Exploiting Predictability in Click-based Graphical Passwords

P.C. van Oorschot and Julie Thorpe*

Abstract

We provide an in-depth study of the security of click-based graphical password schemes like PassPoints (Weidenbeck et al., 2005), by exploring popular points (hot-spots), and examining strategies to predict and exploit them in guessing attacks. We report on both short- and long-term user studies: one lab-controlled, involving 43 users and 17 diverse images, the other a field test of 223 user accounts. We provide empirical evidence that hot-spots do exist for many images, some more so than others. We explore the use of human-computation (in this context, harvesting click-points from a small set of users) to predict these hot-spots. We generate two human-seeded attacks based on this method: one based on a first-order Markov model, another based on an independent probability model. Within 100 guesses, our first-order Markov model-based attack finds 4% of passwords in one image data set, and 10% of passwords in a second image data set. Our independent model-based attack finds 20% within 2^{33} guesses in one image data set and 36% within 2^{31} guesses in a second image data set. These are all for a system whose full password space has cardinality 2^{43}. We also evaluate our first-order Markov model-based attack with cross-validation of the field study data, which finds an average of 7-10% of user passwords within 3 guesses. We also begin to explore some click-order pattern attacks, which we found improve on our independent model-based attacks. Our results suggest that these graphical password schemes (with parameters as originally proposed) are vulnerable to offline and online attacks, even on systems that implement conservative lock-out policies.

1 Introduction

Traditional text-based authentication suffers from a well-known limitation: many users tend to choose passwords that have predictable patterns, allowing for successful guessing attacks. As an alternative, graphical passwords require a user to remember an image (or parts of an image) in place of a word. They have been largely motivated by the well-known fact that people remember images better than words [26], and implied promises that the password spaces of various image-based schemes are not only sufficiently large to resist guessing attacks, but that the effective password spaces (from which users actually choose) are also sufficiently large. The latter, however, is not well established.

Many different types of graphical passwords have been proposed to date; among the more popular approaches in the literature is *PassPoints* [47, 46, 45, 1, 13]. It and other click-based graphical password schemes [2, 22, 38, 9, 5] require users to click on a sequence of points on one or more background images. PassPoints usability studies have been performed to determine the optimal amount of error tolerance based on click-point accuracy [46, 8], login and creation times, login error rates, memorability, and general perception [46, 47, 8]. An important remaining question for such schemes is: how *secure* are they? This issue has previously remained largely unaddressed, despite speculation that the security of these schemes likely suffers from hot-spots (areas of an image that are more probable than others for users to click.

The issue of whether hot-spots exist is tightly related to that of the security; if commonly preferred points exist, then they could be exploited in a number of ways. We confirm the existence of hot-spots, and show that some images are more susceptible to hot-spotting than others. Our work involves two user studies. The first (lab) study used 17 diverse images. In the second (field) study, involving 223 user accounts over a minimum of seven weeks, we explored two of these images in greater depth. We analyzed our lab study data

Manuscript received November 7, 2008; revised July 28, 2010; accepted August 4, 2010. Parts of this work appeared previously in [42] and in the Ph.D. thesis [41] of the second author.

*Authors listed alphabetically. Contact author: Julie Thorpe (julie.thorpe@uoit.ca). She is with the Faculty of Business and Information Technology, University of Ontario Institute of Technology (UOIT), Oshawa, Ontario, Canada. P.C. van Oorschot is with the School of Computer Science, Carleton University, Ottawa, Ontario, Canada, (e-mail: paulv@scs.carleton.ca).

using estimates of formal measures of security to make an informed decision of which two images to use in the field study.

We explore how an attacker might *predict* the hot-spots we observed for use in an offline dictionary attack. Rather than using image processing to predict hot-spots (see discussion under Related Work), we instead use human computation [44], which relies on people to perform tasks that computers (at least currently) find difficult. Human-computation can produce a *human-computed data set*; our human-computed data set is our lab study data set, which effectively indexes the click-points that people would initially choose as part of a password. We process this data set to determine a set of points that are more commonly preferred, to create a *human-seeded* attack. A human-seeded attack can be generally defined as an attack generated by using data collected from people.

We create three different predictive graphical dictionaries [31] (i.e., based on available information related to the user login task, gathered from sources outside of the target password database itself, where a target password database is the set of user passwords under attack): two based on different styles of human-seeded attacks, and another based on click-order patterns. We evaluate these dictionaries, and also combined human-seeded and click-order pattern attacks, using our field study data set. We also perform a 10-fold cross-validation analysis with our field study database to train and test one style of human-seeded attack (based on a first-order Markov model), providing a sense of how well an attacker might do with these methods and an ideal human-computed data set for training.

Our contributions include an in-depth study of hot-spots in click-based (and cued-recall) graphical password schemes, and the impact of these hot-spots on security through two separate user studies. We explore predictive methods of generating attack dictionaries for click-based graphical passwords. Perhaps our most interesting contribution is proposing and exploring the use of human-computation to create graphical dictionaries; we conjecture that this method is generalizable to other types of graphical passwords (e.g., recognition-based) where users are given free choice.

The remainder of this paper proceeds as follows. Section 2 presents relevant background and terminology. Section 3 describes our user studies and hot-spot analysis. Section 4 describes algorithms and methods for creating predictive attacks. Section 5 presents results for all attacks examined herein. Section 6 discusses related work, and we conclude with Section 7.

2 Background and Terminology

Click-based graphical passwords require users to log in by clicking a sequence of points on one or more background images. Many variations are possible (see Section 6), depending on the number of images and what points a user is allowed to select. We study click-based graphical passwords by allowing clicks anywhere on a single image (i.e., PassPoints-style). To allow password verification, user-entered passwords must be encoded in some standard format to allow verification. Assuming that the encoding (e.g. robust discretization [1] or centered discretization[7]) is followed by some form of hashing to preclude trivial attacks, offline attacks [7] are still possible if hashed values are intercepted by an attacker and can be used as verifiable text, or if the attacker obtains a file of system-side verification values.

We use the following terminology. Assume a user chooses a given click-point c as part of his or her password. The *tolerable error* or *tolerance* t is the error (in pixels) allowed for a click-point entered on a subsequent login to be accepted as c. This defines a *tolerance region (T-region)* centered on c, which for our experimental implementation using t = 9 pixels, is a 19×19 pixel square. A *cluster* is a set of one or more click-points that lie within a T-region. Note that clusters arise when the data from multiple users is combined, rather than a single user clicking multiple times in the same area. Our algorithm for computing clusters is described in Section 3.2.1. The number of click-points falling within a cluster is its *size*. A *hot-spot* is indicated by a cluster that is larger than expected by random choice, in an experiment which produces click-points across a set of T-regions. To aid visualization and indicate relative sizes for clusters of size at least two, on figures we sometimes represent the underlying cluster by a shaded circle or *halo* with halo diameter proportional to its size (similar to population density diagrams). An *alphabet* is a set of distinct T-regions; our experimental implementation, using 451×331 pixel images, results in an alphabet of at least m = 414 non-overlapping T-regions. Using passwords composed of 5-clicks on an alphabet of size 414 provides the system with only 2^{43} entries in the full theoretical password space; however, increasing the

number of clicks, size of the image, and/or decreasing the tolerance square size would allow for comparable security to traditional text passwords. We study an implementation with these particular parameters as they are close to those used in other studies [45, 47] that show them to have acceptable usability.

3 User Studies

As mentioned, we conducted two user studies: a single session lab study with 43 users and 17 images, and a long-term field study with 223 user accounts and two images. We use the lab study data as an indicator of the degree of hot-spotting for each image, and as our human-computed data set. We use the field study data to test our attacks. Further details of the lab and field studies are in Section 3.1, the hot-spotting results in Section 3.2, and the user studies' limitations are discussed in Section 3.3.

3.1 Experimental Methodology

We report on the methodology for the short-term lab study in Section 3.1.1 and the long-term field study in Section 3.1.2.

3.1.1 Lab Study

Here we report the details of a university-approved 43-user study of click-based graphical passwords in a controlled lab environment. Each user session was conducted individually and lasted about one hour. Participants were all university students who were not studying (or experts in) computer security. Each user was asked to create a click-based graphical password on 17 different images (most of these are reproduced in Figures 1 and 11; others are available upon request). Four of the images are from a previous click-based graphical password study by Wiedenbeck et al. [46]; the other 13 were selected to provide a range of values based on two image processing measures that we expected to reflect the amount of detail: the number of segments found from image segmentation [14] and the number of corners found from corner detection [19]. Seven of the 13 images were chosen to be those we intuitively believed would encourage fewer hot-spots; this is in addition to the four chosen in earlier research by others [46] using intuition (no further details were provided on their image selection methodology).

We implemented a browser-based lab tool for this study. Each user was provided a brief explanation of what click-based graphical passwords are, and given two images to practice creating and confirming such passwords. To keep the parameters as consistent as possible with previous usability experiments[1] of such passwords [47], we used 5 click-points for each password, an image size of 451×331 pixels, and a 19×19 pixel square of error tolerance. Wiedenbeck et al. [47] used a tolerance of 20×20, allowing 10 pixels of tolerated error on one side and 9 on the other. For consistent error tolerance on all sides, we approximate this using 19×19. Users were instructed to choose a password by clicking on 5 points, with no two the same. Although the software did not enforce this condition, subsequent analysis showed that the effect on the resulting cluster sizes was negligible for all images except *pcb*. For *pcb*, considering all click-points produced 6 clusters of size ≥ 5, but counting at most one click from each user produced 3 clusters of size ≥ 5. We did not assume a specific encoding scheme (e.g., robust discretization [1] or other grid-based methods [7]); the concept of hot-spots and user choice of click-points is general enough to apply across all encoding schemes. To allow for detailed analysis, we stored and compared the actual click-points.

Once users had a chance to practice a few passwords, the main part of the lab experiment began. For each image, users were asked to create a click-based graphical password that they could remember but that others will not be able to guess, and to pretend that it is protecting their bank information. After initial creation, users were asked to confirm their password to ensure they could repeat their click-points. On successful confirmation, users were given 3D mental rotation tasks [33] as a distractor for at least 30 seconds (to remove the password from their visual working memory, and thus simulate the effect of the passage of time). After this period of memory tasks, users were provided the image again and asked to log in using their previously selected password. If users could not confirm after two failed attempts during password creation/confirmation or log in after one failed attempt, they were permitted to reset their password for that

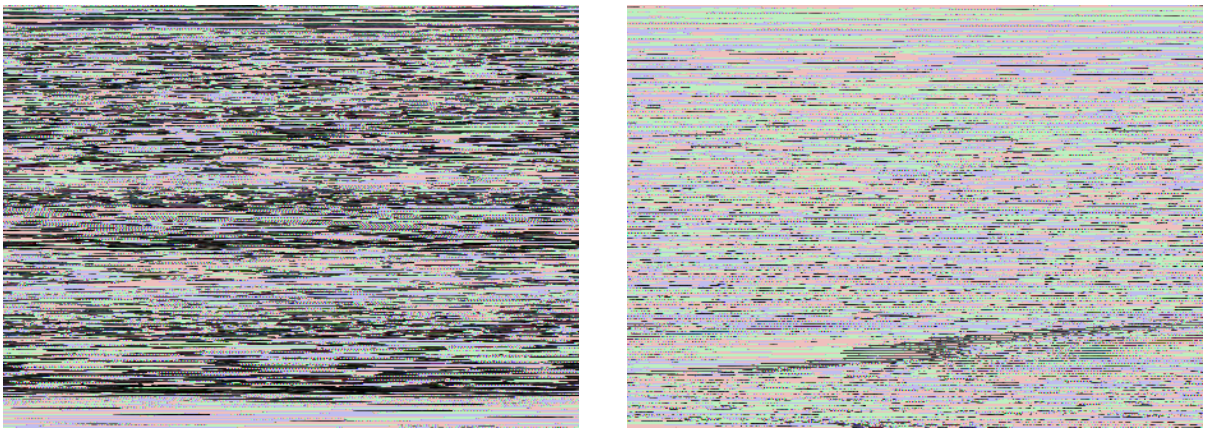
[1] The usability aspects of this study are reported separately [8].

image and try again. If users did not like the image and felt they could not create and remember a password on it, they were permitted to skip the image. Only two of the 17 images had a signi‰nt number of skips: *paperclips* and *bee*. This suggests some passwords for these images were not repeatable, and we suspect our results for these images would show lower relative security in practice.

To avoid any dependence on the order of images presented, each user was presented a random (but unique from other users) shu ed ordering of the 17 images used. Since most users did not make it through all 17 images, the number of graphical passwords created per image ranged from 32 to 40, for the 43 users. Two users had an inaccurate mouse, but we do not expect this to a ect our present focus ‰he location of selected click-points. This short-term lab study was intended to collect data on initial user choice; although the mental rotation tasks work to remove the password from working memory, this study does not account for any e ect caused by password resets over time due to forgotten passwords. For this reason, we use the long-term ‰d study (Section 3.1.2) which does account for this e ect, as the primary data set for testing the success of our attack dictionaries.

3.1.2 Field Study

Here we describe a university-approved ‰d study of 223 user accounts on two di erent background images. We collected click-based graphical password data to evaluate the security of this style of graphical passwords against various attacks. We used the entropy and expected guesses measures from our lab study to choose two images that would apparently o er di erent levels of security (although both are highly detailed): *pool* and *cars* (see Figure 1). The lab study showed that of the images used in previous studies [46], the *pool* image had the closest to a middle ranking in terms of the amount of clustering (see Section 3.2.2). The lab study also showed that the *cars* image had nearly the least amount of clustering among the 17 images tested. Both images had a low number of skips in the lab study, indicating that they did not cause problems for users with password creation. We chose the *pool* image so we had an image from previous studies and also had an amount of clustering that was not extremely high or low (it was closest to the middle rank of the images examined). We chose the *cars* image to give this scheme the best chance we could in terms of anticipated security.



(a) *cars* (originally from [4]). (b) *pool* (originally from [46, 47]).

Figure 1: Images used in lab study.

Our web-based implementation of PassPoints was used by three ‰st-year undergraduate classes: two ‰st year courses for computer science students, and a ‰st year course for non-computer science students enrolled in a science degree. The students used the system for at least 7 weeks to gain access to their course notes, tutorials, and assignment solutions. For comparison with previous usability studies, and our lab study, we used an image size of 451 ‰331 pixels. After the user entered their username and course, the screen displayed their background image and a small black square above the image to indicate their tolerance square size. For about half of users (for each image), a 19 ‰19 T-region was used, and for the other half, a 13 ‰13

T-region.[2] The system enforced that each password had 5 clicks and that no click-point was within $t = 9$ pixels of another (vertically and horizontally). Each user was assigned an image at random. To complete initial password creation, users had to successfully confirm their password once. After initial creation, users were permitted to reset their password at any time using a previously set secret question and answer.

Users were permitted to login from any machine (home, school, or other), and were provided an online FAQ and help. The users were asked that they keep in mind that their click-points are a password, and that while they will need to pick points they can remember, they should not pick points that someone else will be able to guess. Each class was also provided a brief overview of the system, explaining that their click-points in subsequent logins must be within the tolerance shown by a small square above the background image, and that the input order matters. In order to have some confidence that the passwords we analyze have some degree of memorability, we only use the final passwords created by each user that were demonstrated as successfully recalled in at least one subsequent login (after the initial create and confirm). We also only use data from 223 out of 378 accounts, as this was the number that provided explicit consent as required by university policy. These 223 user accounts map to 189 distinct users as 34 users in our study belonged to two classes; all but one of these users were assigned a different image for each account, and both accounts for a given user were set to have the same error tolerance. Of the 223 user accounts, 114 used *pool* and 109 used *cars* as a background image.

3.2 Hot-Spot Results

We present the hot spots found in both the lab and field studies. How we compute hot-spots is described in Section 3.2.1, as well as the hot-spots discovered in the lab study. A comparison of hot-spotting across different lab study images is provided in Section 3.2.2. Finally, the hot-spots discovered in the field study are presented in Section 3.2.3.

3.2.1 Hot-Spots Computed from Lab Study Data

We collected data from the in-lab study as described in Section 3.1.1, and used a clustering algorithm (see below) to determine a set V of (non-empty) clusters and their sizes.

Clustering Algorithm. To calculate clusters (the size of which defines hot-spots) based on any user data set of raw click-points, we assign all of the observed user click-points to clusters as follows. Let R be the raw (unprocessed) set of click-points, M a list of temporary clusters, and V the final resulting set of clusters. M and V are initially empty.

1. For each $c_k \in R$, let C_k be a temporary cluster containing click-point c_k. Temporarily assign all user click-points in R within c_k's T-region to C_k. Add C_k to M. Each $c_k \in R$ can thus be temporarily assigned to multiple clusters C_k.
2. Sort all clusters in M by size, in decreasing order.
3. Greedily make permanent assignments of click-points to clusters as follows. Let C be the largest cluster in M. Permanently assign each click-point $c_k \in B$ to C, then delete each $c_k \in B$ from all other clusters in M. Delete C from M, and add C to V. Repeat until M is empty.

This process determines a set V of (non-empty) clusters and their sizes. We then calculate the observed probability p_j (based on our data set) of the cluster j being clicked, as cluster size divided by total clicks observed.

To begin comparing the 17 images studied, Figure 2 shows the sizes of the top 10 most popular clusters, and the total number of popular clusters.

Given the cluster sizes, we then calculate the observed probability p_j (based on our user data set) of the cluster j being clicked, as cluster size divided by total clicks observed. When the probability p_j of a certain cluster is sufficiently high, we can place a confidence interval around it for future populations (of users who are similar in background to those in our study) using formula (1) as discussed below.

Each probability p_j estimates the probability of a cluster being clicked for a *single* click. For 5-click passwords, we approximate the probability that a user chooses cluster j in a password by $5p_j$. Note that the

[2] Analysis showed little difference between the points chosen for these different tolerance groups.

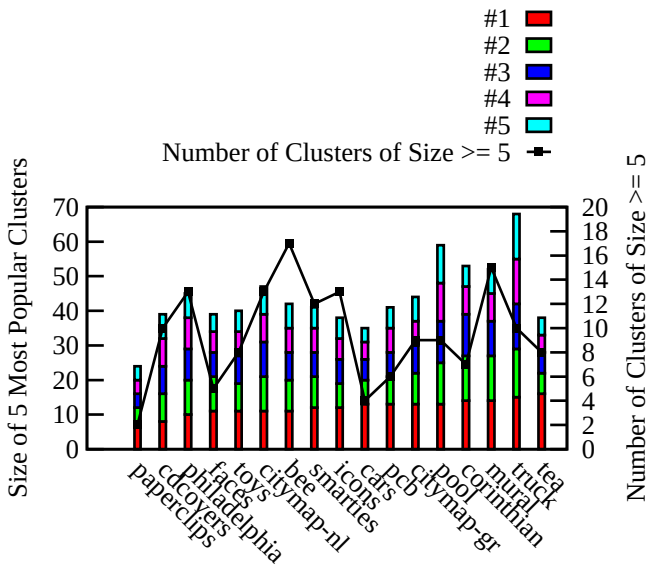


Figure 2: The most popular clusters (in terms of size, i.e., number of times selected), and number of popular clusters (of size ≥ 5). Results are from 32-40 users, depending on the image, for the final passwords created on each image.

probability for a cluster j increases slightly as other clicks occur (due to the constraint of 5 distinct clusters in a password); we ignore this in our estimate.

Our results in Figure 2 indicate a significant number of hot-spots for our sample of the full population (32–40 users per image). Previous conservative assumptions [47] were that half of the available alphabet of T-regions would be used in practice (or 207 in our case. If this were the case, and all T-regions in the alphabet were equi-probable, we would expect to see some clusters of size 2, but none of size 3 after 40 participants; we observed significantly more on all 17 images. Figure 2 shows that some images were clearly worse than others in terms of the amount of hot-spotting. There were many clusters of size at least 5, and some as large as 16 (see *tea* image). If a cluster in our lab study received 5 or more clicks (in which case we call it a *popular* or *high-probability* cluster) then statistically, this allows determination of a confidence interval, using formula (1) which provides the $100(1-\alpha)\%$ confidence interval for a population proportion [12, page 288].

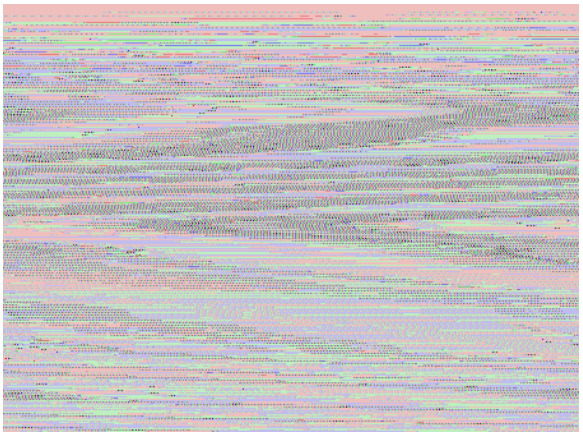
$$p \pm z_{\alpha/2} \sqrt{\frac{pq}{m}} \qquad (1)$$

Here m is the total number of clicks (i.e., 5 times the number of users), p takes the role of p_j, $q = 1-p$, and $z_{\alpha/2}$ is from a z-table. A confidence interval can be placed around p_j (and thus $5p_j$) using (1) when $mp \geq 5$ and $mq \geq 5$. For clusters of size $k \geq 5$, $p = \frac{k}{m}$, then $mp = k$ and $mq = m-k$. In our case, $m \geq 32 \cdot 5$ and $m-k \geq 5$, as statistically required to use (1).

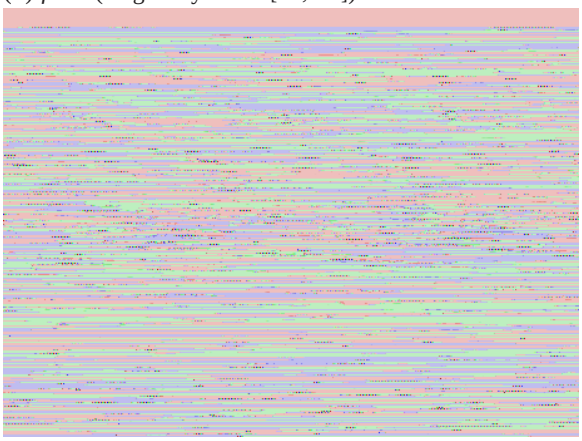
Table 1 shows these confidence intervals for four images, predicting that in future similar populations many of these points would be clicked by 10-50% of users, and some points would be clicked by 20-60% of users with 95% confidence ($\alpha = .05$). For example, in Table 1(a), the first row shows the highest frequency cluster (of size 13); as our sample for this image was 35 users, we observed approximately 37.1% of our participants choosing this cluster as part of their password. Using (1), between 17.7% and 56.6% of users from future populations are expected to choose this same cluster (with 95% confidence).

Figure 2 and Table 1 show the popularity of the hottest clusters; Figure 2's line graph also shows the number of popular clusters. The clustering effect evident in Figures 2, 3, and Table 1 clearly establishes that

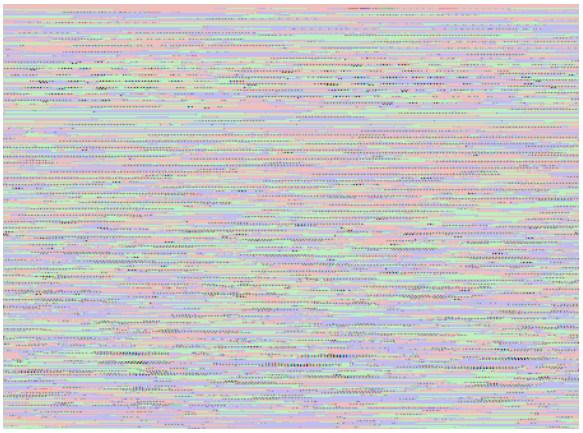
(a) *pool* (originally from [46, 47]).



(b) *mural* (originally from [46]).



(c) *philadelphia* (originally from [46]).



(d) *truck* (originally from [15]).

Figure 3: Observed click-points from lab study. Halo diameters are 10 times the size of the underlying cluster, illustrating cluster popularity.

(a) *pool* image			(b) *mural* image		
Cluster size	$5p_j$	95% CI ($5p_j$)	Cluster size	$5p_j$	95% CI ($5p_j$)
13	0.371	(0.177; 0.566)	14	0.400	(0.199; 0.601)
12	0.343	(0.156; 0.530)	13	0.371	(0.177; 0.566)
12	0.343	(0.156; 0.530)	10	0.286	(0.114; 0.458)
11	0.314	(0.134; 0.494)	8	0.229	(0.074; 0.383)
11	0.314	(0.134; 0.494)	7	0.200	(0.055; 0.345)

(c) *philadelphia* image			(d) *truck* image		
Cluster size	$5p_j$	95% CI ($5p_j$)	Cluster size	$5p_j$	95% CI ($5p_j$)
10	0.286	(0.114; 0.458)	15	0.429	(0.221; 0.636)
10	0.286	(0.114; 0.458)	14	0.400	(0.199; 0.601)
9	0.257	(0.094; 0.421)	13	0.371	(0.177; 0.566)
9	0.257	(0.094; 0.421)	13	0.371	(0.177; 0.566)
7	0.200	(0.055; 0.345)	13	0.371	(0.177; 0.566)

Table 1: 95% confidence intervals for the top 5 clusters found in each of four images. The two numbers separated by semicolons represent the lower and upper bounds on the probability that users are expected to choose this cluster in future populations.

hot-spots are very prominent on a wide range of images. We further pursue how these hot-spots impact the practical security of full 5-click passwords in Section 4.2. As a partial summary, our results suggest that many images have significantly more hot-spots than would be expected if all T-regions were equi-probable. The *paperclips*, *cars*, *faces*, and *tea* images are not as susceptible to hot-spotting as others (e.g., *mural*, *truck*, and *philadelphia*). For example, the *cars* image had only 4 clusters of size at least 5, and only one with frequency at least 10. The *mural* image had 15 clusters of size at least 5, and 3 of the top 5 frequency clusters had frequency at least 10. Given that our sample size for the *mural* image was only 36 users, these clusters are suprisingly popular. This demonstrates the range of effect the background image can have.

While previous work [46] suggests using intuition for choosing more secure background images, our results show that intuition may not always be a good indicator. Of the four images used in other click-based graphical passwords studies, three showed a large degree of clustering (*pool*, *mural*, and *philadelphia*). Furthermore, two other images that we intuitively believed would be more secure background images were among the worst (*truck* and *citymap-nl*). The *truck* image had 10 clusters of size at least 5, and the top 5 clusters had frequency at least 13. Discussing criteria for image selection is outside of the scope of this paper.

Given these remarks, we next explore the impact of hot-spotting across images to help choose two images for further analysis.

3.2.2 Measurement and Comparison of Hot-Spotting for Different Images

To compare the relative impact of hot-spotting on each image studied, we calculated two formal measures of password security for each image: entropy $H(X)$ per equation (2), and the expected number of guesses $E(f(X))$ per equation (3), to correctly guess a password assuming the attacker knows the probabilities $w_i > 0$ for each password i. The relationship between $H(X)$ and $E(f(X))$ for password guessing is discussed by Massey [27]. Of course in general, the w_i are unknown, and our study gives only very coarse estimates; nonetheless, we found it helpful to use this to develop an estimate of which images will have the least impact from hot-spotting. For (2) and (3), n is the number of passwords (of probability > 0), random variable X ranges over the passwords, and $w_i = Prob(X = x_i)$ is calculated as described below.

$$H(X) = \sum_{i=1}^{n} w_i \log(w_i) \tag{2}$$

$$E(f(X)) = \sum_{i=1}^{n} i \cdot w_i, \text{ where } w_i \geq w_{i+1}, \text{ and} \quad (3)$$

$f(X)$ is the number of guesses before success.

We calculate these measures based on our observed user data. For this purpose, we assume that users will choose from a set of click-points (following the associated probabilities), and combine 5 of them randomly. This assumption almost certainly over-estimates both $E(f(X))$ and $H(X)$ relative to actual practice, as it does not consider click-order patterns or dependencies. Thus, popular clusters likely reduce security by even more than we estimate here.

We define P_{clstr} to be the set of all 5-permutations derivable from the clusters resulting from our user study data set (as computed in Section 3.2.1). Using the probabilities p_j of each cluster, the probabilities w_i of each password in P_{clstr} are estimated as follows. Pick a combination of 5 observed clusters $j_1, \ldots, j_5$ with respective probabilities $p_{j1}, \ldots, p_{j5}$. For each permutation of these clusters, calculate the probability of that permutation occurring as a password. Due to our lab study instructions that no two click-points in a password can fall in the same T-region, these probabilities change as each point is clicked. Thus, for password $i = (j_1, j_2, j_3, j_4, j_5)$, $w_i = (p_{j1} \cdot p_{j2}/(1 - p_{j1})] \cdot p_{j3}/(1 - (p_{j1} + p_{j2}))] \cdot \ldots$).

The resulting set P_{clstr} is a set of click-based graphical passwords (with associated probabilities) that coarsely approximates the effective password space if the clusters observed in our lab study are representative of those in larger similar populations. We can order the elements of P_{clstr} using the probabilities w_i based on our lab study. An ordered P_{clstr} could be used as the basis of an attack dictionary; this ordering could be much improved, for example, by exploiting expected patterns in click-order as in Section 4.2.

For further comparison to previous conservative estimates that half of the available click-points (our T-regions) would be used in practice, we calculate P_{uni} as follows. P_{uni} is the set of all 5-permutations of clusters we expect to find after observing 32 users, assuming click-points are selected independently and uniformly at random from an alphabet of size 207. We use P_{uni} as a comparison baseline that approximates what we would expect to see after running 32 users (the lowest number of users we have for any image), if previous estimates were accurate, and T-regions were equi-probable.

We use entropy and expected number of guesses as an estimate of the security (measured in bits). Fig. 4 depicts the entropy and expected number of guesses for P_{clstr}. Notice the range between images, and the drop in $E(f(X))$ from P_{uni} to values of P_{clstr}. Comparison to the marked P_{uni} values for (1) $H(X)$, and (2) $E(f(X))$, indicates that previous rough estimates are a security overestimate for practical security in all images, some much more so than others. This is at least partially due to click-points not being equi-probable in practice (as illustrated by hot-spots), and apparently also due to the previously suggested effective alphabet size (half of the full alphabet) being an overestimate. Indeed, a large alphabet is a big part of the theoretical security advantage that these graphical passwords have over text passwords. If the effective alphabet size is not as large as previously expected, or is not well-distributed, then we should reduce our expectations of the security.

These results appear to provide fair approximation of the entropy and expected number of guesses for the larger set of users in the field study; we performed these same calculations again using the field study data, with the following results. For both of the two images, the entropy measures were within one bit of values computed for the lab study (less than a bit higher for *pool*, and about one bit lower for *cars*). The expected number of guesses required using the field study data increased for both images (by 1.3 bits for *cars*, and 2.5 bits for *pool*).

The variation across all images shows that the background image can have a significant impact, even when using images that are intuitively good to some people. For example, the image that exhibited the most hot-spotting was the *mural* image, chosen for an earlier PassPoints usability study [46]. We note that the *paperclips* image scores best in the charted security measures (its $H(X)$ measure is within a standard deviation of P_{uni}); however, 8 of 36 users who created a password on this image could not perform the subsequent login (and skipped it as noted earlier), so the data for this image represents some passwords that are not repeatable, and thus we suspect it would have lower relative security in practice.

Overall, we conclude that image choice can have a significant impact on the resulting security, and that developing reliable methods to filter out images that are the most susceptible to hot-spotting would be an interesting avenue for future research. We used our computed values of these formal measures to make an

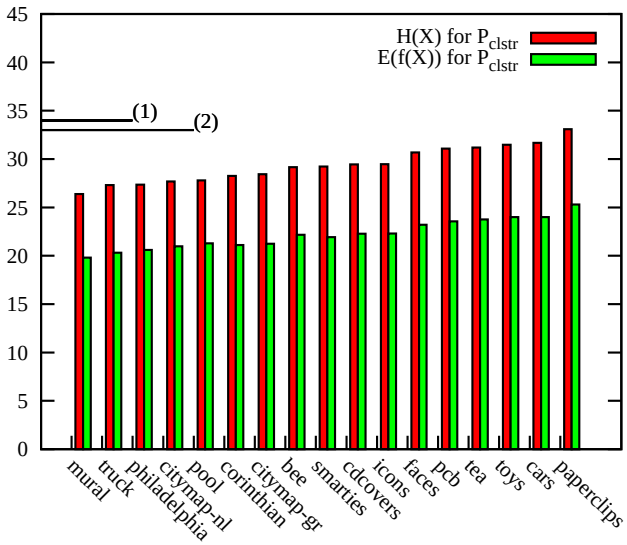


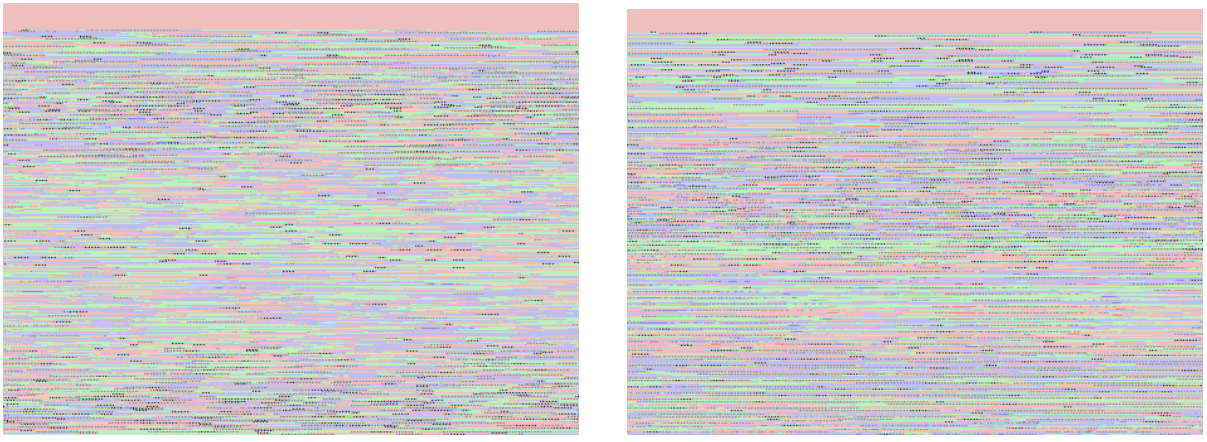
Figure 4: Security estimates for each image (in bits). P_{clstr} is based on data from the lab study of 32–40 passwords (depending on image). For comparison to a uniform distribution, (1) marks $H(X)$ for P_{uni}, and (2) marks $E(f(X))$ for P_{uni}.

informed decision on which images to study further in our field study.

3.2.3 Field Study Hot Spots and Relation to Lab Study Results

Here we present the clustering results from the two images used in the field study, and compare results to those on the same two (of 17) images from the lab study.

Fig. 5b shows that the areas that emerge as hot-spots in the lab study (cf. Fig. 3a) were also popular in the field study, but other clusters also began to emerge. Fig. 5a shows that even our best image from the lab study (in terms of apparent resistance to clustering, after eliminating an image with poor memorability) also exhibits a clustering effect after gathering 109 passwords. Table 2 provides a closer examination of the clustering effect observed.



(a) *cars* (originally from [4]). (b) *pool* (originally from [46, 47]).

Figure 5: Observed clustering (field study). Halo diameter is 5 times the number of underlying clicks.

Image Name	Size of most popular clusters					number of clusters of size ≥ 5
	# 1	# 2	# 3	# 4	# 5	
cars	26 (24%)	25 (23%)	24 (22%)	22 (20%)	22 (20%)	32
pool	35 (31%)	30 (26%)	30 (26%)	27 (24%)	27 (24%)	28

Table 2: Most popular clusters (lab study). The number of user accounts was 114 (*pool*) and 109 (*cars*).

Table 2 shows that on *pool*, there were 5 points that 24-31% of users chose as part of their password. On *cars*, there were 5 points that 20-24% of users chose as part of their password. The clustering on the *cars* image indicates that even highly detailed images with many possible choices have hot spots. Indeed, we were surprised to see a set of points that were this popular, given the small amount of observed clustering on this image from our smaller lab study.

The prediction intervals calculated from our lab study (recall Section 3.1.1) provide reasonable predictions of what we observed in the field study. For *cars*, 3 out of the 4 popular clusters fell within the 95% prediction interval. For *pool*, 8 out of the 9 popular clusters fell within the 95% prediction interval. The anomalous cluster on *cars* was still quite popular (chosen by 12% of users); the lower end of the lab study's prediction interval for this cluster was 20%. The anomalous cluster on *pool* was also still quite popular (chosen by 18% of users); the lower end of the lab study's prediction interval for this cluster was 19%.

Our studies allow us to comment on what fraction of all T-regions on an image will be chosen by users. After collecting 570 and 545 points, we only observed 111 and 133 unique clusters (for *pool* and *cars* respectively); thus, one quarter to one third of all T-regions seems to be a reasonable estimate for highly detailed images, and the relative probabilities of these regions should be expected to vary quite considerably.

3.3 Limitations of User Studies

As with all user studies, it is important to discuss possible limitations. There are differences between our lab and field studies, which is reflected in the amount of clustering observed between studies for the *cars* image, but the clustering for the *pool* image is quite similar in each study. More details regarding these differences are discussed in Section 5.5. Here we discuss possible reasons for these differences.

One possible reason for the differences in user choice between the two studies that the field study users may not have been as motivated as the lab study users to create difficult to guess graphical passwords. The field study users were using their passwords to protect class notes, whereas the lab study users were asked to pretend the password was for banking. It is unclear how a user might measure whether they are creating a graphical password that is difficult to guess, and whether in trying, if users would actually change their password's strength; one study [36] found that only 40% of users actually change the complexity of their text passwords according to the security of the site.

Another equally possible explanation might be that the lab study users chose more difficult passwords than they would have in practice, as they were aware there was no requirement for long term recall, and also did not have a chance to forget and subsequently reset their passwords to something more memorable. In the field study, users had a requirement for long-term recall.

Another possible reason might be that the user's task focus in the lab study had an influence, such that they were more motivated to create a more complex password than they might be in a regular usage environment. Finally, demographic differences between the lab user group and students in the field study classes may have been a contributing factor.

With our current data, it seems unlikely that we can conclusively determine a reason for these differences. Despite any differences, Section 5 illustrates that there is still enough similarity between the two groups to launch effective attacks as discussed in Section 4.1.

4 Attack Methodology

We used our single-session lab study data (recall Section 3.1.1) as a human-computed data set containing raw user click-points. This set of click-points is first reduced to a set of unique points and associated

probabilities, using the clustering algorithm described immediately below. We then use these unique points to generate different styles of human-seeded graphical dictionaries as described in Section 4.1. We also examined dictionaries based only on click-order patterns in Section 4.2. Finally, we perform a cross-validation analysis of one of our human-seeded methods using real field study user passwords (see Section 4.3). Our experimental results for each of these dictionaries is given in Section 5.

4.1 Human-Seeded Attacks

We are interested in predicting the hot-spots that users will choose in their passwords, for use in a guessing attack. Human-seeded attacks are motivated by the following two conjectures. Regardless of the degree to which these conjectures are true, by using them as guiding principles we were able to demonstrate working attacks, which is our objective herein, rather than proving or disproving these conjectures per se.

Conjecture 1 *Since an arbitrary group of people is likely to collectively prefer some areas of an image, the aggregate effect across a group of users will be that a significant subset choose click-based graphical passwords composed of some points that have a higher collective preference across another group of users.*

Conjecture 2 *The most popular points selected in a human-computed data set can be used to distinguish points of higher collective preference across another group of users.*

Our *human-seeded* attacks use a human-computed data set as input to generate an attack. We refer to a human-seeded dictionary as one whose passwords are composed of click-points corresponding to T-regions that users prefer, as defined by popular points observed in a human-computed data set.

We examine two different methods of generating human-seeded attacks based on a human-computed data set of click-points: one assuming that cluster probabilities are independent (Section 4.1.1), and the other assuming that cluster probabilities are dependent only on the previous click-point (Section 4.1.2).

4.1.1 Independent Cluster Probabilities

Here we assume that each click-point in a password is independent of the other click-points. The final dictionary, denoted P_{clstr} contains all 5-permutations of the final clusters in V as computed by the clustering algorithm above. The probability of each 5-permutation in P_{clstr} is defined by the product of the probability of its 5 composite clusters, whose individual probabilities p_j are derived from a human-computed data set. We call our particular implementation of this human-seeded dictionary S_{hs-ind}.

4.1.2 First-Order Markov Model

A method that has been used by text password cracking software (e.g., [29]) is to use Markov models of language (under the predictive assumption that users will choose passwords from their language), and to generate passwords using bi- or tri-grams from that language, ordered by decreasing probability. Here, using our human-computed data set for training, we create a human-seeded dictionary for PassPoints that we call S_{hs-dep}, based on a first-order Markov model. The main difference from the similar method of Davis et al. [11] (see discussion in Section 6) is the use of this human-computed data set (instead of a real password database from the same population), and as such our S_{hs-dep} is much more easily launched by an attacker. We also separately perform an experiment (on the field study data set) similar to the random sub-sampling experiment of Davis et al. [11], as outlined in Section 4.3.

To use a Markov model for our purposes, we assume that each click-point in a click-based graphical password depends only on the previous click-point. To capture this dependency, we created bi-grams based on the passwords collected in the lab study. In this work, our bi-gram is an ordered pair of click-points; each 5-click password will produce four bi-grams. We further assume that bi-grams are more likely to occur at the specific positions in which they were observed within the training passwords; for example, if the pair of (x,y) pixel coordinates [(100, 100), (200, 200)] was only observed as the first bi-gram in a password, we assume that it is more likely that it will occur as the first two points in a password than in any other position. Thus, we include counts of the observed bi-gram positions in our training (each bi-gram would be observed in at least one of the four possible bi-gram positions).

Using the full human-computed data set B (i.e., the click-points collected from all users) for a single image from our lab study,[3] we use the following method:

(I) Create position-aware, normalized bi-grams as follows. Using B: (1) calculate clusters per the algorithm in Section 3.2.1; (2) for each password $i \in B$, normalize each of the 5 click-points to the center of the cluster each belongs to; and (3) split each normalized password into four bi-grams $[(x_1, y_1), (x_2, y_2)]$, $[(x_2, y_2), (x_3, y_3)]$, $[(x_3, y_3), (x_4, y_4)]$, $[(x_4, y_4), (x_5, y_5)]$. Each position-aware, normalized bi-gram has a set of 5 counts: one for total frequency, and four position frequencies (i.e., one frequency count for each of four possible observed bi-gram positions).

(II) Generate all possible distinct passwords based on the bi-grams created in (I), using each bi-gram in a given position if the frequency count at that position is greater than zero. The probability of a generated password is based on the product of the frequencies of each bi-gram at its position within the generated password.

(III) Sort generated passwords by decreasing probability, and use the sorted list in an exhaustive attack to guess the lab study passwords.

4.2 Click-Order Pattern Attacks

Dependencies between click-points could drastically reduce the size of the effective password space; thus, we are also interested in predicting the dependencies between click-points (beyond those considered using the bi-grams of Section 4.1.2) that users may choose in their passwords. Findings that people are better at recalling fewer pieces of visual information [25], and tend to "chunk" information together to aid memorability [10], motivate us to propose and explore Conjecture 3.

Conjecture 3 *Since people find it easier to recall fewer pieces of information, a significant subset of users are likely to choose sets of points that they can "chunk" together by an association between all or most of the points. Such associations include visual similarity (such as in the shape, color, or intensity of the underlying objects), or the overall formation of a simple geometric pattern on the image (such as left to right).*

Consequently, we next explore in Section 5.2 click-based graphical passwords composed of a sequence of points that follow a simple click-order pattern independent of the underlying image (we do not explore other types of visual similarity in the present paper). Later in Section 5.3 we combine click-order patterns with human-seeded attacks.

Click-order pattern dictionaries can consider any click-order pattern that a user might use to relate his/her click points to one another. For example, this might include general sweeping directions from left to right or right to left. We consider a small set of such click-order patterns herein: DIAG (click-based graphical passwords composed of click points in a consistent horizontal *and* vertical direction, which includes straight lines as in Figure 6), HOR (click points in a consistent horizontal direction), VER (click points in a consistent vertical direction), CWCCW (click points in a consistent clockwise or counter-clockwise direction). We prefix each of these subclasses with $S_{clk-ord}$- to denote our particular classification.

We estimate the number of passwords in our lab study database that would be guessed by a click-order pattern dictionary using a program to test the conditions (for each click-order pattern) described further below against each password. We estimate the size of each click-order pattern dictionary in an image-independent manner that is a function of the image dimension and T-region size, using the centers of all T-regions in the entire alphabet space for non-overlapping coverage of the image. For the purpose of this analysis, our base set of T-region centers are aligned such that their T-regions do not overlap, meaning that they begin at pixel coordinates (10, 10), and are in subsequent increments of the T-region size (19 pixels). Only those 5-permutations whose click-points (x_i, y_i), $i = 1, 2, \ldots, 5$ follow one of the following click-order pattern conditions are counted for the corresponding dictionary:

(i) HOR: left-to-right (LR, with $x_i \leq x_{i+1}$) or right-to-left (RL, with $x_i \geq x_{i+1}$).

(ii) VER: top-to-bottom (TB, with $y_i \leq y_{i+1}$) or bottom-to-top (BT, with $y_i \geq y_{i+1}$).

[3] This could be done with any password data set, but in this case, we use those from our lab study.

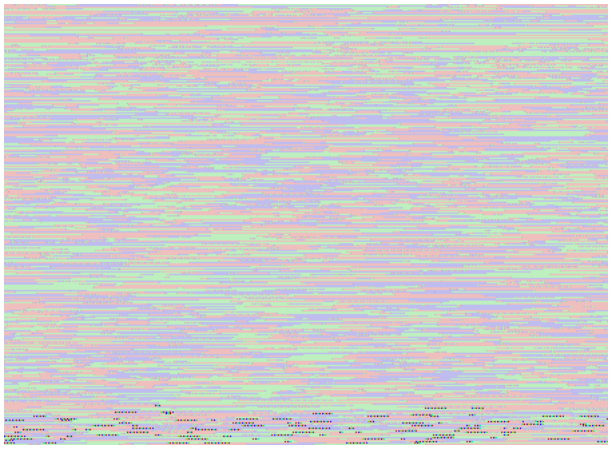


Figure 6: Example $S_{click\text{-}order}$-DIAG password.

(iii) DIAG: LR and (TB or BT), or RL and (TB or BT).

(iv) CWCCW: All sequences of three consecutive points are in the same direction (clockwise or counter-clockwise as computed by Bourke [3]), and the sum of the angles between the three sequences of three consecutive points in the password is no greater than 360 degrees.

In words, HOR is a horizontal sweep from right to left or left to right; VER is a vertical sweep from top to bottom or bottom to top; DIAG is a sweep in both a certain horizontal and vertical direction; and CWCCW is either a clockwise or counter-clockwise, non-overlapping sweep. However, in each of the listed conditions, equality takes the error tolerance t into account; for example, if x_1 to x_4 all follow a left to right click-order pattern, and then $x_5 \geq x_4 - t$, the entire password will be considered to have a HOR click-order pattern.

4.3 Cross-Validation Analysis

Here we describe the 10-fold cross-validation methodology for our human-seeded attack that is based on a first-order Markov model. Cross-validation is performed on a single data set that is partitioned into k *folds*, $k - 1$ of them used for training and the remaining one for testing. Each of the k folds take turns being used as the testing fold, and the results are averaged. Cross-validation (in particular where k = 10) is the recommended method for evaluating the performance of a classifier [24]. This method is similar to the random sub-sampling method used by Davis et al. [11] on recognition based graphical passwords, but has the advantage that each datum is tested precisely once in the k rounds.

We perform a random shuffling of our field study password database, then partition the shuffled database into 10 folds of approximately equal size. One fold is kept for testing, the other 9 are used to train a first-order Markov model as in Section 4.1.2. The fold offset location (controlling the partitioning) is randomly chosen in each of 30 rounds to ensure the fold location does not affect our results.

This methodology may result in a higher guessing success rate than when we use the lab study data set for a variety of reasons. It is conceivable that the field study data is more representative of passwords actually chosen in the long-term, or that the training fold is similar to the testing folds because they are from the same population. If its success is related to the former, the cross-validation analysis models an attacker that is able to obtain a real cleartext password database for training. Although it is considerably more difficult for an attacker to obtain a real cleartext password database, it is possible (e.g., by setting up a web service that uses the system and background image under attack). In either case, the efficacy of such an attack should be examined as an estimate of how well an attacker with an ideal data set could perform.

After randomly shuffling the database, we perform the following method 30 times, for a given image: (i) select a random offset into our field study password database, and divide (starting from the offset) into 10 approximately equal folds; (ii) for each of the 10 folds, keep one out as a test set. Then (a) use the

remaining 9 folds to create position-aware, normalized bi-grams; (b) from the bi-grams, generate passwords as described in Section 4.1.2, sorted by decreasing probability; and (c) use the sorted list to guess passwords among the test set.

5 Results

We present the results of applying each of the types of dictionary attack described in Section 4 to our [illegible] study database. The limitations discussed in Section 3.3 should be taken into consideration when interpreting these results.

5.1 Human-Seeded Attack Results

For reference, Table 3 summarizes the terminology and symbols used in this paper.

Dictionary Name	Description
P_{clstr}	An unordered dictionary containing all possible 5-permutations derivable from all clusters computed from a data set of click-points (as computed in Section 3.2.1).
P^{u}_{clstr}	An unordered dictionary containing all possible 5-permutations derivable from all clusters computed from a data set of click-points (for a given data set with u users).
P_{uni}	All possible 5-permutations of clusters we expect to [illegible] after observing 32 users (the smallest number of lab study users we have for any image), assuming click-points are selected independently and uniformly at random from an alphabet of size 207 (half of the full alphabet size).
P_{raw}	An unordered dictionary containing all possible 5-permutations from a data set of raw click-points (not processed into clusters).
P^{u}_{raw}	An unordered dictionary containing all possible 5-permutations from a data set of raw click-points (for a given data set with u users).
$S_{clk-ord}$	Pre[illegible]used for one of our particular implementations of an (unordered) click-order pattern dictionary.
S_{hs-ind}	An ordered P_{clstr} dictionary, using our data sets to calculate clusters and the probability (and thus order) of the resulting dictionary entries.
S_{hs-dep}	An ordered dictionary, generated using a [illegible]st-order Markov model of all clusters from our data sets of click-based graphical passwords.

Table 3: Summary of various dictionary sets used in this paper.

5.1.1 Results for Independent Cluster Probabilities

Our results in Table 4 are for human-seeded attacks on the [illegible] study database, using the lab study data as our human-computed data set. We use two di erent types of dictionary: P_{clstr}, an ordered dictionary that uses the independent cluster probabilities of Section 4.1.1, and P_{raw}, an unordered dictionary that uses the raw (unprocessed) click-points chosen by the lab users. Note that although the clustering algorithm reduces the size of the dictionary, it also reduces its e cacy. The full P_{clstr} dictionaries (line 2 in Table 4) nonetheless still eventually [illegible] a large number of passwords (20-36%, which is more than half to two-thirds of the number of passwords eventually found by P_{raw}), while reducing the number of entries in the dictionaries by a factor of $2^{3.3}$ to 2^{6}. Table 4 also shows the e ect of reducing the number of people to generate a human-computed data set: as expected, it also reduces the e cacy of the dictionary generated, but 5 click-points each from as few as 15 di erent people can generate enough information to [illegible] 11-23% of passwords (on average), with dictionaries containing $2^{26.4}$ to $2^{28.8}$ entries.

The most striking result shown is that initial password choices harvested from 15 users, in a setting where long term recall is not required, allowed us to [illegible] (on average) 23% of the [illegible] study passwords for *pool* (see

Set	*cars* (u = 33)				*pool* (u = 35)			
	bit-size	number of passwords guessed out of 109			bit-size	number of passwords guessed out of 114		
		avg	min	max		avg	min	max
P^{u}_{raw}	36.7	37(34%)	‰	‰	37.1	59(52%)	‰	‰
P^{u}_{clstr}	33.4	22(20%)	‰	‰	31.1	41(36%)	‰	‰
P^{25}_{raw}	34.7	24(22%)	9(8%)	35(32%)	34.7	42(37%)	29(25%)	56(49%)
P^{25}_{clstr}	31.9	21(19%)	7(6%)	27(25%)	29.2	34(29%)	19(17%)	47(41%)
P^{20}_{raw}	33.1	22(20%)	8(7%)	32(29%)	33.1	35(31%)	24(21%)	55(48%)
P^{20}_{clstr}	30.6	17(16%)	8(7%)	30(28%)	28.2	28(25%)	18(16%)	43(38%)
P^{15}_{raw}	30.9	14(13%)	4(4%)	25(23%)	30.9	30(27%)	20(18%)	45(39%)
P^{15}_{clstr}	28.8	12(11%)	4(4%)	24(22%)	26.4	26(23%)	14(12%)	43(38%)

Table 4: Dictionary attacks using di erent sets. All percentages in the table (after the first two rows) are the result of 10 randomly selected subsets of u = 15, 20, 25 lab study user passwords. Bitsize of x implies 2^x dictionary entries. For rows 1 and 2, note that u = 33 and 35. See text for descriptions of P_{clstr} and P_{raw}. P^{u}_{clstr} is what we refer to as S_{hs-ind}. The first two rows use all data from the short-term study to seed a single dictionary, and as such, there are no average, max, or min values to report.

P^{15}_{clstr}), and with a smaller dictionary. As we expected, *cars* was not as easily attacked as *pool* (guessing on average 11% for P^{15}_{clstr}); more user passwords are required to seed a dictionary that achieves similar success rates (see P^{25}_{clstr}).

We can place an ordering on the S_{hs-ind} (i.e., P^{u}_{clstr}) dictionary such that the passwords are ordered from most to least probable (as defined by the probabilities, from the human-seeded data set, of each cluster in the password). Using this ordering, we next examine the cumulative distribution function (CDF) of P^{u}_{clstr} for each image, as shown in Figure 7. This provides a much more e cient attack than guessing P^{u}_{raw} or P^{u}_{clstr} in no particular order, supported by the information in Table 4.

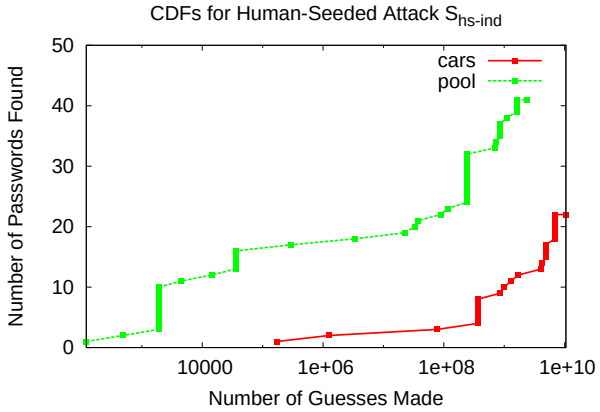


Figure 7: CDF of S_{hs-ind} (i.e., an ordered P^{u}_{clstr}) for *pool* and *cars*. Passwords in the lab study database: 109 (*cars*), 114 (*pool*).

Figure 7 illustrates how much more e ective the ordered S_{hs-ind} dictionary is for *pool* than for *cars*: about 10% of passwords are found in the first 10 000 guesses, and 5% are found within the first 2 000. In contrast, the S_{hs-ind} dictionary for *cars* found 10% of passwords only after over the first 10^9 dictionary entries, and 5% after over 4×10^8 guesses. This is likely due to the low amount of clustering observed in the data collected in the lab study on *cars*, leading to most clusters having the same probability, producing less advantage from ordering.

In Figure 7 (and in later CDF figures), it appears that some guesses match a large number of passwords.

In this attack, it is not meaningful to give a particular ordering of a combination of five click-points a higher priority over another. Thus, we report the number of successful guesses for a combination of points after having guessed all 120 permutations, meaning that when a guess appears to be particularly popular, it indicates the combination of points is popular, not necessarily a single permutation. The graphs and tables are generated such that they report the results for a combination after all permutations have been guessed.

5.1.2 Results for First-Order Markov Model

Our results for the method of Section 4.1.2, presented as a CDF in Figure 8, show that using our method based on a first-order Markov model (as opposed to independent cluster probabilities) guesses more passwords earlier (after 100 guesses, 10% for *pool* and 4% for *cars*), but once the dictionary has been exhausted, it guesses fewer than with independent probabilities (a total of 11% for *pool* and 4% for *cars*). This is because the attack generates fewer dictionary entries than when independent probabilities are used, because when a bi-gram is not observed in the human-seeded data set, it is assumed to have probability 0. Thus, not all bi-grams will generate more than one password (e.g., a single password in the human-seeded data set, whose click-points do not belong to any other clusters, will only be part of an equivalent single password in the final S_{hs-dep} dictionary). This suggests a human-seeded dictionary attack strategy of first guessing those passwords generated based on a first-order Markov model, and then the passwords generated with independent cluster probabilities as per Section 5.1.1. We do not however pursue this strategy further in the present paper.

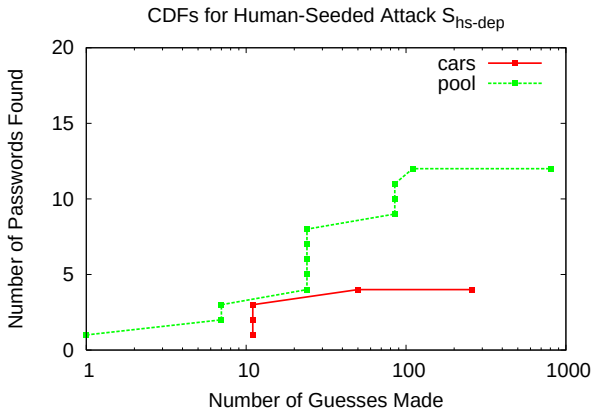


Figure 8: CDF of human-seeded attack based on a first order Markov model (i.e., S_{hs-dep}) for *pool* and *cars*. Passwords in the second study database: 109 (*cars*), 114 (*pool*).

Figure 8 demonstrates that a number of users in the second study chose the exact same click-points, in the exact same order. The 11^{th} guess for *cars* yields 3 passwords, and the 24^{th} guess for *pool* yields 5 passwords. The attack results in Section 5.1.1 are for all orders/permutations for each combination, whereas this attack uses a different (order-based) model.

5.2 Click-Order Pattern Attack Results

Table 5 shows the number of passwords that would be found after applying $S_{clk-ord}$ to attack the second study password database for various click-order patterns per Section 4.2.

The most striking result in Table 5 is that the DIAG click-order pattern, which produces the smallest $S_{clk-ord}$ dictionary, would guess almost 46% of passwords for *cars*, and 26% for *pool*, implying that the dictionary resulting from the $S_{clk-ord}$-DIAG pattern is more effective than that of S_{hs-ind}. These results also give some insight as to which click-order patterns are most popular, and how much the effectiveness of these click-order patterns can differ depending on the image.[4] For example, the *DIAG*, *HOR*, and *VER*

[4] More precisely, while these click-order pattern dictionaries are constant across all images, their effectiveness varies as a result of specific images inducing users to select passwords that fall into patterns.

Dictionary	bitsize of dictionary	*cars* number of passwords	*pool* number of passwords
DIAG	33.0	50/109 (45.9%)	30/114 (26.3%)
HOR	38.0	65/109 (59.6%)	53/114 (46.5%)
VER	38.3	71/109 (65.1%)	39/114 (34.2%)
CWCCW	‰	8/109 (7.3%)	13/114 (11.4%)

Table 5: $S_{clk‰ord}$ results using various click-order patterns for all T-regions. Bitsize of x implies 2^x dictionary entries. ‰This value was too combinatorially costly to compute per our method.

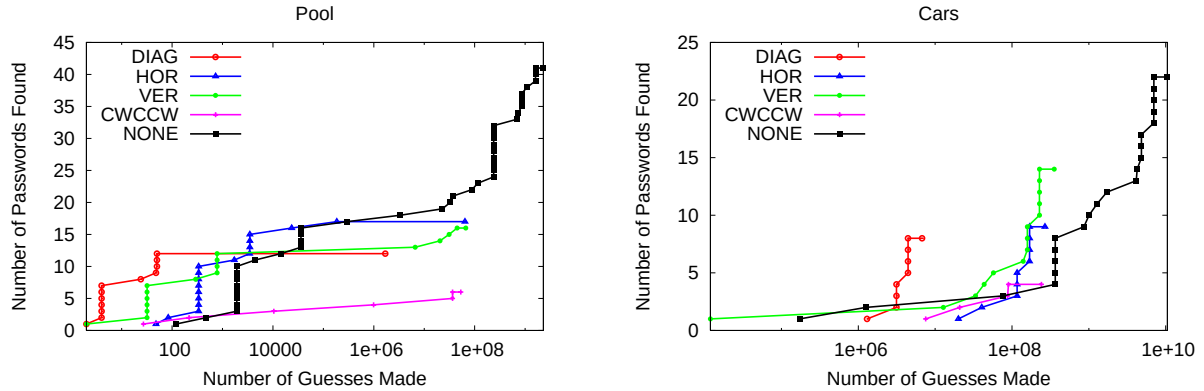
click-order patterns are much more popular in *cars* than *pool*, which is sensible given that *cars* depicts cars parked in straight rows. It is interesting that despite few obvious straight-line structures in *pool* (aside from the pillars on the left hand side), the *DIAG*, *HOR*, and *VER* patterns are all still quite popular. Only 7-11% of passwords followed the *CWCCW* pattern, which is thus the least popular of those examined.

Overall, Table 5 motivates the following attack ordering optimization within $S_{clk-ord}$: *DIAG*, *HOR*, *VER*, *CWCCW*. Of course, since some of these sets have a non-null intersection, the intersection with previous groups should be removed in later groups. For example, the *HOR* and *VER* dictionaries both contain *DIAG* as a subset. Further work on click-order patterns is beyond the scope of this paper, and is provided in another paper [30].

5.3 Combined Human-Seeded and Click-Order Attack Results

Here we brie‰explore combining human-seeded attacks with click-order patterns, by examining the e cacy of intersecting $S_{clk-ord}$ with S_{hs-ind}.

We ordered entries in the intersection of the human-seeded (P^u_{clstr}, i.e., S_{hs-ind}) and click-order pattern dictionaries from most to least probable (as de‰d by the product of the probabilities, based on the lab study data set, of each cluster in the password). The results of applying this ordering of $S_{hs-ind} \cap S_{clk-ord}$ for each image are shown in Figure 9(a) and 9(b).



(a) *pool* (114 ‰d study user passwords) (b) *cars* (109 ‰d study user passwords)

Figure 9: CDFs of $S_{hs‰ind} \cap S_{clk‰ord}$. Each click-order pattern label is abbreviated (e.g, CWCCW denotes $S_{clk‰ord}$-CWCCW). NONE is for $S_{hs‰ind}$ alone, i.e., the human-seeded attack with independent cluster probabilities, and no click-order pattern applied.

Overall, the results show that intersecting S_{hs-ind} with the $S_{clk-ord}$ dictionaries (except *CWCCW*) provides better performance than the S_{hs-ind} dictionary alone (at least initially). This is evident by comparing each line in Figure 9 to the NONE line, which illustrates the CDF of S_{hs-ind} alone. The e ect is more striking for *pool* than for *cars*; for example, in Fig. 9(a) we see the $S_{hs-ind} \cap S_{clk-ord}$-DIAG dictionary ‰ds the ‰st 7 passwords (6% of the total) within 5 guesses. In general for *pool*, Fig. 9(a) shows that all S_{hs-ind}

$\cap S_{clk-ord}$ dictionaries (except $CWCCW$) perform better than S_{hs-ind} alone initially, but by the time they are exhausted, the performance is better for S_{hs-ind} alone. For *cars*, Fig. 9(b) shows that each $S_{hs-ind} \cap S_{clk-ord}$ dictionary performs better than S_{hs-ind} alone (except for the first three correct guesses for which only $S_{clk-ord}$ of VER is superior).

5.4 Cross-Validation Analysis Results

Here we provide cross-validation analysis results, i.e., from 30 rounds of 10-fold cross-validation using only the field study data per Section 4.3. For each of the 10 folds, we average the number of passwords guessed after making 1, 2, 3, 4, 5, 10, 50, 100, 150, 200, 500, 5000, and 10000 guesses. The average and standard deviation of this average for the 30 rounds are plotted in Figure 10.

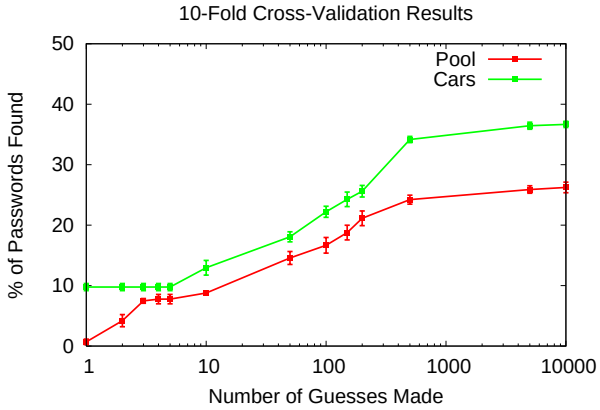


Figure 10: CDF for cross-validation analysis of human-seeded Markov model-based attack using the PassPoints field study database. The % of passwords guessed is the average over 30 rounds; error bars represent standard deviation.

Figure 10 demonstrates our best result; after the first 3 guesses, on average 10% of passwords on *cars* and 7% of passwords on *pool* were correctly guessed. Indeed, for the cars image, 11 participants or 10% of them chose the same password, which is of course quite bad for security. Figure 10 shows that an online attack is possible against PassPoints-style graphical passwords, even on systems that implement conservative account lockout policies.

In general, Figure 10 demonstrates the existance of highly probable passwords other than those captured by a smaller human-computed data set (as in our human-seeded dictionaries), or the simple click-order patterns (as in our click-order pattern dictionaries). This result should be seriously considered, as it indicates an attacker may have even better success given access to better and/or larger human-computed data sets.

5.5 Measuring Differences Between Lab and Field Study Data

Our cross-validation analysis in Section 5.4, which uses the same method as S_{hs-dep} except that it generates the dictionary using the field study data, implies there are differences between the lab and field study data sets. To understand these differences further, we perform random sub-sampling experiments on the lab study to generate P_{clstr} and P_{raw} (recall Section 5.1.1). We also use P_{raw} to minimize information loss (which occurs to some extent when clustering is used). Random sub-sampling is similar to cross-validation, except that the testing set is drawn randomly from the entire set for each trial (as opposed to partitioning the data set).

We used 10 randomly selected sets of 25 users from the lab study to generate both P_{raw} and P_{clstr} against the remaining 8-10 lab study users. For *pool*, the attack appeared to work similarly to when applied to the field study for *pool*, but not for *cars*: the average percentage of guessed lab study passwords for *pool* is 28% using P^{25}_{raw} and 20% using P^{25}_{clstr} (about 9% less than the results when applied to the field study data, as shown in Table 4), but no passwords were guessed for *cars*. These results may indicate differences, for some images, between the passwords selected by the lab study and field study users' real passwords.

6 Related Work

A variety of graphical passwords have been proposed to date (see surveys [40, 28]). Here we focus on click-based graphical password schemes and other work specifically on graphical password guessing attacks.

Click-based graphical password schemes require a user to click on a set of points on one or more presented background images. Blonder [2] presented the first such scheme, whereby the user is asked to choose click-points from a set of predefined tap regions. *V-go*, a system created by PassLogix [32], uses a set of predefined objects in the picture, and asks users to click a sequence of these objects. Both Blonder's and PassLogix's schemes limit what parts of the image a user may click. Jansen et al. [22] propose a variation designed for PDAs, which requires users to click an ordered sequence of visible squares imposed on a background image. The squares are the result of a grid placed over the image, to help the user repeat their click-points in subsequent logins.

PassPoints [45, 47, 46] allows users to click a sequence of five points anywhere on an image while allowing a degree of error tolerance using robust discretization [1] (but see also Chiasson et al. [7]). Various studies have shown that PassPoints has acceptable usability [46, 45, 47, 8]. *VisKey*, a commercial system intended for the Pocket PC, appears similar to PassPoints, but allows the user to choose the number of click-points and to set the error tolerance. Cued Click-Points (CCP) [9] (see also PCCP [5]) is another variation whereby a user clicks on a single point on each of five images; each image (after the first) is dependent on the previous click-point, presumably complicating attacks.

A few other studies have examined the security of click-based graphical passwords. One way that an attacker could predict hot-spots is by using image processing tools to locate areas of interest. Dirik et al. [13] create and evaluate an automated tool for guessing PassPoints passwords. Their method was tested against a database of single-session user choices for two images, albeit one may be too simple for meaningful comparison to other work. With the other image, their method guessed 8% of passwords using an attack dictionary with 2^{32} entries, where their implementation had a 40-bit full password space. In previous work [42], we examine an automated method (based on a variation of Itti et al.'s [21] model of visual attention), guessing an average of 7% (over 17 images) of our lab study passwords using an attack dictionary with 2^{35} entries compared to a full pasword space of 2^{43} passwords. Our predictive dictionaries in the present paper are both more effective (guessing a higher percentage of passwords), and more efficient (smaller in size, requiring fewer guesses). Most recently, Salehi-Abari et al. [37] combine attacks employing focus-of-attention automated image processing tools and click-order patterns (compare to Section 4.2), as well as relaxing constraints on click-order patterns independent of focus-of-attention models, for improved automated attacks.

Another way to examine the security of click-based graphical passwords is to identify click-order patterns that may be popular choices and can be used to define small guessing dictionaries; this method allows attacks that are image independent while not requiring the use of people for human-computation. Click-order pattern attacks were first demonstrated by Thorpe et al. [42] and were also shown to optimize human-seeded attacks. Chiasson et al. [6] further compared the popularity of a set of click-order patterns across three different schemes: PassPoints, CCP, and PCCP. Salehi-Abari et al. [37] defined additional patterns and demonstrated their exploitability with both strict and relaxed pattern definitions; an extension of that paper [30] indicates that some click-order patterns result in a better offline attack than S_{hs-ind} herein, but the accuracy of S_{hs-dep} herein remains superior. As such, human-seeded and click-order pattern based attacks are complimentary approaches: human seeded attacks offer better guessing accuracy such that they are the primary threat in online environments, whereas click-order pattern attacks yield a higher percentage of passwords in an offline attack.

User choice has been successfully modeled for other types of graphical passwords. A variation of the method of Section 4.1.2 was used by Davis et al. [11] to determine (for certain sex/race groups) which sequences of images users were more likely to select for the Faces and Story recognition-based schemes. They create bi-grams (using a training data set containing 80% of their collected user passwords) as an ordered pair of two images from at least one user password. The assumption in the bi-gram model is that each image is dependent upon the image chosen in the previous panel. They use those bi-grams to regenerate passwords, and created a dictionary ordered by decreasing probability as mainly defined by the bi-gram frequencies. They further created an ordering of the entire password space, such that those passwords without representative bi-grams in the training set of passwords are included in the dictionary. They found that 25% of passwords for Faces could be guessed in 13 guesses, and 25% of passwords for Story could be guessed in

113 guesses.

Based on cognitive studies, van Oorschot and Thorpe [31] model user choice in Draw-A-Secret (DAS) pure-recall graphical passwords [23]. The idea of exploiting the structure between click-points is similar to that of exploiting the small number of DAS strokes [31]. The resulting reduction of the e ective password space is similar, as fewer permutations of the click-points (or strokes in the case of DAS) are possible; for further discussion, see Thorpe [41].

7 Concluding Remarks

We provide the first in-depth empirical evaluation of hot-spots in click-based graphical passwords. All of the 17 images used in our lab study showed hot-spotting, although some much more so than others, and the relative probabilities of these hot-spots varied quite considerably.

We have presented what are to date the most e ective attacks against click-based (or cued-recall) graphical passwords. Our attacks are quite e ective even on an image that, according to an in-lab study, was found to have the least hot-spotting. On our two di erent field study background images, one predictive dictionary found 20-36% of passwords (depending on the image) using a guessing dictionary with a factor of 1000 fewer entries than the full 43-bit password space; another found 26-46% with a guessing dictionary similarly smaller by a factor of 1000 times; and a third found 4-10% in only 100 guesses. Furthermore, combinations of these dictionaries found 10-17% of passwords using dictionaries that are smaller by a factor of 10^6 of (i.e., more than 20 bits smaller than) the full password space, and 6% of passwords within 5 guesses on one of the two images. Additionally, we found that e ective human-seeded dictionaries can be generated using data from as few as 15 people.

This work introduces and demonstrates the first application of human-computation to create human-seeded attacks. We conject ure that such human-seeded attacks are generalizable to other non-PassPoints styles of graphical password (e.g., recognition-based including Passfaces [35] and Story [11]). The primary di erence would be the type of data collected in the human-computation phase; for recognition-based graphical passwords, an attacker might collect which images people more commonly select from a set of presented images, rather than what parts of an image people more commonly select.

The use of human-computation, though it requires more (non-computer based) work on the attacker's part than purely automated methods, represents a viable attack strategy. Our human-seeded attacks are based on a human-computed data set collected in a lab study that did not have a long-term component, and thus could be collected quickly. Such data could be obtained by many means (e.g., friends, paying a small number of people, as a side e ect to playing games [44], or restricting access to a popular website until the computation task is complete).

Interesting results emerged when we combined our human-seeded attacks with click-order patterns that exploit dependencies between the hot-spots (i.e., using either the DIAG click-order pattern or the first-order Markov model). The first-order Markov model-based dictionary in general provides a better starting point for an attack, and the DIAG click-order pattern combined with the human-seeded attack (with independent cluster probabilities) would provide the best continuation of an attack once the dictionary entries based on the first-order Markov model have been exhausted.

Our results for click-order pattern attacks indicate that some of the click-order patterns herein are popular and can be used to define a small dictionary. We further demonstrate how they can be used as an optimization for human-seeded dictionaries. As discussed under Related Work, additional results on patterns can be found in other papers [42, 37, 6, 30].

Our dictionary based on the first-order Markov model found an average of 7-10% of passwords within 3 guesses when trained and tested using 10-fold cross-validation on the field study database. Although this attack is generated using the same password database (of real passwords in use for a period of time), it presents an estimate of how well an attacker could perform in a guessing attack when armed with more ideal training data.

Our results suggest that even the better background images have exploitable hot-spots in practice and are vulnerable to the attacks we present herein. Due to one attack herein finding 7-10% of passwords in 3 guesses, it is di cult to recommend the use of PassPoints-style graphical passwords with parameters as implemented and explored in this and previous papers (e.g., [47]). For these parameters, which yield a full

password space of 2^{43} elements, our view is that such systems appear problematic in almost any environment. For other parameter choices, which of course also necessitate re-examining usability, we have not explored the e ectiveness of our attacks nor how the weaknesses discussed herein manifest.

Acknowledgments

We thank Sonia Chiasson and Robert Biddle for their cooperative e ort with us in running the user studies; Prosenjit Bose, Louis D. Nel, Weixuan Li, and their Fall 2006 classes at Carleton University for participating in our pilot study; the anonymous referees for helping improve this work; and Fabian Monrose for his comments and guidence on earlier versions of this work [42, 41]. The first author acknowledges NSERC for funding an NSERC Discovery Grant and his Canada Research Chair in Network and Software Security. The second author acknowledges NSERC for funding a Canada Graduate Scholarship.

References

[1] J.C. Birget, D. Hong, and N. Memon. Robust Discretization, with an Application to Graphical Passwords. *IEEE Transactions on Information Forensics and Security*, 1:395–409, 2006.

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

[3] Paul Bourke. Determining Whether or Not a Polygon (2D) Has its Vertices Ordered Clockwise or Counterclockwise, 1998. `http://local.wasp.uwa.edu.au/~pbourke/geometry/clockwise/index.html`.

[4] Ian Britton (Reproduced by Permission of). Image Ref: 21-35-3. `http://www.freefoto.com`.

[5] S. Chiasson, A. Forget, R. Biddle, and P.C. van Oorschot. Influencing Users Towards Better Passwords: Persuasive Cued Click-Points. In *Proceedings of HCI*, 2008.

[6] S. Chiasson, A. Forget, R. Biddle, and P.C. van Oorschot. User Interface Design A ects Security: Patterns in Click-Based Graphical Passwords. *International Journal of Information Security*, 8(6):387–398, 2009.

[7] S. Chiasson, J. Srinivasan, P.C. van Oorschot, and R. Biddle. Centered Discretization with Application to Graphical Passwords. In *Proceedings of the First USENIX Workshop on Usability, Psychology, and Security (UPSEC)*, 2008.

[8] S. Chiasson, P.C. van Oorschot, and R. Biddle. A Second Look at the Usability of Click-Based Graphical Passwords. In *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS)*, 2007.

[9] S. Chiasson, P.C. van Oorschot, and R. Biddle. Graphical Password Authentication Using Cued Click Points. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, 2007.

[10] N. Cowan. The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity. *Behavioral and Brain Sciences*, 24:87–185, 2000.

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

[12] J.L. Devore. *Probability and Statistics for Engineering and the Sciences*. Brooks/Cole Publishing Company, Pacific Grove, CA, USA, 4th edition, 1995.

[13] A. Dirik, N. Memon, and J.-C. Birget. Modeling User Choice in the PassPoints Graphical Password Scheme. In *3rd Symposium on Usable Privacy and Security (SOUPS)*, 2007.

[14] P.F. Felzenszwalb and D.P. Huttenlocher. E cient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. Code available from: `http://people.cs.uchicago.edu/~pff/segment/`.

[15] FreeImages.com (Photograph Courtesy of). Image ID: wm_asian_places_047. http://www.freeimages.com.

[16] FreeImages.com (Photograph Courtesy of). Image ID: wm_recreation_005. http://www.freeimages.com.

[17] Freeimages.co.uk (Photograph Courtesy of). Image ID: paperclips.jpg. http://www.freeimages.co.uk.

[18] Freeimages.co.uk (Photograph Courtesy of). Image ID: pcb04090023.jpg. http://www.freeimages.co.uk.

[19] C.G. Harris and M.J. Stephens. A Combined Corner and Edge Detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[20] Thomas Hawk (Photograph Courtesy of). Flickr Photo Download: Where I've Been Lately. http://www.flickr.com/photo_zoom.gne?id=96968793&size=o.

[21] L. Itti, C. Koch, and E. Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[22] W. Jansen, S. Gavrilla, V. Korolev, R. Ayers, and Swanstrom R. Picture password: A visual login technique for mobile devices. NIST Report - NISTIR7030, 2003.

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

[24] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* , 1995.

[25] S. J. Luck and E. K Vogel. The Capacity of Visual Working Memory for Features and Conjunctions. *Nature*, 390:279–281, 1997.

[26] 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.

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

[28] F. Monrose and M. K. Reiter. Graphical Passwords. In L. Cranor and S. Garfinkel, editors, *Security and Usability*, chapter 9, pages 147–164. O'Reilly, 2005.

[29] A. Narayanan and V. Shmatikov. Fast Dictionary Attacks on Passwords Using Time-Space Tradeo . In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pages 364–372, 2005.

[30] P.C. van Oorschot, A. Salehi-Abari, and J. Thorpe. Purely Automated Attacks on PassPoints-Style Graphical Passwords. *IEEE Transactions on Information Forensics and Security (TIFS)*, (to appear), 2010.

[31] P.C. van Oorschot and J. Thorpe. On Predictive Models and User-Drawn Graphical Passwords. *ACM Transactions on Information and System Security*, 10(4):1–33, November 2007.

[32] Passlogix. http://www.passlogix.com, site accessed Feb. 2, 2007.

[33] M. Peters, B. Laeng, K. Latham, M. Jackson, R. Zaiyouna, and C. Richardson. A Redrawn Vandenberg and Kuse Mental Rotations Test: Di erent Versions and Factors That A ect Performance. *Brain and Cognition*, 28:39–58, 1995.

[34] Photobucket.com (Photograph Courtesy of). Image ID: 11_12_1_web. http://i26.photobucket.com/albums/c136/hamm239/11_12_1_web.jpg.

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

[36] Shannon Riley. What Users Know and What They Actually Do. *Usability News*, 8(1), February 2006. http://psychology.wichita.edu/surl/usabilitynews/81/Passwords.htm.

[37] A. Salehi-Abari, J. Thorpe, and P.C. van Oorschot. On Purely Automated Attacks and Click-Based Graphical Passwords. In *Proceedings of the 24th Annual Computer Security Applications Conference (ACSAC)*, 2008.

[38] SFR IT-Engineering. The Graphical Login Solution For your Pocket PC - visKey. http://www.sfr-software.de/cms/EN/pocketpc/viskey/index.html, site accessed March 18, 2007.

[39] Siracusa (Reproduced by Permission of), John. Clutter: About 130 Windows. http://arstechnica.com/reviews/os/macosx-10.3.ars/5.

[40] Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. Graphical Passwords: A Survey. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC)*, 2005.

[41] J. Thorpe. *On the Predictability and Security of User Choice in Passwords*. PhD thesis, Carleton University, January 2008.

[42] J. Thorpe and P.C. van Oorschot. Human-Seeded Attacks and Exploiting Hot Spots in Graphical Passwords. In *Proceedings of the 16th USENIX Security Symposium*, 2007.

[43] J. Thorpe (Personal Photograph by). Bee in Garden, 2006.

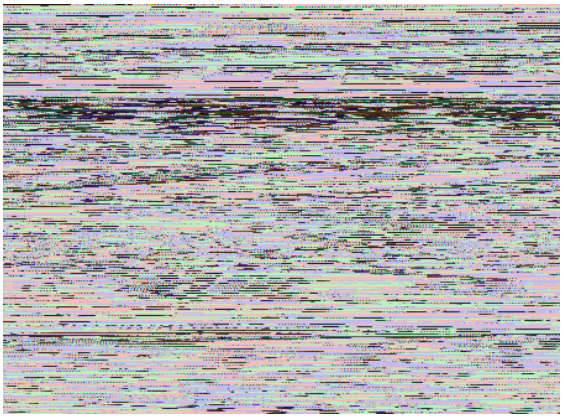
[44] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A Game for Locating Objects in Images. In *Conference on Human Factors in Computing Systems (CHI)*, 2006.

[45] S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskiy, and N. Memon. Authentication Using Graphical Passwords: Basic Results. In *Human-Computer Interaction International (HCII)*, 2005.

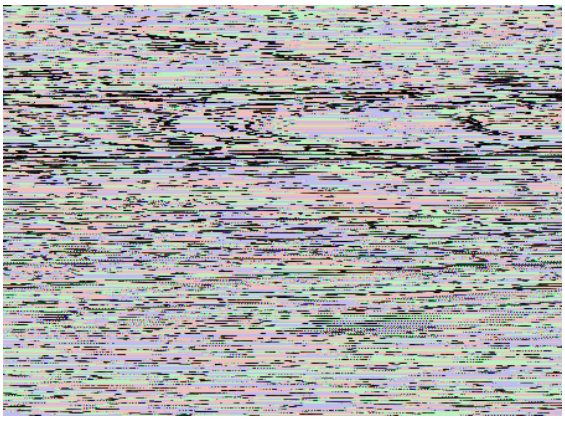
[46] S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskiy, and N. Memon. Authentication Using Graphical Passwords: Effects of Tolerance and Image Choice. In *Proceedings of the 1st Symposium on Usable Privacy and Security (SOUPS)*, 2005.

[47] S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskiy, and N. Memon. PassPoints: Design and Longitudinal Evaluation of a Graphical Password System. *International Journal of Human-Computer Studies (Special Issue on HCI Research in Privacy and Security)*, 63:102–127, 2005.

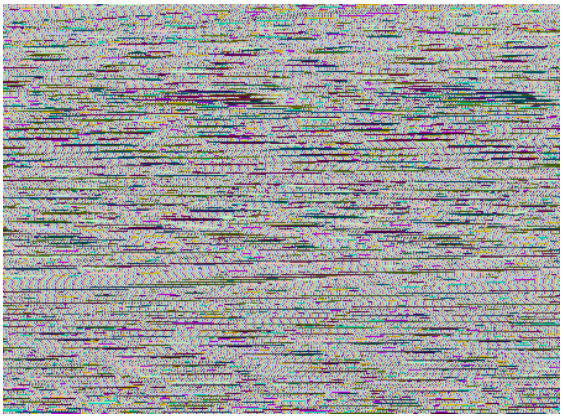
Appendix A - Subset of Images Used in Lab Study



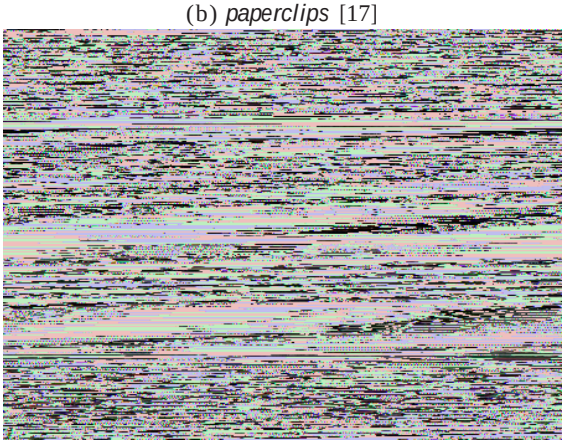
(a) *truck* [15]



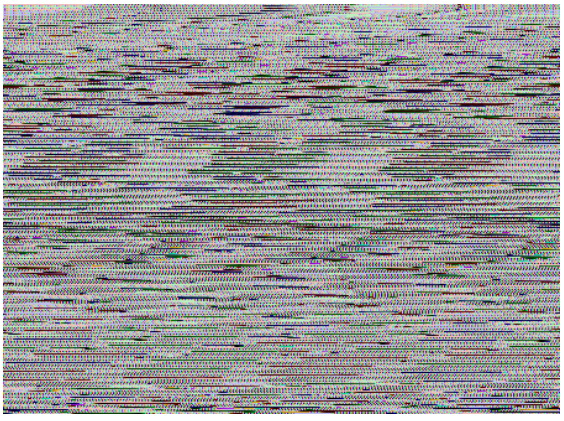
(b) *paperclips* [17]



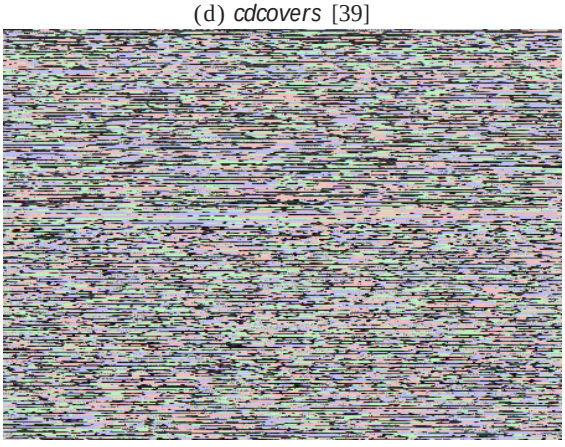
(c) *bee* [43]



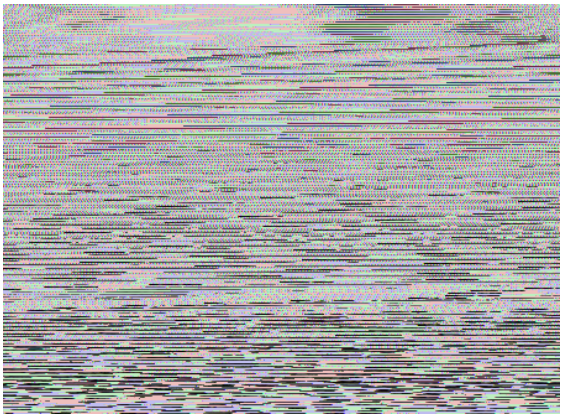
(d) *cdcovers* [39]



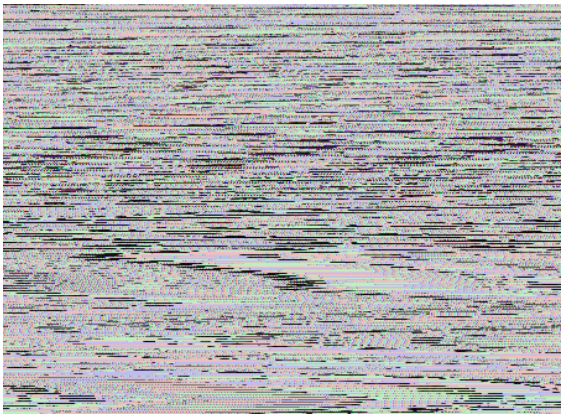
(e) *smarties* [34]



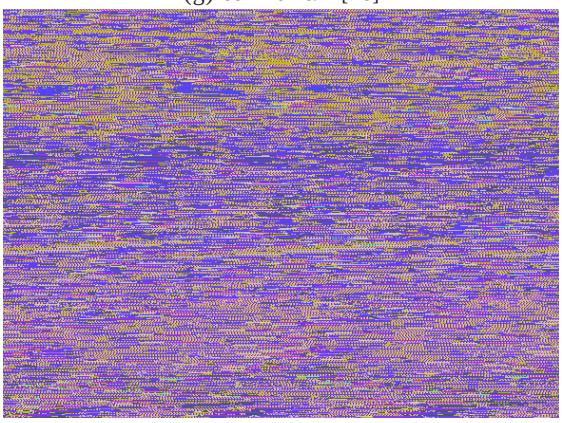
(f) *pcb* [18]



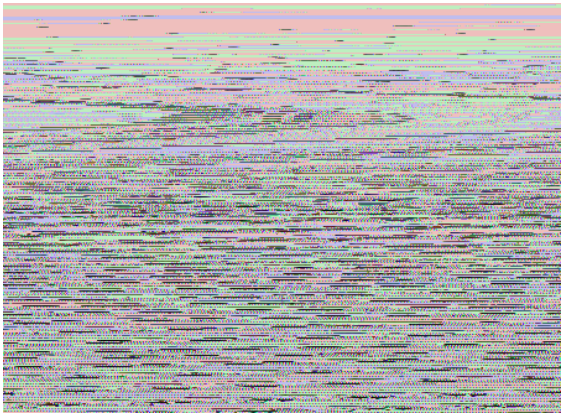
(g) *corinthian* [16]



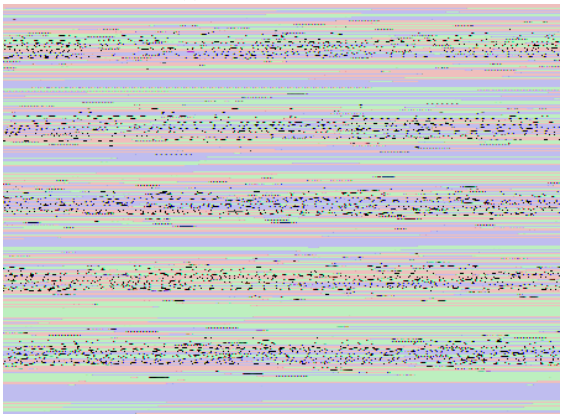
(h) *tea* [46]



(i) *philadelphia* [46]



(j) *mural* [46]



(k) *icons* [20]

Figure 11: Subset of images used in the lab study. See Figure 1 for *cars* and *pool*. The remaining four images used (*citymap-nl*, *citymap-gr*, *faces*, and *toys*) are available from the second author; we were not able to obtain permission to reproduce them herein.