CCBS – A METHOD TO MAINTAIN MEMORABILITY, ACCURACY OF PASSWORD SUBMISSION AND THE EFFECTIVE PASSWORD SPACE IN CLICK-BASED VISUAL PASSWORDS

Haider al-Khateeb, Carsten Maple
Institute for Research in Applicable Computing, University of Bedfordshire, Luton, LU1 3JU, United Kingdom

ABSTRACT

Text passwords are vulnerable to many security attacks due to a number of reasons such as the insecure practices of end users who select weak passwords to maintain their long term memory. As such, visual password (VP) solutions were developed to maintain the security and usability of user authentication in collaborative systems. This paper focuses on the challenges facing click-based visual password systems and proposes a novel method in response to them. For instance, Hotspots reveal a serious vulnerability. They occur because users are attracted to specific parts of an image and neglect other areas. Undertaking image analysis to identify these high probability areas can assist dictionary attacks.

Another concern is that click-based systems do not guide users towards the correct click-point they are aiming to select. For instance, users might recall the correct spot or area but still fail to include their click within the tolerance distance around the original click-point which results in more incorrect password submissions.

Nevertheless, the Passpoints study by Wiedenbeck et al., 2005 inspected the retention of their VP in comparison with text passwords over the long term. Despite being cued-recall the successful rate of their VP submission was not superior to text passwords as it decreased from 85% (the instant retention on the day of registration) to 55% after 2 weeks. This result was identical to that of the text password in the same experiment. The successful submission rates after 6 weeks were also 55% for both VP and text passwords.

This paper addresses these issues, and then presents a novel method (CCBS) as a usable solution supported by an empirical proof. A user study is conducted and the results are evaluated against a comparative study.

KEYWORDS

Authentication, visual passwords, click-based systems, hotspots, password space

1. INTRODUCTION

Due to the limitation of current technology, text passwords are relatively secure against guessing, dictionary and brute-force attacks when their length is eight characters or more, consist of a complex mix of characters (digits, letters and symbols) and are absolutely random, but that is hard to achieve (Furnell, 2003) (Belgers, 1993). ASCII keyboards have 94 printable characters, hence in a traditional text based password system, given the advised length of eight characters there is a password space of $94^8 \approx 6 \times 10^{15}$ words. However, in practice, attackers exploit possible patterns to reduce the number of possible string combination and perform efficient dictionary attacks against the system. For example, if we assume that a group of users tend to use an English word as a password, the effective password space in this case will be equal to the number of words in the English dictionary, this is impossible to count accurately, but the number approaches three quarters of a million words only as estimated by Oxford Dictionary (AskOxford, 2009). As such, a text password regardless of its length has an effective password space much smaller than the theoretical space.

Similarly, click-based passwords are vulnerable to dictionary attacks as discussed and analysed in the following section. A click-based system is a VP authentication schemes in which the VP is a sequence of click-points on one image or more (Wiedenbeck et al., 2005b) (al-Khateeb et al., 2009). Users find the retention of a click-based VP easier if included within specific hotspots of an image. Some click-points are also easier to select based on their location. For instance, recalling then selecting a click-point visually

represented by the edge of a square is easier than selecting a click-point located inside the square in an empty space. As such, click-based systems require a method to support all their click-points with a memorable cue.

The remainder of this paper organised as follows: Section 2 provides background discussion of related researches. Section 3 proposes CCBS. The experiment's methodology is demonstrated in Section 4. Section 5 presents the results of the experiment and Section 6 discusses and concludes the results.

2. BACKGROUND AND RELATED WORK

The effective password space in click-based VPs: Click-based VP schemes have a large theoretical password space which can be increased further by adding more grid squares through expanding the clickable area or adding more images to the user portfolio. However, the effective password space in click-based schemes is significantly smaller than the theoretical space. That is because, if people are not guided or interrupted they are attracted to a limited number of predictable areas (hotspots) when looking at an image (Wolfe, 2000, Thorpe and Oorschot, 2007) (Erik, 2009) (al-Khateeb et al., 2010). Hence, hotspots can be used to perform an effective dictionary attack. (Thorpe and Oorschot, 2007) used data from a relatively small set of users to explore popular clusters and was able to correctly guess 36% of the passwords within 2^{31} guesses.

(Dirik et al., 2007) proposed a model to pre-identify hotspots in a given image. However, Section 2.1 of the same paper shows how carefully selected images are still vulnerable. Another proposal (Chiasson et al., 2008) highlights a random area in the image being used. A user may not click outside this area, but they can press a shuffle button to randomly reposition the highlighted area. This might help to achieve better distribution of clicks, but cannot increase the usability of low probability areas.

Accuracy of password submission: (al-Khateeb et al., 2010) shows that 70% of the incorrect clicks submitted by users were rejected due to exceeding tolerance by up to 4 pixels. (Wiedenbeck et al., 2005a) conducted a study to examine tolerance effect concluded that smaller tolerances (10x10 in their case) are harder to encode in users' memory, hence resulting in more incorrect password submissions. Nevertheless, retaining the VPs after one week shows that the number of incorrect submissions with the smaller tolerance (10x10) was significantly higher than the larger tolerance of 14x14.

This problem persists because password cues in click-based systems guide users towards areas but not specific click-points. Increasing the tolerance can eliminate this problem but it reduces the overall effective password space of the system. In CCP (Chiasson et al., 2007), every click results in a unique path of images until the VP is submitted. This helps the user to reselect a click-point before password submission if the consequent image is not part of their portfolio. While this can partially solve the problem, it can be time consuming and exposes the system to shoulder-surfing attacks as addressed by its authors.

VP retention in click-based systems: Cued-recall authentication such as click-based systems provides cues to trigger users' memory while entering their password. Each cue should aid the LTM to retain a particular task successfully. However, a laboratory study by (Wiedenbeck et al., 2005b) showed that the number of participants who failed to submit valid click-based passwords during the experiment was almost identical to that of users who were asked to retain text passwords. Success rates for both type of passwords decreased from 85% (instant retention on the day of registration) to 55% after 1 week from registration/first retention (R1) and the same percentage of 55% was achieved after 4 weeks from the second password retention (R2). This implies that the visual cues failed to significantly maintain users' memory to recall passwords.

3. CUED CLICK-BASED SYSTEM (CCBS)

We propose Cued Click-Based System (CCBS) as a method to overcome some of the main limitations of click-based systems discussed in the previous section. In CCBS, two types of cues are implemented to trigger the user's memory: graphical and textual, to retain and submit the correct click-points. Each image is transparently divided into click-cells representing the available symbols to form a VP. The visual cues to recall these click-cells (similar to other click-based systems) consist of all or part of the figures and features

of an image existing in the area of that particular cell. In addition, each click-cell is accompanied by a unique textual cue as illustrated Figure 1. The textual cue appears when the relevant cell is hovered by the mouse.

Figure 1. Visual and textual cues.

These cues are formed of short but informative sentences. It is essential to avoid confusing users by locating similar textual cues next to each other. For instance, if one cue is talking about London being the capital of the UK, click-cells next to it must contain different information that doesn't include keywords like London, Capital or the UK. CCBS has been developed with the following assumptions:

First, in response to the 'effective password space' problem: A uniform distribution of click-points (no hotspots) will be achieved via system-generated passwords while maintaining memorability. In this user study, we provide empirical evidence that system generated passwords in CCBS doesn't cause memorability problems in comparison with the comparative user studies (i.e. CCP and Passpoints).
Second, in response to the 'accuracy of password submission' problem: Users can accurately select the intended click-points hence the number of incorrect password submissions will significantly reduce.
Third, in response to the 'VP retention' problem: VP retention in CCBS will be significantly higher than the comparative schemes: Passpoints and Alphanumeric (text password) on the long term.

4. EXPERIMENT METHODOLOGY

Experimental design. The experiment continued for 6 weeks and consisted of 3 sessions. Session 1 was undertaken during week 1 in which participants were introduced to the system and asked to create a new user account using a VP that is randomly assigned to them. This was followed by a learning task were the VP is requested multiple times. Participants were then asked to complete a questionnaire. Finally, they retained their VP for the first time. Consequently, sessions 2 and 3 took place during weeks 2 and 6 to retain the VPs again. Session 3 was followed by another questionnaire about their experience with the system.

Materials. The system was implemented based on the HybridPass prototype (al-Khateeb et al., 2009) but the text password interfaces were excluded to match that of the comparative study: Passpoints (Wiedenbeck et al., 2005b). The clickable area display six different pictures and in addition to the visual representation of click-points, textual cues are used.

The same six images are used to create portfolios and random VPs to all users. The size of the clickable area was 230x100 pixels and the tolerance around the original click was set to 4 pixels, which represents each click-point with a 9x9 grid square. Hence, instead of returning the coordinates the system calculates an identifier of the grid square containing the click-point. As such, the VP space is 1.4×10^{16} .

A single computer was used in this experiment with a screen resolution of 1280 x 800. The experiment included a questionnaire in which the perception of end users towards the system is measured.

Procedure. The experiment was completed individually. Participants were first introduced to the experiment with a 5 minutes presentation. In session one the registration form included two input fields to capture the user ID, full name and an input method to capture the VP. However, VPs were not entered based on the participant's preference but rather randomly assigned to them. A unique VP formed of 5 click-points is shown to each participant during registration to adopt and use. They were asked to memorise these click-points and their order to select them again in the future. The registration form was validated using JavaScript, thus the 'Submit' button can be clicked if the ID and full name fields are filled and exactly 5 click-points are

selected. The following step is for password confirmation, participants are asked to re-enter their VP one more time. If the two passwords did not match, users are asked to repeat their registration.

The learning task consists of multiple password submissions until the participant succeeds to submit the correct password 10 times. The correct password is shown after each incorrect submission. Then to distract the participants from the system, they are asked to complete a questionnaire followed by a login trial after around 30 minutes, R1. In session 2, users are asked to retain their passwords for the second time, R2. If the password is wrong they can try again. After five attempts users can see their correct password to refresh their memory. Finally in session 3, password retention (R3) is followed by a questionnaire.

Participants. The comparative study had 20 participants taking part in the graphic password scheme. This study included the same number. Participants were computer science and business students who use computers on a regular basis. Most of them were Masters or PhD students. The mean age is 26.65 years (SD = 2.79) and the range was between 23 and 34 years. There were 11 females and 9 males in the sample.

5. ANALYSIS OF RESULTS

The results are evaluated against the comparative study: Passpoints (Wiedenbeck et al., 2005b).

5.1 Registration phase

Table 1 compares the time (of all attempts) and the number of attempts required to register a new user account with the Passpoints scheme and the alphanumeric password from the comparative study. Password confirmation time is reported for CCBS only because the comparative study did not include confirmation.

Student's *t*-test was used to analyse and compare the results. The total number of attempts to register a new account was less in CCBS. The difference was significant compared to the text password: $t(38) = 10.22$, $p < .005$ and not significant compared to Passpoints. In CCBS, 19 participants were able to register from the first attempt versus 11 of 20 participants who took two or more attempts to create a valid password. Registering a new account in CCBS took significantly more time in contrast with the text password: $t(38) = 3.3$, $p < .005$ and Passpoints: $t(38) = 7$, $p < .005$. However, considering that the difference in means is 31 seconds with the text password and 48 seconds with Passpoints, this does not imply a problem with the CCBS scheme considering that the time measured for Passpoints is for selecting 5 click-points correctly, while in CCBS it included inputting the user ID and full name as well.

Table 1. Calculated data of the total number of attempts and time (in seconds). Confirmation time is the time spent to re-enter the 5 click-points. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

	Scheme: Mean (SD)
Total attempts	CCBS: 1.05 (0.22) – Passpoints: 1.10 (0.07) – Text: 1.70 (0.18)
Total time to register	CCBS: 112.47 (21.79) - Passpoints: 64.03 (21.93) - Text: 81.10 (36.50)
Total time to confirm	CCBS: 54.04 (23.85)

Registration questions phase as shown in Table 2. *t*-test shows no significant difference between them.

Table 2. Questions about the registration phase. A Likert-scale of 7-points was used to answer each question with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

The question	Scheme: Mean (SD)
I did not have much trouble creating a password	CCBS: 2.35 (1.18) – Passpoints: 2.35 (1.57) - Text: 3.30 (1.59)
It did not take me long to create the password	CCBS: 2.95 (1.09) - Passpoints: 2.60 (1.42) - Text: 3.15 (1.63)

5.2 Learning phase

Table 3 shows the means and standard deviations of the number of incorrect submissions and submission time during the learning phase. Analysing the results using *t*-test shows that the number of incorrect submissions in the CCBS scheme was significantly less than Passpoints: $t(38) = 2.9$, $p < .01$ while the difference with the Text scheme did not reach significance. In CCBS 3 participants had a single incorrect submission compared to 4 participants with 1 incorrect submission and 2 with 2 incorrect submissions in the Text password scheme. Comparisons of the details are illustrated in Figure 2.

Table 3. Means and Standard Deviations (SD) of the total practice time (in seconds) and the number of incorrect password submissions in the learning phase. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

	Scheme: Mean (SD)
Number of incorrect submissions	CCBS: 0.15 (0.36) - Passpoints: 4.80 (7.16) - Text: 0.40 (0.68)
Total practice time	CCBS: 37.18 (11.10) - Passpoints: 171.89 (24.46) - Text: 66.08 (04.92)

Questions asked after the learning-phase are shown in Table 4. Analysis of question one shows the participants of the Text scheme agree that it did not take them long to input their passwords 10 times with a significant difference compared to Passpoints: $t(38) = 3.49$, $p < .005$ and CCBS: $t(38) = 5.87$, $p < .005$. There was no significant difference between CCBS and Passpoints. In question two there was a significant difference between CCBS and Passpoints: $t(38) = 2.6$, $p < .02$ and Text & Passpoints: $t(38) = 2.14$, $p < .05$. There was no significant difference between CCBS and Text. Further, no significant differences found in questions three and four.

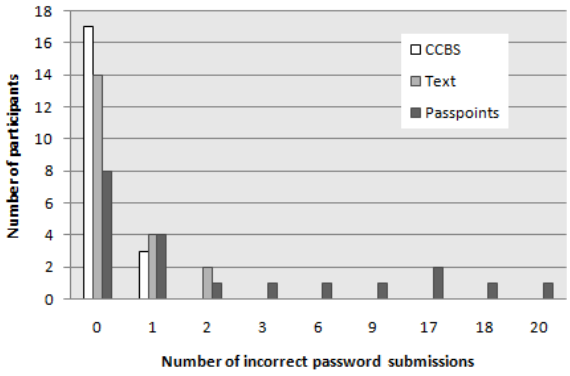

Figure 2. Incorrect submission (learning phase).

Finally, in question five there was a significant difference between CCBS and Text: $t(38) = 2.88$, $p < .01$ which implies that participants found that inputting a text password is easier than using CCBS. However, the difference between CCBS and Passpoints did not reach significance.

Table 4. Means and Standard Deviations (SD) of the learning phase questions. A Likert-scale of 7-points was used with lower numbers indicating strong agreement (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

	The question	Scheme: Mean (SD)
1.	It did not take me long to input my password 10 times	CCBS: 4.1 (1.74) - Passpoints: 3.40 (2.14) - Text: 1.65 (0.67)
2.	Once I created my password I was able to input it correctly	CCBS: 1.5 (0.51) - Passpoints: 2.60 (1.82) - Text: 1.65 (0.79)
3.	My password input got better with practice	CCBS: 1.2 (0.41) - Passpoints: 1.15 (0.50) - Text: 1.20 (0.52)
4.	Inputting my password was easy	CCBS: 2.55 (1.23) - Passpoints: 2.70 (2.18)- Text: 1.90 (1.02)
5.	Inputting my password was fast	CCBS: 4.0 (1.71) - Passpoints: 3.05 (1.73) - Text: 2.35 (1.14)

5.3 Retention phase

The number of incorrect submissions and time for the correct password submission are measures and compared to the results from the comparative study in Table 5. There were no incorrect submissions for CCBS in R1. *t*-test shows a significant difference with Passpoints: $t(38) = 4.41$, $p < .005$ and no significant difference with Text. In R2, the number of incorrect submissions in CCBS are significantly less than Passpoints: $t(38) = 2.3$, $p < .05$ and Text: $t(38) = 2.28$, $p < .05$. In R3, the number of incorrect submissions in CCBS is fewer but no significant difference was found with Passpoints or Text.

Table 5. Means and Standard Deviations (SD) of correct submissions time (in seconds). And the number of incorrect password submissions in the retention phase. (CCBS $N = 20$, Passpoints $N = 20$, Text $N = 20$).

	Scheme	Mean R1 (SD)	Mean R2 (SD)	Mean R3 (SD)
Number of incorrect submissions	CCBS	0.0 (0.0)	0.6 (1.53)	0.55 (1.53)
	Passpoints	1.55 (1.57)	2.75 (3.88)	1.50 (2.80)
	Text	0.25 (0.79)	2.20 (2.73)	1.75 (2.47)
Time for correct submission	CCBS	27.9 (7.63)	69.55 (38.22)	59.65 (24.05)
	Passpoints	8.78 (4.40)	24.25 (15.21)	19.38 (17.57)
	Text	5.23 (1.66)	9.42 (3.70)	9.24 (03.72)

Time for correct submissions is more in CCBS across R1, R2 and R3. In R1, the difference was significant with Passpoints: $t(38) = 9.7$, $p < .001$ and Text $t(38) = 12.98$, $p < .001$. In R2, the difference was significant with Passpoints: $t(38) = 4.92$, $p < .001$ and Text: $t(38) = 7$, $p < .001$. In R3, the difference was significant with Passpoints: $t(38) = 6.04$, $p < .001$ and Text: $t(38) = 9.26$, $p < .001$. Further, the number of participants who failed to submit the correct password at their first attempt in each session is calculated. The result is illustrated in Figure 4 and compared to Passpoints and Text. Figure 3 illustrates the number of participants who failed to submit the correct password after five attempts.

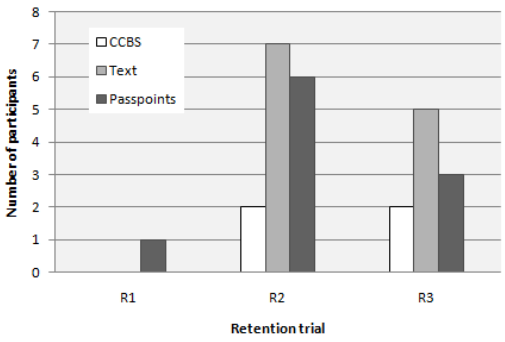

Figure 3. Participants who failed to submit their correct password after five attempts

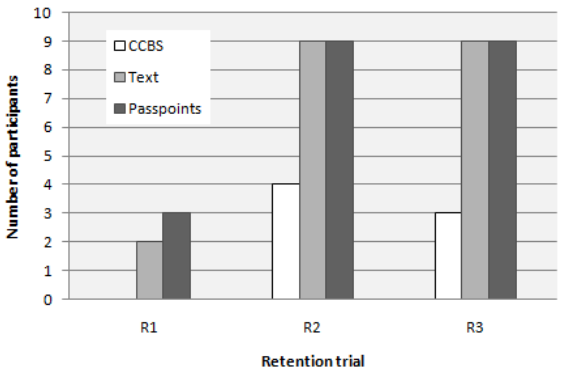

Figure 4. Participants who failed to submit their correct password on the first attempt.

Four questions were asked after the final retention phase as shown in Table 6. *t*-test analyses of the results shows no significant difference in question one. In question two, there was one significant difference between CCBS and Text: $t(38) = 3.71$, $p < .01$. Hence, participants of the text scheme agree more that inputting their password was faster. The same apply to question three, there was one significant difference between CCBS and Text: $t(38) = 2.39$, $p < .05$. The participants in this question agree more than CCBS was pleasant to use. Finally, there was a significant difference between Passpoints and Text: $t(38) = 2.81$, $p < .01$ and between CCBS and Text: $t(38) = 4.09$, $p < .001$. Hence, there was significant agreement that remembering the password is easier in CCBS and Passpoints.

Table 6. Means and Standard Deviations (SD) of questions about the retention phase. A Likert-scale of 7-points was used with lower numbers indicating agreement (CCBS *N* = 20, Passpoints *N* = 20, Text *N* = 20).

	The question	Scheme: Mean (SD)
1.	Inputting my password was easy	CCBS: 2.45 (0.99) - Passpoints: 2.70 (2.18) - Text: 1.90 (1.02)
2.	Inputting my password was fast	CCBS: 4.05 (1.70) - Passpoints: 3.05 (1.73) - Text: 2.35 (1.14)
3.	I think the password system was pleasant to use	CCBS: 2.10 (1.02) - Passpoints: 2.40 (1.57) - Text: 3.00 (1.34)
4.	I think that the rules [about password creation] make it easy to remember the password	CCBS: 3.35 (1.18) - Passpoints: 3.55 (2.09) - Text: 5.25 (1.71)

5.3 Click-points

A PHP function called rand() was used to generate random click-points for this experiment. The analysis of these clicks shows no hotspots in any of the employed images as seen in Figure 5. In the illustration each image is divided into 36 squares of 27x27 pixels and the probability distribution of each square is calculated. The equal distribution line represents the case if all clicks are distributed equally among the 36 squares.

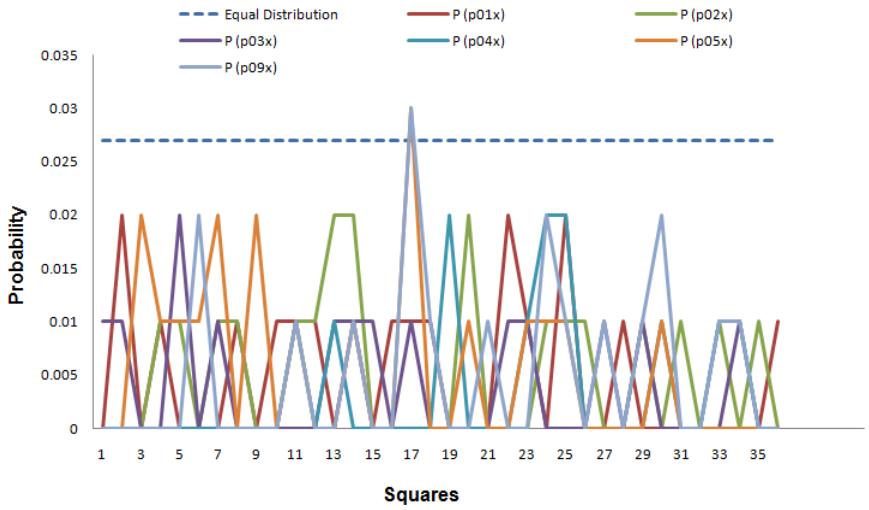

Figure 5. Line chart illustrating the probability distribution of 100 clicks. As such, P (p01x) is the Probability of the image: p01x. The 'Equal Distribution' line represents the case if all clicks are distributed equally.

6. DISCUSSION AND CONCLUSION

No hotspots emerged and using system-generated passwords (click-points) did not affect user memorability. Hence, password retention in R1, R2 and R3 for CCBS was better than the comparative schemes, hence memorability was maintained. In addition, users accurately selected the intended click-points to input their VPs hence the number of incorrect password submissions in the learning phase using CCBS was less than the two other schemes. The difference was significant with Passpoints. Further, in the retention phase there was no incorrect submission in R1. The number was also less in R2 and R3. The difference in R2 was significant. Nevertheless, the illustration in Figure 4 shows success rates for Passpoints and Text to be 85% and 90% respectively in R1. Success rates were then decreased to 55% in R2 and R3. Meanwhile, success rates using CCBS were 100% in R1, 80% in R2 and 85% in R3. CCBS rates in R3 were identical to that of the instant retention using Passpoints in R1.

CCBS users found it easy to register a new account and the scheme succeeded to maintain high registration rate while enforcing system generated passwords. However, they required more time. This difference is acceptable considering that 64 seconds in Passpoints is the time required to input 5 click-points, while 112 s in CCBS included time to enter the user ID and full name in addition to selecting 5 click-points.

In the learning phase the results imply that the technique used in CCBS was very successful in guiding the users to select the correct click-points. The effect was also reflected in significantly shorter total learning time compared to Passpoints. The authors of the comparative study concluded that the most common problem in Passpoints was clicking close to the correct click-point but outside tolerance which supports our explanation of the problem. CCBS resists this via a confirmation message sent to the user before selecting a particular click-point although tolerance used in CCBS is 9x9 that is less than that of Passpoints 20x20.

A further advantage of the CCBS design is its flexibility to further development. For instance, potential enhancements are possible to consider shoulder-surfing and provide keyboard support. Shoulder-surfing is a concern for VP since password submission is exposed on the monitor. Keyboard support is critical because it is less visible to other individuals. This can be achieved through adding a unique code at the beginning of every textual cue. The format can be: 'click-point code: textual cue', as such if the code is A3B, the result is: 'A3B: Venus is the Roman goddess of love and beauty'. After locating the correct click-point, the user can choose to either click or else type the relevant codes in a password field to submit the password using the keyboard. Nevertheless, click-points can also be highlighted using the keyboard alone. This is possible in CCBS because the coordinates of each click-point is consistent in the image. Whilst in Passpoints, the coordinates/area of a click-point is calculated after a mouse-click event (not consistent).

REFERENCES

AL-KHATEEB, H., MAPLE, C. & CONRAD, M. 2009. HybridPass: Authentication Mechanism for Web Applications-Both Secure and User-Friendly. *In:* KOMMERS, P. & ISAÍAS, P. (eds.) *IADIS International Conference e-Society 2009.* Barcelona, Spain.

AL-KHATEEB, H., MAPLE, C. & CONRAD, M. 2010. Enhancing Usability and Security in Click-Based Visual Password Systems. *IADIS International Conference e-Society 2010.* Porto, Portugal.

ASKOXFORD. 2009. *How many words are there in the English language?* [Online]. Available: http://www.askoxford.com/asktheexperts/faq/aboutenglish/numberwords?view=uk [Accessed Oct 25 2009].

BELGERS, W. 1993. *UNIX Password Security* [Online]. Available: http://www.het.brown.edu/guide/UNIX-password-security.txt [Accessed March 06 2010].

CHIASSON, S., FORGET, A., BIDDLE, R. & OORSCHOT, P. C. V. 2008. Influencing users towards better passwords: persuasive cued click-points. *Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008: People and Computers XXII: Culture, Creativity, Interaction - Volume 1.* Liverpool, United Kingdom: British Computer Society.

CHIASSON, S., OORSCHOT, P. C. V. & BIDDLE, R. 2007. Graphical Password Authentication Using Cued Click-points. *ESORICS 2007.*

DIRIK, A. E., MEMON, N. & BIRGET, J.-C. 2007. Modeling user choice in the PassPoints graphical password scheme. *Proceedings of the 3rd symposium on Usable privacy and security.* Pittsburgh, Pennsylvania: ACM.

ERIK. 2009. *PassClicks* [Online]. Available: http://labs.mininova.org/passclicks/ [Accessed Oct 26 2009].

FURNELL, S. 2003. Cybercrime: vandalizing the information society. *International Conference on Web Engineering, ICWE 2003.* Oviedo, Spain: Springer.

THORPE, J. & OORSCHOT, P. C. V. 2007. Human-seeded attacks and exploiting hot-spots in graphical passwords. *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium.* Boston, MA: USENIX Association.

WIEDENBECK, S., WATERS, J., BIRGET, J.-C., BRODSKIY, A. & MEMON, N. 2005a. Authentication using graphical passwords: effects of tolerance and image choice. *Symposium on Usable Privacy and Security (SOUPS'05).* Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA: ACM Press: New York, NY, USA.

WIEDENBECK, S., WATERS, J., BIRGET, J.-C., BRODSKIY, A. & MEMON, N. 2005b. PassPoints: design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies,* 63**,** 102-127.

WOLFE, J. 2000. Visual Attention. *In:* KK, D. V. (ed.) 2nd ed. San Diego, CA: Academic Press.