Graphical Password Authentication Using Cued Click Points *

Sonia Chiasson[1,2], P.C. van Oorschot[1], and Robert Biddle[2]

[1] School of Computer Science, Carleton University, Ottawa, Canada
[2] Human-Oriented Technology Lab, Carleton University, Ottawa, Canada
(chiasson,paulv)@scs.carleton.ca, robert_biddle@carleton.ca

Abstract. We propose and examine the usability and security of Cued Click Points (CCP), a cued-recall graphical password technique. Users click on one point per image for a sequence of images. The next image is based on the previous click-point. We present the results of an initial user study which revealed positive results. Performance was very good in terms of speed, accuracy, and number of errors. Users preferred CCP to PassPoints (Wiedenbeck et al., 2005), saying that selecting and remembering only one point per image was easier, and that seeing each image triggered their memory of where the corresponding point was located. We also suggest that CCP provides greater security than PassPoints because the number of images increases the workload for attackers.

Key words: Graphical Passwords, Computer Security, Authentication, Usable Security, User Study

1 Introduction

Various graphical password schemes [14] have been proposed as alternatives to text-based passwords. Research and experience have shown that text-based passwords are fraught with both usability and security problems that make them less than desirable solutions [21]. Psychology studies have revealed that the human brain is better at recognizing and recalling images than text [8]; graphical passwords are intended to capitalize on this human characteristic in hopes that by reducing the memory burden on users, coupled with a larger full password space offered by images, more secure passwords can be produced and users will not resort to unsafe practices in order to cope.

In this paper, we propose a new click-based graphical password scheme called Cued Click Points (CCP). It can be viewed as a combination of PassPoints [19, 20], Passfaces [9], and Story [5]. A password consists of one click-point per image

* version: June 29, 2007
ESORICS 2007, Dresden Germany, September 2007.
J.Biskup and J. Lopez (Eds.): ESORICS 2007, LNCS 4734, pp.359-374, 2007.

for a sequence of images. The next image displayed is based on the previous click-point so users receive immediate implicit feedback as to whether they are on the correct path when logging in. CCP offers both improved usability and security.

We conducted an in-lab user study with 24 participants and a total of 257 trials. Users had high success rates, could quickly create and re-enter their passwords, and were very accurate when entering their click-points. Participants rated the system positively and indicated that they preferred CCP to a PassPoints-style system. They also said that they appreciated the immediate implicit feedback telling them whether their latest click-point was correctly entered.

A preliminary security analysis of this new scheme is also presented. Hotspots (i.e., areas of the image that users are more likely to select) are a concern in click-based passwords [6, 16], so CCP uses a large set of images that will be difficult for attackers to obtain. For our proposed system, hotspot analysis requires proportionally more effort by attackers, as each image must be collected and analyzed individually. CCP appears to allow greater security than PassPoints; the workload for attackers of CCP can be arbitrarily increased by augmenting the number of images in the system. As with most graphical passwords, CCP is not intended for environments where shoulder-surfing is a serious threat.

Section 2 provides background information on related techniques. Cued Click Points (CCP) is described in Section 3, the user study and its results are available in Sections 4 and 5 respectively, and an initial security analysis is given in Section 6. Section 7 provides an interpretation and discussion of the results including possible enhancements, while conclusions and future work appear in Section 8.

2 Background and Related Work

Graphical password schemes can be grouped into three general categories based on the type of cognitive activity required to remember the password: recognition, recall, and cued recall [5, 12]. Recognition is the easiest for human memory whereas pure recall is most difficult since the information must be accessed from memory with no triggers. Cued recall falls somewhere between these two as it offers a cue which should establish context and trigger the stored memory [12].

Among existing graphical passwords, CCP most closely resembles aspects of Passfaces [9], Story [5], and PassPoints [19, 20]. Therefore these graphical password schemes are presented in more detail. Conceptually, CCP is a blend of the three; in terms of implementation, it is most similar to PassPoints. It also avoids the complex user training requirements found in a number of graphical password proposals, such as that of Weinshall [18].

Passfaces [9] is a graphical password scheme based primarily on recognizing human faces. During password creation, users select a number of images from a larger set. To log in, users must identify one of their pre-selected images from amongst several decoys. Users must correctly respond to a number of these challenges for each login. Davis et al. [5] implemented their own version called Faces and conducted a long-term user study. Results showed that users could accu-

rately remember their images but that user-chosen passwords were predictable to the point of being insecure.

Davis et al. [5] proposed an alternative scheme, Story, that used everyday images instead of faces and required that users select their images in the correct order. Users were encouraged to create a story as a memory aid. It fared somewhat worse than Faces for memorability [5], but user choices were much less predictable.

The idea of click-based graphical passwords originated with Blonder [2] who proposed a scheme where a password consisted of a series of clicks on predefined regions of an image. Later, Wiedenbeck et al. [19, 20] proposed PassPoints, wherein passwords could be composed of several (e.g., 5) points anywhere on an image. They also proposed a "robust discretization" scheme [1], with three overlapping grids, allowing for login attempts that were approximately correct to be accepted and converting the entered password into a cryptographic verification key.

Wiedenbeck et al. [19, 20] examined the usability of PassPoints in three separate in-lab user studies to compare text passwords to PassPoints, test whether the choice of image impacted usability, and determine the minimum size of the tolerance square. The overall conclusion was that PassPoints was a usable authentication scheme.

We recently conducted two user studies [3] on a PassPoints-style system. Our initial lab study revisited the original usability claims, explored usability of such passwords on a wider range of images (17 images), and gathered information about users' password choices. Next, we conducted a large-scale field study that examined click-based graphical passwords in practice.

Intuitively, it seems obvious that some areas of an image are more attractive to users as click-points [13]. If this phenomenon is too strong, the likelihood that attackers can guess a password significantly increases. If attackers learn which images are being used, they can select a set of likely hotspots through image processing tools or by observing a small set of users on the target image and then building an attack dictionary based on those points [6, 16]. For further discussion, see Section 6.

3 Cued Click Points

Cued Click Points (CCP) is a proposed alternative to PassPoints. In CCP, users click one point on each of $c = 5$ images rather than on five points on one image. It offers cued-recall and introduces visual cues that instantly alert valid users if they have made a mistake when entering their latest click-point (at which point they can cancel their attempt and retry from the beginning). It also makes attacks based on hotspot analysis more challenging, as we discuss later. As shown in Figure 1, each click results in showing a next-image, in effect leading users down a "path" as they click on their sequence of points. A wrong click leads down an incorrect path, *with an explicit indication of authentication failure only after the final click*. Users can choose their images only to the extent that their click-point

dictates the next image. If they dislike the resulting images, they could create a new password involving different click-points to get different images.

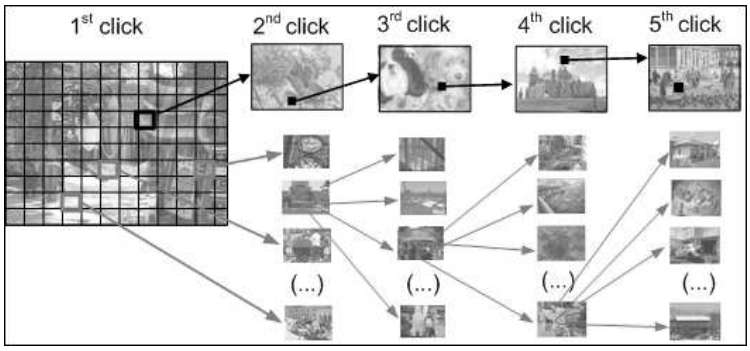

Fig. 1. CCP passwords can be regarded as a choice-dependent path of images

We envision that CCP fits into an authentication model where a user has a client device (which displays the images) to access an online server (which authenticates the user). We assume that the images are stored server-side with client communication through SSL/TLS. For further discussion, see Section 6.

For implementation, CCP initially functions like PassPoints. During password creation, a discretization method (e.g., see [1]) is used to determine a click-point's tolerance square and corresponding grid. For each click-point in a subsequent login attempt, this grid is retrieved and used to determine whether the click-point falls within tolerance of the original point. With CCP, we further need to determine which next-image to display.

Similar to the PassPoints studies, our example system had images of size 451x331 pixels and tolerance squares of 19x19 pixels. If we used robust discretization [1], we would have 3 overlapping candidate grids each containing approximately 400 squares and in the simplest design, 1200 tolerance squares per image (although only 400 are used in a given grid). We use a function $f(username, currentImage, currentToleranceSquare)$ that uniquely maps each tolerance square to a next-image. This suggests a minimum set of 1200 images required at each stage. One argument against using fewer images, and having multiple tolerance squares map to the same next-image, is that this could potentially result in misleading implicit feedback in (albeit rare) situations where users click on an incorrect point yet still see the correct next-image.

Each of the 1200 next-images would have 1200 tolerance squares and thus require 1200 next-images of their own. The number of images would quickly become quite large. So we propose re-using the image set across stages. By re-using images, there is a slight chance that users see duplicate images. During the 5 stages in password creation, the image indices $i_1, ..., i_5$ for the images in the password sequence are each in the range $1 \leq i_j \leq 1200$. When computing the next-image index, if any is a repeat (i.e., the next i_j is equal to i_k for some

$k < j$), then the next-image selection function f is deterministically perturbed to select a distinct image.

A user's initial image is selected by the system based on some user characteristic (as an argument to f above; we used *username*). The sequence is re-generated on-the-fly from the function each time a user enters the password. If a user enters an incorrect click-point, then the sequence of images from that point onwards will be incorrect and thus the login attempt will fail. For an attacker who does not know the correct sequence of images, this cue will not be helpful.

We expect that hotspots [6, 16] will appear as in PassPoints, but since the number of images is significantly increased, analysis will require more effort which increases proportionally with the configurable number of images in the system. For example, if attackers identify thirty likely click-points on the first image, they then need to analyze the thirty corresponding second images (once they determine both the indices of these images and get access to the images themselves), and so on, growing exponentially.

A major usability improvement over PassPoints is the fact that legitimate users get immediate feedback about an error when trying to log in. When they see an incorrect image, they know that the latest click-point was incorrect and can immediately cancel this attempt and try again from the beginning. The visual cue does not explicitly reveal "right" or "wrong" but is evident using knowledge only the legitimate user should possess. As with text passwords, PassPoints can only safely provide feedback at the end and cannot reveal the cause of error. Providing explicit feedback in PassPoints before the final click-point could allow PassPoints attackers to mount an online attack to prune potential password subspaces, whereas CCP's visual cues should not help attackers in this way. Another usability improvement is that being cued to recall one point on each of five images appears easier than remembering an ordered sequence of five points on one image.

4 User Study

We conducted an in-lab user study of CCP with 24 participants. The methodology was reviewed and approved by the university's research ethics committee. The participants (12 females and 12 males) were university students with diverse backgrounds. None were specifically studying computer security but all were regular web users. They ranged in age from 17 to 26 years. Two had participated in our previous in-lab study [3], testing a PassPoints-style system.

All participants completed an individual one-hour session in our usability lab. They first read and signed the consent form and were given an introduction to the tasks they would be completing during the session. This introduction included showing them an example image with superimposed squares, demonstrating how accurate they needed to be when re-entering their points. The tolerance squares used in this study were 19x19 pixels. We also explained that the next image in the sequence depended on where they clicked on the current image. They

were told that if they suddenly saw an image they did not recognize during the Confirm or Login phases, then they were likely on the wrong path. Participants completed two practice trials followed by at most 12 real trials. In total, 257 real CCP trials were completed.

A trial consisted of the following steps. The phases indicated in steps 1, 2, and 5 represent the password phases used in later analysis.

1. Create phase: Create a password by clicking on one point in each of five system-selected images presented in sequence.

2. Confirm phase: Confirm this password by re-entering it correctly. Users incorrectly confirming their password could retry the confirmation or return to Step 1. A new password started with the same initial image, but generally included different images thereafter, depending on the click-points.

3. Two questions: Answer two 10-point Likert-scale questions on the computer about their current password's ease of creation and predicted memorability. Likert-scale questions ask respondents to indicate their level of agreement with the given statement on a scale ranging from strongly agree to strongly disagree.

4. MRT: Complete a Mental Rotations Test (MRT) puzzle [10]. This paper-based task was used to distract users for a minimum of 30 seconds by giving them a visual task to complete in order to clear their working memory.

5. Login phase: Log in with their current password. If users noticed an error during login, they could cancel their login attempt and try again. Alternatively, if they did not know their password, they could create a new password, effectively returning to Step 1 of the trial with the same initial image as a starting point. If users felt too frustrated with the particular images to try again, they could skip this trial and move on to the next trial.

Participants completed as many trials as they wished in the one-hour session, to a maximum of 14 (2 practice + 12 real trials). At the midpoint, participants took a break and completed a demographics questionnaire. The last ten minutes of the session were devoted to completing a Likert-scale and open-ended questionnaire about their perceptions and opinions of these graphical passwords. For each participant, data from the two practice trials were discarded, so all results reported in this paper are based on data from the subsequent trials.

When time remained in the one-hour session, participants were given one further task: to complete a trial with our PassPoints-style system, where they selected five points on one image. The experimenter was careful to identify the second system as "the other system we are looking at" rather than the "old" system, to not bias participants into thinking that they should rate CCP more favourably. Users were then asked which version they preferred.

A prototype application was developed in J#. A set of 330 images was compiled from personal collections as well as from websites providing free-for-use

images. The prototype system did not hash the passwords or use a discretization method as would a real system, but simply stored the exact pixel coordinates so that the users' choice of click-points and their accuracy on re-entry could be examined. The system also implemented an improvised image selection process to reduce the size of the required image set since with several unique trials per participant, we would have needed several thousand images to implement the proposed scheme. The first time a user clicked on a point, a new image was associated with that point. If a user clicked within the tolerance region of that point again, either for re-entering or for resetting a password, the same image was shown. Once associated with a click-point, an image was not re-used for any other click-point during the entire session. Only areas where the user clicked had images associated with them, therefore reducing the total number of images required while still behaving in a manner consistent with the actual proposed scheme from the user's perspective.

Consistent with published PassPoints results, the images were 451x331 pixels in size and were displayed on a 19-inch screen with its resolution set to 1024x768 pixels. We used tolerance squares of size 19x19 (PassPoints studies report a 20x20 pixel tolerance).

Three methods were used to collect data in this study: system logs, questionnaires, and observations. The system recorded the exact pixel coordinates of each click-point on every image visited by participants for every Create, Confirm, or Login attempt, along with the time of each event.

A post-test questionnaire was used to gather information about users' perceptions and opinions. A second questionnaire was used to collect demographic data to help frame the results of the study. Users were also asked two online questions immediately after successfully confirming their password to get an immediate reaction of how easy it was to create the password and how difficult they expect it would be to remember their password in a week.

Finally, an observer sat with each participant throughout the session, noting any difficulties or unexpected behaviour as well as comments made by the participants. While participants were not instructed to use a talk-aloud protocol, they were not discouraged from speaking if they had comments as they worked. Because comments may have slowed down completion times, any questions by the observer were asked between trials where they would not affect the timings.

5 Collected Results

Restarts. Participants used the reset button as soon as they saw an incorrect image and realized they were on the wrong path. This effectively cancelled the current attempt and returned them to the first image where they could start entering their password again. A few times, participants restarted even when they saw the correct image because they had forgotten the image. Failed login attempts (where users pressed the login button and were explicitly told that their password was incorrect) occurred only when users clicked on the wrong point for the last image since they did not receive any implicit feedback for that click-

point. Because these were so few, failed login attempts are included in the restart counts. Participants said that confirming the password helped them to remember it. Once they had successfully confirmed the password, logging on even after the distraction task was relatively easy. This fact is reflected in Table 1 which shows that the vast majority of restarts occurred during the Confirm phase.

Table 1. Total number of restarts, success rates, and completion times per phase

	Create	Confirm	Login
Total Number of Restarts	7	101	14
Success Rates	251/257 (98%)	213/257 (83%)	246/257 (96%)
Mean Time (SD)(in seconds)	24.7 (16.4)	10.9 (13.1)	7.4 (5.5)
Median Time (in seconds)	19.1	7.4	6.0

Four participants completed all their trials without any restarts, i.e., they made no errors during the entire session. In total, 201 of 257 trials (79%) were completed without restarts in any phase. The success rates were high for all phases, as shown in Table 1. Success rates were calculated as the number of trials completed without errors or restarts over the total number of trials.

Accuracy. Participants were extremely accurate in re-entering their passwords. As a measure of accuracy, all individual click-points in the Confirm and Login phases were evaluated. This totalled 1569 click-points for the Confirm phase and 1325 click-points for the Login phase. For each point, the accuracy was computed as the maximum of $|x_{original} - x_{current}|$ and $|y_{original} - y_{current}|$. All click-points were considered in the analysis, even those that were unsuccessful. A few times, participants reached an incorrect image and still proceeded to click on a point. These were included in the 51+ category since the point was obviously forgotten. As indicated in Figure 2, 86% of points were within 4 pixels of the original click-point for the Confirm phase compared to 92% for the Login phase. Falling within 4 pixels of the original point means that these click-points would have been accepted within a tolerance square of 9x9 pixels.

Times for password entry. As expected, participants took longest to create their password and then were progressively quicker in entering it during the Confirm and Login phases. The reported times encompass from the first click in a phase until the last click, including any restarts. The mean and median times reflected in Table 1 are slightly elevated because some participants paused to comment as they were entering their password, which slowed their performance. Despite this fact, login times are well below 10 seconds and the total time to create and confirm a CCP password is approximately half a minute, which we expect would be quite acceptable in many applications or environments.

Questionnaires. Participants completed two sets of Likert-scale questions. Ten-point Likert scales were used, where 1 indicted strong disagreement and

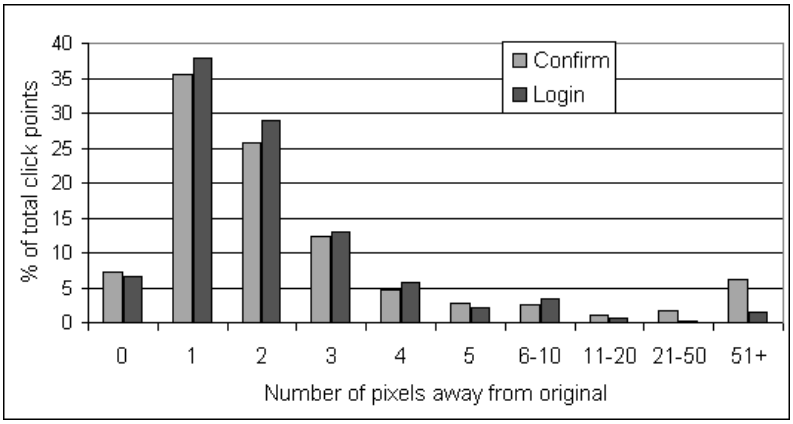

Fig. 2. Accuracy for each phase

10 equalled strong agreement with the given statement. First they answered two online questions immediately after successfully confirming each of their passwords. They gave both "ease of creating a password" and "ease of remembering their password in a week" median scores of 5 (means of 4.6). Secondly, they completed a post-test questionnaire at the end of the one-hour session. In Table 2, we report on a subset of the questions, corresponding to the questions reported in our study of a PassPoints-style system [3]. Some of the questions were inverted to avoid bias; as a result the scores for the statements marked with (*) were reversed before calculating the means and medians. A higher score always indicates a more positive result for CCP.

Table 2. Questionnaire responses. Scores are out of 10. * indicates scale was reversed

Questions	Mean	Median
I could easily create a graphical password	8.2	8.5
* Someone who knows me would be better at guessing my graphical password than a stranger	4.4	5
Logging on using a graphical password was easy	7.5	7
Graphical passwords are easy to remember	7.2	6.5
* I prefer text passwords to graphical passwords	3.6	5
* Text passwords are more secure than graphical passwords	4.4	5
I think that other people would choose different points than me for a graphical password	8.0	8
With practice, I could quickly enter my graphical password	8.3	9

All post-test questionnaire questions had median values of neutral or higher, with several questions showing high levels of satisfaction. Participants showed some concern over the perceived security of graphical passwords and indicated a preference for text-based passwords. Looking at the two online questions shows that users initially felt that it was somewhat difficult to select passwords. Interestingly, by the time they responded to the post-test questionnaire, they felt

much better about password creation. They also showed some hesitation about whether they would be able to remember their password in a week. This may have been exacerbated by the fact that they were creating multiple passwords in a row and did not feel that they would be able to remember all of them. Additionally, in [3] we show that long-term memorability of click-based passwords did not appear to be an issue.

Preference between CCP and PassPoints. When time permitted, participants were introduced to a PassPoints-style system and asked which they preferred. Ten participants attempted a trial with the PassPoints-style system. An additional two people had participated in our earlier study [3]. Of these 12 people, 9 strongly preferred CCP, one preferred PassPoints, and two felt that PassPoints was easier but that CCP was more secure.

User choice. Users were told in the preamble to the session to pretend that their passwords were protecting bank information and as such they should choose points that were memorable to them but difficult for others to guess. Users took these instructions seriously and many commented on how they were avoiding certain areas because they would be too easy to guess or because they felt that others would select the same points.

Users developed strategies for selecting their points. Some tried to pick geometric patterns that applied across images such as selecting items in a row along the bottom of the images, but most talked about picking things that have meaning to them such as their initial from a sign or a familiar toy. One participant made up elaborate stories about each of the click-points. Users indicated that they preferred to click on things that were small and "clickable", such as centers of letters or circles.

Participants felt strongly about the suitability of some images, with strongest reactions to images they disliked. They preferred images that were not too cluttered, that contained a variety of distinct items, that had small well-defined areas, and that featured contrasting colors. The most disliked images were those that were uniform and repetitive, such as a circuit board or field of flowers, that were highly cluttered, or that had few items with well-defined borders.

6 Preliminary Security Analysis

Any proposed authentication scheme needs to be evaluated in terms of possible threats. We begin by clarifying our target scenario for CCP and the particular assumptions made about the system.

We recommend that CCP be implemented and deployed in systems where offline attacks are not possible, and where any attack will be made against an online system that can limit the number of guesses made per account in a given time period (this limit should include restarts as well). This follows related comments by Davis et al. [5] regarding Faces and Story, even though we expect the

security of CCP to be substantially stronger than those schemes. We further assume that all communication between the client and server will be made securely through SSL, maintaining secrecy of selected click-points and corresponding images, therefore avoiding simple attacks based on network sniffing.

We suggest that the image mappings (the mapping of tolerance squares to next-images based on f) be done on a per-user basis as a function of the username, as a form of salting to complicate the construction of general attack dictionaries. We also suggest that the image set across all users is a superset containing a very large number of images and that users are assigned a subset of these images for their image-maps.

General attacks against such a system, where attackers try to break into any account [11], are slowed due to the precautions mentioned above. We assume that the function f would eventually become known to attackers. Hotspot analysis might be used to increase the efficiency of an attack dictionary but images would need to be collected and such a dictionary would need to be generated on a per-user basis. Online attacks against specific users are more worrisome and require further examination. Even for online systems where the account is locked after t failed login attempts, non-trivial security is still necessary to guard against system-wide attacks over W accounts since an attacker gets $t * W$ guesses per time window [11]. Several scenarios are discussed below.

Shoulder-surfing and other information capture from users. Most graphical passwords are vulnerable to shoulder-surfing attacks [15]. With today's small cameras and camera phones, it is easy to video-capture a user's screen or keyboard as they are logging in. CCP is also susceptible to such attacks and indeed in its present form the change in images may be easier to see from further away than mouse pointer movements in PassPoints. With knowledge of which images to look for in systems allowing sufficient numbers of online guesses, attackers could try a brute-force attack of clicking on points until the correct next image appears and use this in a divide-and-conquer password recovery.

If the username, the image sequence, and the click-points are observed through shoulder-surfing then an attacker has all of the information needed to break in to the account, as is the case with PassPoints and most other password systems. Having a compromised computer is also a threat because malware may capture the login information and relay that information elsewhere. Whereas a keylogger suffices for text passwords, for graphical passwords software is needed to capture the images and the cursor positions.

When only some of the information is known, it can be used to narrow the search for a correct guess. With PassPoints, knowing the username is enough to retrieve the user's sole image. Hotspot analysis [6, 16] can then be conducted on that particular image. With CCP, the username and an online guess reveal only the first image so hotspot analysis of this image gives only limited information to an attacker.

Knowing some images and their position in a user's sequence allows pruning of an attack dictionary. The more images are known, the smaller the attack

dictionary and the easier the attacker's job. Thus CCP is not suitable in environments where shoulder-surfing is a realistic threat, or environments where user images can otherwise be recorded (e.g., by insiders, malicious software on the client machine, etc.).

Hotspots and dictionary attacks. In cases where attackers are not in a position to capture information from the user, they are limited to what they can deduce through image analysis.

Hotspots are specific areas in the image that have a higher probability of being selected by users as part of their passwords. If attackers can accurately predict the hotspots in an image, then a dictionary of passwords containing combinations of these hotspots can be built. Hotspots are known to be problematic for PassPoints [6, 16]; further analysis is needed to determine whether precautions such as carefully selecting images can minimize this threat.

Our example system had 400 tolerance squares per grid for a given image. Because the chosen grid is stored during password creation, the correct grid is always retrieved by the system during login so the fact that there are several grids (and 1200 images) does not come into play. This means that for each image, there is a 1/400 chance of clicking within the correct tolerance square. However, due to hotspots some of these have a much higher probability of being correct than others. Knowing the hotspots would allow an attacker to modify an attack dictionary to test passwords with higher probability first. For example, re-examining the data from our larger PassPoints-style study [3] we found that, as a general result across 17 images used, the 30 largest hotspots on an image cover approximately 50% of user-chosen click-points. Assuming that attackers are first able to extract the necessary images and perform hotspot analysis, there is approximately a 3% ($.5^5$) chance that a password is contained in a dictionary of 2^{25} entries built entirely from hotspots.

A key advantage of CCP over PassPoints is that attackers need to analyze hotspots on a large set of images rather than only one image since they do not know the sequence of images used for a given password. Secondly, using different subsets of images for different users means that an attacker must somehow gather information about the specific subset assigned to the current user.

Further testing is required to gather a larger sample of click-points per image for CCP, but preliminary analysis provides evidence that users are no more likely to select a popular hotspot as their click-point in CCP than with PassPoints. When presented with the same images, users selected similar points in both our CCP and PassPoints-style [3] user studies.

7 Further Discussion

From a usability point of view, CCP appears quite successful. Participants were satisfied, their performance was good in terms of success rates and accuracy, and they felt that using this type of password was getting easier as they progressed through the session. The median time required to create (19.1 seconds) and

confirm a password (7.4 seconds) is acceptable and login times (6.0 seconds) are reasonable as well. Success rates were high, with 96% of logins being successful.

Users appreciated the implicit feedback. As soon as they saw an unfamiliar image, they knew they were on the wrong path and restarted. They liked being able to narrow down exactly which click was erroneous. They also felt that seeing each image triggered the memory of where they had clicked.

Participants were surprisingly accurate in their targeting of click-points. During the Login phase, 92% of click-points fell within a 9x9 pixel square of the original click-points. In agreement with our earlier studies with a PassPoints-style system [3], the accuracy findings for CCP provide further evidence that tolerance squares as small as of 9x9 pixels may be acceptable terms of usability. Our previous studies [3] also showed that varying screen resolutions did not appear to impact performance so we predict that the same will apply to CCP.

We can also compare results of CCP with our own PassPoints-style user studies [3]. When comparing only the lab studies, participants performed similarly well in terms of login accuracy and success rates. The median login click-time for our PassPoints-style system was 7.0 seconds while for CCP it was 6.0 seconds but CCP's time also included "think-time" as each image appeared (as opposed to PassPoints where the majority of "think-time" occurred before the first click and as such is not included in these click-time results). Of those participants who tried both systems, a strong preference for CCP was evident. The most common reasons were because seeing each image triggered their memory of their click-point, there was no need to remember the order of the click-points, and they received implicit feedback about the correctness of their latest click. This comparison is somewhat biased since users had much more practice with one system than the other, but these responses do correspond to what would intuitively be expected.

7.1 Potential Improvements

With any password-based authentication scheme, a primary goal is to maximize the effective password space (i.e., that subset of the full password space actually used in practice) in order to make it more resistant to attack. A few alternatives exist to increase the effective password space for CCP.

Adding more click-points. As with PassPoints, one way to increase the password space is to increase the number of click-points contained in a password. This comes at the cost of increasing the memory burden on users. Although we have no empirical evidence to support this hypothesis, it seems that the negative impact would be less with CCP than with PassPoints since a one-to-one mapping between images and click-points in CCP would appear to be easier for users to manage. Therefore moving to 6 click-points may be a reasonable strategy for CCP. Alternatively it is possible to enforce a minimum number of clicks (images) but allow users to decide for themselves how many clicks their password contains, similar to minimum password lengths for text-based passwords.

In this case, the system would continue to show the next image in the sequence but the user would determine at which point to stop clicking and press the login button. Granted, most users would probably pick the minimum length, but a user concerned about security could build a longer password. If k bits of security are assumed per image used, then for a password using c images, the security would be $c * k$.

Adjusting the image and tolerance sizes. A simple way of enlarging the password space is to use larger images or reduce the tolerance. Both have the effect of adding squares to the grids. Tolerance cannot be reduced past a certain threshold because it becomes impossible for users to enter their passwords. Results of this and our earlier study [3] however indicate that it may be possible to reduce tolerance more than was originally believed [19] (at least on full-sized monitors) since users were very accurate in targeting their click-points. For example, with images of size 451x331 as used in these studies, there are approximately 400 19x19 pixel grid squares, giving $2^{8.6}$ squares per image. If we reduced the tolerance squares to 9x9 pixels, this would increase to $2^{10.9}$ squares per image. With CCP, we can multiply by 5 for the number of images in the password sequence, increasing from 2^{43} to $2^{54.5}$ choices. Enlargement of the image is restricted by the size of the screen used. Increasing the size of the image may also make it more susceptible to shoulder-surfing. Zooming, which has been suggested elsewhere, including by Wiedenbeck et al. [20] for PassPoints, often has usability problems of its own, and thus we hesitate to propose it here.

Using a larger set of images. At minimum, the size of CCP's total image set should match the number of squares in a tolerance grid (i.e., 400 in our example system). This strategy would imply that the set of images in the system is re-used across users and at each stage in the password for each user.

In this case, if users make a mistake during login, there is a small chance that they accidentally see an image belonging somewhere else in their password sequence. They may realize the mistake immediately or subsequently when an unknown next-image appears. The possibility of such collisions can be reduced or eliminated if the number of images is increased to reduce the overlap between password stages. However, depending on implementation details, this could imply that the entire sequence could be deduced from knowing only the last image in a password as discussed below.

As suggested earlier, it is possible to have a larger set of images in the system and to use a subset for each user. Additionally, the subset for each user may also hold extra images so that not every image is re-used at each stage. This can reduce the possibility of collisions during incorrect login. It also increases the amount of work required by attackers to identify images and determine hotspots as this work increases proportionally with the number of images used in the system. If attackers are using a brute-force attack where a dictionary is built from all possible combinations of images and click-points, then this also forces a larger dictionary of size $totalImages * totalGridSquares * totalClicks$.

In comparison, with PassPoints only one image needs to be analyzed per user and this image is accessible by simply knowing the username.

If attackers know the image-mapping function f and the set of images used, then having more images has no effect on the password space beyond requiring more processing time to determine hotspots. However, even if attackers know f, collecting the set of images still poses a challenge because they must either have insider access to the system or they must discover the images one at a time by selecting different click-points during login attempts on the particular account. This can prove time-consuming since the number of unsuccessful login attempts allowed on a particular account can be restricted (e.g., see [17]). When both f and the image set are known, the password space is determined by the number of paths through the image-map tree (generated by f), based on the number of squares in the tolerance grid, not the number of images available. If a dictionary was built containing all paths through the tree, the number of entries would be the same (2^{43} for grids containing 400 squares and 5 clicks) regardless of the number of images used (although the entries would be different).

In cases where attackers know f and the set of images used, as well as one or more images in the password (gathered through shoulder-surfing or malware installed on the client machine), then having a very large set of images for a given user actually reduces the attacker's search space (although the amount of work required for hotspot analysis is still increased). Since not all images will be used within each image-map, attackers can use this information to eliminate branches of the image-map tree that do not contain the known image at the correct stage. At the extreme case where there are no duplicate images, then knowing the last image of a sequence would identify a unique path through the tree and reveal the password. Conversely, when all images are re-used at each stage then no branches can be eliminated and knowing the last image will not result in a unique path.

Another alternative for increasing the number of images available is to use larger images but crop them differently for each user. This would complicate hotspot analysis for attackers because the coordinates of hotspots determined for one account could not be applied directly to other accounts.

8 Conclusions and Future Work

The proposed Cued Click Points scheme shows promise as a usable and memorable authentication mechanism. By taking advantage of users' ability to recognize images and the memory trigger associated with seeing a new image, CCP has advantages over PassPoints in terms of usability. Being cued as each image is shown and having to remember only one click-point per image appears easier than having to remember an ordered series of clicks on one image. In our small comparison group, users strongly preferred CCP.

We believe that CCP offers a more secure alternative to PassPoints. CCP increases the workload for attackers by forcing them to first acquire image sets for each user, and then conduct hotspot analysis on each of these images. Fur-

thermore, the system's flexibility to increase the overall number of images in the system allows us to arbitrarily increase this workload.

Future work should include a thorough assessment of the viability of CCP as an authentication mechanism, including a long term study of how these passwords work in practice and whether longer CCP passwords would be usable. The security of CCP also deserves closer examination, and should address how attackers might exploit the emergence of hotspots.

Acknowledgements. We thank the participants of our lab study and the reviewers for their valuable feedback. The first and third authors are supported in part by the "Legal and Policy Approaches to Identity Theft" project funded by ORNEC. The second author acknowledges NSERC for funding his Discovery Grant and his Canada Research Chair in Network and Software Security.

References

1. Birget, J.C., D. Hong, and N. Memon. Graphical Passwords Based on Robust Discretization. IEEE Trans. Info. Forensics and Security, 1(3), September 2006.
2. Blonder, G.E. Graphical Passwords. United States Patent 5,559,961, 1996.
3. Chiasson, S., R. Biddle, R., and P.C. van Oorschot. A Second Look at the Usability of Click-based Graphical Passwords. ACM SOUPS, 2007.
4. Cranor, L.F., S. Garfinkel. *Security and Usability.* O'Reilly Media, 2005.
5. Davis, D., F. Monrose, and M.K. Reiter. On User Choice in Graphical Password Schemes. 13th USENIX Security Symposium, 2004.
6. Dirik, A.E., N. Menon, and J.C Birget. Modeling user choice in the PassPoints graphical password scheme. ACM SOUPS, 2007.
7. Jermyn, I., A. Mayer, F. Monrose, M.K. Reiter, and A.D. Rubin. The Design and Analysis of Graphical Passwords. 8th USENIX Security Symposium, 1999.
8. Nelson, D.L., U.S. Reed, and J.R. Walling. Picture Superiority Effect. Journal of Experimental Psychology: Human Learning and Memory 3, 485-497, 1977.
9. Passfaces. http://www.realuser.com Last accessed: December 1, 2006.
10. Peters, M. Revised Vandenberg & Kuse Mental Rotations Tests: forms MRT-A to MRT-D. Technical Report, Department of Psychology, University of Guelph, 1995.
11. Pinkas, B. and T. Sander. Securing Passwords Against Dictionary Attacks. ACM CCS, 2002.
12. Renaud, K. Evaluating Authentication Mechanisms. Chapter 6 in [4].
13. Renaud, K., and A. De Angeli. My password is here! An investigation into visio-spatial authentication mechanisms. Interacting with Computers 16, 1017-1041, 2004.
14. Suo, X, Y. Zhu, and G.S. Owen. Graphical Passwords: A Survey. Annual Computer Security Applications Conference, 2005.
15. Tari, F., A.A. Ozok, and S.H. Holden. A Comparison of Perceived and Real Shoulder-surfing Risks between Alphanumeric and Graphical Passwords. ACM SOUPS, 2006.
16. Thorpe, J. and P.C. van Oorschot. Human-Seeded Attacks and Exploiting Hot-Spots in Graphical Passwords. 16th USENIX Security Symposium, 2007.

17. van Oorschot, P.C., S. Stubblebine. On Countering Online Dictionary Attacks with Login Histories and Humans-in-the-Loop. ACM Trans. Information and System Security 9(3), 235-258, 2006.
18. Weinshall, D. Cognitive Authentication Schemes Safe Against Spyware (Short Paper). IEEE Symposium on Security and Privacy, 2006.
19. Wiedenbeck, S., J.C. Birget, A. Brodskiy, and N. Memon. Authentication Using Graphical Passwords: Effects of Tolerance and Image Choice. ACM SOUPS, 2005.
20. Wiedenbeck, S., 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 63, 102-127, 2005.
21. Yan, J., A. Blackwell, R. Anderson, and A. Grant. Password Memorability and Security: Empirical Results IEEE Security & Privacy Magazine, 2(5), 2004.