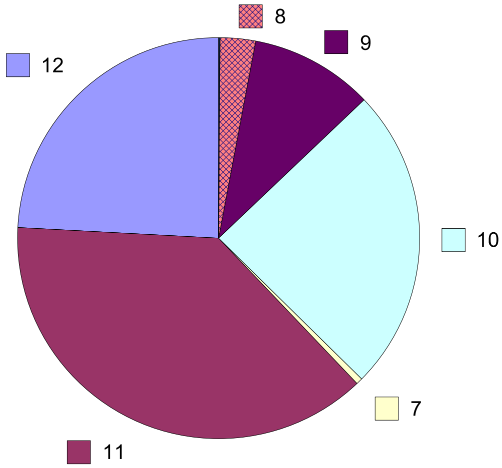

Figure 4. Proportion of password space attributable to passwords consisting of exactly X strokes. Here the maximum password length is 12, on a 5 by 5 grid. Note that for 6 or fewer strokes, the proportion is so small it is not visible.

3.4. Security Implications

Our results show that the stroke-count in a DAS password is quite important. Note that the space of DAS passwords when $L_{max} = 12$, restricted to at most 4 strokes, or alternatively, with no strokes of length 1, is only approximately 40 bits; 40 bit searches are now considered very easy in practice given modern processor speeds. If users often choose passwords with a small stroke-count, or with no strokes of length one, the security of the DAS scheme could easily be compromised by a dictionary attack, trying all passwords with at most 4 strokes, or alternatively those with no strokes of length 1. An attacker could also use this knowledge to further prioritize a dictionary (e.g. to the subset of symmetric passwords described in [26], which we suspect would decrease in size if limited to a maximum stroke-count of 4).

Unfortunately (from a security perspective), there is some evidence suggesting that users may tend to choose passwords with a small stroke-count. One psychological study [5] has shown that people optimally recall 6 to 8 dots in a pattern when given 0.5 seconds to memorize each. Another study [7] 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. Thus a conservative analogy of how these studies relate to DAS graphical passwords is to assume users naturally recall at most 4 strokes.

This motivated the examination of an informal user study of 16 students [17] for user preference in terms of the stroke-count in the user-chosen DAS passwords. This examination, although very small, suggested a user tendency to choose passwords with a small stroke-count: 80% of users chose passwords with a stroke-count of 1-3 and 90% chose passwords with a stroke-count of 6 or less.

We conjecture that this is attributable to the information that a user must recall to reproduce their password. For each stroke, a user must recall a start point, an end point, and a visual pattern connecting the two. The only part of this that appeals to visual memory is the pattern itself; the location of the start and end point are additional pieces of information (unless the user can recall the visual pattern(s) and grid together as one image; however, the requirement for a user to recall the order in which strokes are entered may detract from this possibility). We also conjecture that this requirement would encourage users to choose fewer composite strokes in their password, given free choice.

If our conjectures are true in the user population, the probability distribution of the DAS password space is highly non-uniform, which is of course highly undesirable as the entropy of the password space is significantly reduced. If users tend to choose passwords with certain characteristics that define a relatively small subset of the full password space, their probabilities are higher than in a uniform distribution, adversely affecting entropy. Entropy can be used for theoretical analysis [23, 13, 12], however we find it less useful for placing this work in practical context.

To provide context for the practical implications of our results, we discuss in §3.5 how long it might take to exhaust a DAS password dictionary consisting simply of all passwords of length ≤ 12, on a 5×5 grid, for each maximum stroke-count X.

3.5. Times for Exhaustive Attacks

To highlight the practical implications of the results shown in Fig. 3, we present our results in terms of the time to exhaust a DAS password dictionary containing all passwords of length ≤ 12, on a 5×5 grid, for each maximum stroke-count X.

The exact method used to perform any 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 function, then comparing the hashed password to the password file entry for the user.[1] 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.

[1] An alternative is to use the hashed password as a cryptographic key for decrypting a check-word for authentication or to encrypt files.

Here the attack time is at least the time to hash each candidate password. Thus, we 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 1000 such machines, with which linear speed-up is achieved. It is reasonable to consider that a determined attacker could exploit 1000, or even 100 000 machines using a worm, to distribute the password-cracking load. Using a MD5 performance result of 3.66 cycles/byte for a *Pentium 3* 800 MHz machine [8] (which we extrapolate for 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 time to generate the password hashes for comparison is given in Table 1.

Table 1. Estimated time to exhaust various dictionaries using 3.2GHz machines (5 by 5 grid, maximum password length of 12). Note that X=12 corresponds to the full DAS space.

Maximum no. of strokes (X)	Time to exhaust (1 machine)	Time to exhaust (1000 machines)
12	541.8 years	197.8 days
11	409.7 years	149.5 days
10	205.3 years	74.9 days
9	72.6 years	26.5 days
8	18.1 years	6.6 days
7	3.2 years	1.2 days
6	157.1 days	3.8 hours
5	14.9 days	21.4 mins
4	1.1 days	1.5 mins
3	1.2 hours	4.4 seconds
2	2.3 minutes	0.1 seconds
1	1.9 seconds	0.002 seconds

The times provided in Table 1 highlight the implications of the graphical dictionary size, and the importance of not choosing passwords with a low stroke-count. If users choose passwords of length at most 12, with a stroke-count of at most 4, an attacker could guess their password using one machine in only 1.1 days. If we want an attacker to require an average of 10 years to exhaust a dictionary using 1000 computers with the above mentioned resources, the minimum dictionary size must be approximately 2^{63}. Assuming that users choose the conjectured 4 or fewer strokes, referring to our tabulated data (see [27]) when $X = 4$, the DAS password space is only above this size when a 10×10 grid is used and $L_{max} \geq 19$. This implies that for this level of security, it is advisable to require either more composite strokes or a larger grid size (as discussed in §4) in conjunction with longer passwords.

Some of the larger "successful" textual password dictionaries contain approximately 4×10^7 entries [19]. If an attacker uses a dictionary of passwords containing 4 or fewer strokes, it still exceeds this number of textual dictionary entries when a 5×5 grid is used and $L_{max} \geq 6$. This implies that even if users choose passwords composed of 4 or fewer strokes, provided the password length is at least 6, the DAS scheme may still offer greater security than textual passwords against dictionary attacks.

4. Increasing size of DAS Password Space

In discussions of DAS [9, 26], increasing the password space by increasing the grid size is briefly mentioned, but not quantified. Increasing the grid size may have a negative effect on the memorability of DAS passwords, since it has been found that the recall performance of subjects (for dot patterns on grids) decreases as a function of the grid size [7]. In §4.1, we examine the impact of the grid size on the DAS password space. We present a technique to increase the DAS password space using grids with (we expect) minimal inconvenience to the user in §4.2. These results can be used in determining practical design choices for implementing DAS.

4.1. Effect of Grid Size Increase

We computed sets of results with 3 control variables: maximum stroke length L_{max} (from 1 to 20), maximum stroke-count X (from 1 to 20 or password length), and grid dimension (5, 6, 7, and 10, assumed to be squared). We justify stopping our data when the grid dimensions are 10×10 due to size limitations on the interface; to have each grid cell (the level of user-input error tolerated) with $1cm \times 1cm$ dimensions, we have a $10cm \times 10cm$ input area. This would already outsize the screen dimensions of many PDAs (a likely candidate device for the DAS scheme), and it is already a fairly complex grid in terms of what users must recall.

We display results for $L_{max} = 12$ and selected values of X in graphical form in Fig. 5. The full tabulated sets are provided in the extended version of this paper [27]. Fig. 5 shows that the password space increases with the grid size regardless of the X values, but not as much as anticipated. It appears that the bulk of the password space growth (22.6 bits) occurs when $X = L_{max}$ (the top line), but when $X < \lfloor \frac{L_{max}}{2} \rfloor$, we only gain 5-11 more bits by increasing the grid area from 25 cells (5×5 grid) to 100 cells (10×10 grid). The bulk of the password space growth when $L_{max} = 16$ and 20 was also found to occur between

$X = \frac{L_{max}}{2}$ and $X = L_{max}$. Since we know that passwords composed of short strokes make up a large proportion of the password space when $X \geq \lfloor \frac{L_{max}}{2} \rfloor$, this result indicates that the bulk of the gain achieved by increasing the grid size applies to passwords with a large number of small strokes. Additionally, the password space growth achieved by increasing the grid size for a fixed X is comparable when $L_{max} = 12$ and 20, indicating a negligible advantage for increasing both L_{max} and grid size unless it means a larger number of strokes are used. In practice, unless we can somehow ensure through password rules (see §5) that users will choose a large number of strokes, increasing the grid size provides low security pay-back.

If we follow the expected example of a user choosing a password of length 12, composed of 4 or fewer strokes, the increase in the password space by increasing the grid size from a 5×5 grid to a 10×10 grid is comparable to the increase achieved by keeping a 5×5 grid and increasing the stroke-count to 7, or keeping the number of strokes and increasing L_{max} to 17. Furthermore, asking users to use only dots as composite strokes for their passwords (which we do not recommend for usability reasons) on a 5×5 grid gives 5 more bits than increasing the grid size to 10×10 and allowing users to choose a small stroke-count (≤ 4).

The result sets given here and in §3.3 show that the stroke-count X has a larger potential impact on the effective bit-size of the DAS password space, than the oft-mentioned password length and grid dimensions. The question remains as to which of these parameters will have the least negative impact on users' ability to recall their passwords. It is entirely possible that users would prefer to use a larger grid rather than using passwords with a large number of short strokes and/or of greater length. In response to the dissapointing results of the grid size increase, we propose a way of simulating a large grid size while minimizing the impact on the user in §4.2.

4.2. Using Grid Selection

We suspect that increasing the grid size (recall §4.1) to increase the DAS password space does not provide enough security pay-back to compensate for the increased difficulty to reproduce a password. Alternatively, we propose the general technique of a *selection grid* (an initial large, fine-grained grid from which the user selects a *drawing grid*, a rectangular region to zoom in on, in which they may enter their password; for an illustrative representation of the idea, see Fig. 6). In Fig. 6, the selection grid is on the left, the drawing grid on the right. Our selection grid in Fig. 6 is checkered with grey and white 5×5 sub-grids to aid the user in locating specific cells. The idea of zooming in on an area is similar to that discussed by Birget et al. [2], except we are zooming in on a grid to draw in, not a picture to click a point within. This general technique, which we call *grid selection*, could add up to another 16 bits (see below) to the password space, with (we suggest) minimal inconvenience to the user. Using grid selection, we suspect the benefits of an increased password space would be combined with a minimal increase in input time and stress on the user's memory, while remaining within the limits of the input display. We now discuss how many more passwords can reasonably be added to the DAS password space by a grid selection implementation.

We first consider a reasonable size for the selection grid. Assuming a $10cm \times 10cm$ input area[2], and that one cm^2 can be partitioned into 9 cells and maintain approximately the same resolution as 10pt font, a 30×30 selection grid resolution would be reasonable. A user does not have to draw their password in this grid, but select their drawing grid (with a pre-specified area range, e.g. between 25 and 100 cells) by e.g. selecting two of its opposite four corner cells. If a user is permitted to select any drawing grid size, the benefits of this approach may be diminished as we would expect small drawing grids to be selected. Thus, a restriction for the minimum drawing grid width D_W or height D_H must be defined; for this example, we assume 5. Alternatively, we want the user to be able to easily reproduce their drawing, thus a reasonable level of error-tolerance (achieved by the grid resolution) is necessary. Thus, a restriction on how large the grid can be must also be defined; we assume that $1cm^2$ cells provide a desirable level of error tolerance giving a maximum D_W and D_H of 10.

In order to choose a drawing grid from the selection grid, the user must select a starting point $p_s = (x, y)$ and an ending point $p_e = (x, y)$ (the difference between p_s and p_e defines D_W and D_H). There are 31^2 possible values for p_s. Depending on the location of p_s on the selection grid, there are different possible values of p_e (and thus D_W and D_H). We determine the number of possible grids for each possible p_s using our assumed parameters of 30×30 selection grid, a minimum drawing grid width of 5, and a maximum drawing grid width of 10. Assuming the input order of p_s and p_e matter, these parameters give a total number of possible drawing grids to be $79\,524$, which could add 16 bits to our DAS encodings.

Of course, a grid selection implementation may be used in a predictable manner: e.g. users may be more likely to choose drawing grids that trace the 5×5 checker squares of the selection grid. If these options are disallowed, we lose $36 \times 4 = 144$ grids. If we disallow all square grids, we lose 13324 grids, leaving $79\,524 - 13324 = 66\,200$ drawing grids. Thus, in either case, we could still achieve a gain of approximately 16 bits. To perform a brute-force attack, an

[2]Note these dimensions could be larger for a larger input screen on e.g. a tablet PC (see [14]), or smaller for a PDA.

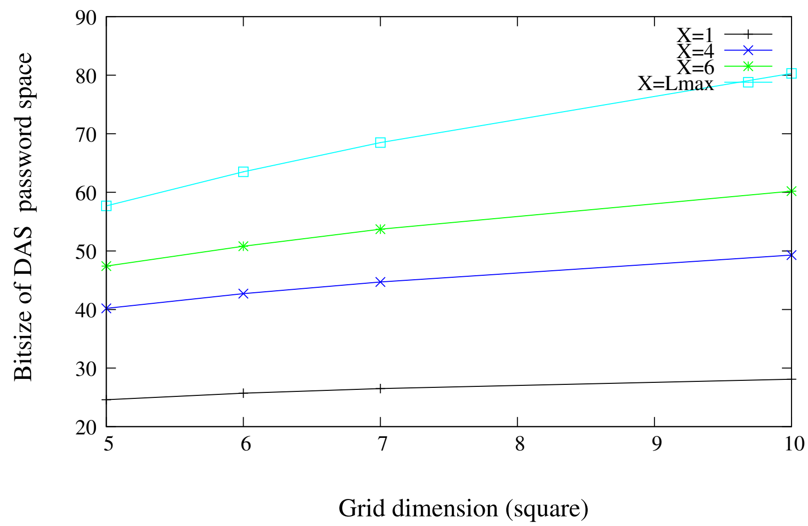

Figure 5. Effect of grid size on bit-size of DAS password space for a maximum password length of 12 and selected values of X.

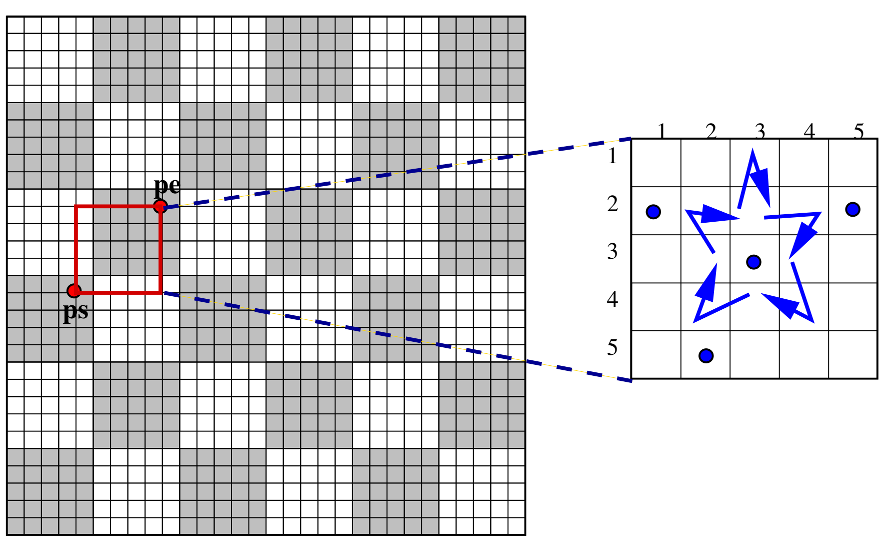

Figure 6. Grid selection: a user selects a drawing grid in which to draw their password.

attacker must run an attack against each possible drawing grid selectable from the selection grid. This discussion is of course only illustrative of the idea – there are possibly more "likely" grids that should be disallowed. Alternatively, to gain more grids, we could allow users to select any set of cells from the selection grid, then normalize the drawing grid size to be within the pre-specified area range.

5. Observations and Further Discussion

Our study of the DAS password space shows that of the complexity properties examined, the stroke-count in a password has the largest impact. The length also has a significant impact on the size of the password space, but its impact is not as strong as the stroke-count. From Fig. 3 we note that increasing the stroke-count by 1 (when there are less than $\frac{L_{max}}{2}$ strokes) results in more possible passwords than increasing L_{max} by 1.

The results of how assuming either a small stroke-count or assuming no strokes of length 1 affect the size of the password space (recall §3.3) suggest the use and enforcement of DAS password rules. Given our current understanding of the DAS password space, we suggest that users choose passwords with a stroke-count of at least $\lfloor \frac{L_{max}}{2} \rfloor$. We also recommend at least one composite stroke be of length 1. Finally, given results from another study [26], we suggest the user avoids global reflective (mirror) symmetry in their passwords. An example of a recommended DAS password is shown in Fig. 6. One of the dots in this password is placed such that the overall picture is not globally symmetric, the length is 19, and it is composed of 9 strokes (exactly $\lfloor \frac{19}{2} \rfloor$).

We believe that increasing the grid size (recall §4.1) to increase the full DAS password space does not provide enough security pay-back to compensate for the increased information a user must recall to reproduce their password. Alternatively, using new implementation techniques (such as that detailed in §4.2) may increase the bit-size of the DAS password space with higher security pay-back. To obtain higher security pay-back, such implementation techniques are required to maintain acceptable user input times and a low overhead increase in what a user must recall, while retaining an acceptable level of error-tolerance.

6. Concluding Remarks

We believe that this work significantly extends and compliments existing analysis/understanding of DAS graphical passwords – comparing the bit-size of the probable password space shows that a more viable graphical password attack strategy follows from our present results than that of using symmetry alone [26]. We believe that without taking these results into consideration, the practical security of DAS implementations may be over-estimated. We postulated a preliminary set of DAS password rules based on our analysis (recall §5) and determined that one obvious method to increase the password space (i.e. greater grid granularity) appears to be less effective than previously believed. Better knowledge of relative sizes of DAS password space subsets (defined by password characteristics) motivates the study of relationships between human memory and password complexity properties to obtain a more secure implementation.

It is possible that greater effective security may be achieved by graphical password schemes having larger probable password spaces, even if at the expense of a smaller full (theoretical) password space. Encouraging users to draw passwords with more strokes might result in an increase to the size of the probable password space, by reducing the difficulty for a user to recall their password. This could be achieved by e.g. having the direction of strokes not matter. A better understanding of the breakdown of what users have the most difficulty recalling (leading to a more formal definition of DAS password complexity properties) would be beneficial to understanding how to strengthen DAS implementations. We base our definition of complexity properties (recall §1) on a particular psychology study by Attneave [1], which provides some hints by its examination of memory and complexity factors for visual patterns.

Further study is required to determine how complexity properties (e.g. grid dimensions, password length, number and direction of composite strokes) impact memorability and user choice in passwords. Psychological and user studies could be examined for how the complexity properties of drawings affect memorability, giving direction as to which complexity properties may be relaxed to encourage users to choose passwords consisting of more strokes. Alternatively, research is required to determine whether mnemonic strategies exist for graphical passwords to aid memorization of complex graphical passwords. Research to determine how such mnemonic strategies affect memorability, similar to that performed by Yan et al. [29] for textual passwords, would be useful. Our work highlights the need for psychological and user studies to understand more about what users recall best, and how to take advantage of the strengths of human memory to create a graphical password implementation and set of guidelines that work in practice.

Finally, there is room to enhance the underlying encoding scheme to obtain a larger DAS password space. Many DAS passwords are considered invalid due to ambiguity as to which cells compose the user-drawn stroke (e.g. strokes on grid lines or passing through grid cell corners). An encoding scheme that would allow these disallowed passwords may significantly increase the size of the DAS password space.

7. Acknowledgements

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

References

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

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

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

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

[7] 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.

[8] 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.

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

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

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

[12] J. Massey. Guessing and Entropy. In *ISIT: Proceedings IEEE International Symposium on Information Theory*, page 204, 1994.

[13] R. McEliece. *The Theory of Information and Coding*, volume 3 of *Encyclopedia of Mathematics and its Applications*, chapter Entropy and Mutual Information, pages 15–46. Addison-Wesley Publishing Company, 1977.

[14] Microsoft. Microsoft XP Tablet PC Edition. http://www.microsoft.com/windowsxp/tabletpc, site accessed May 14, 2004.

[15] F. Monrose. *Towards Stronger User Authentication*. PhD thesis, NY University, 1999.

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

[17] D. Nali and J. Thorpe. Analyzing User Choice in Graphical Passwords. Tech. Report TR-04-01, School of Computer Science, Carleton University, Canada, 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] 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.

[21] 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.

[22] Real User Corporation. About Passfaces. http://www.realuser.com, site accessed May 24, 2004.

[23] C. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[24] E. Spafford. Crisis and Aftermath (The Internet Worm). *Comm. of the ACM*, 32(6):678–687, 1989.

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

[26] J. Thorpe and P. van Oorschot. Graphical Dictionaries and the Memorable Space of Graphical Passwords. In *13th USENIX Security Symposium*, 2004.

[27] J. Thorpe and P. van Oorschot. Towards Secure Design Choices for Implementing Graphical Passwords (Extended Version). http://www.scs.carleton.ca/~jthorpe, 2004.

[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.

[29] J. Yan, A. Blackwell, R. Anderson, and A. Grant. The Memorability and Security of Passwords – Some Empirical Results. Technical Report No. 500,Computer Laboratory,University of Cambridge, 2000. http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/tr500.pdf, site accessed September 6, 2004.