Journal of Computer Applications (JCA)
ISSN: 0974-1925, Volume IV, Issue 3, 2011

Automated Attacks On Pass Point-Style Graphical Passwords

K.Dinesh Kumar [a,*]

Abstract - Users click on one point per image for a sequence of images, the next image is based on the previous click-point. Users preferred Cued Click Points (CCP) to Pass Points 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. Purely automated attacks against Pass Points-style graphical passwords is introduced and evaluated. For generating these attacks, a graph-based algorithm is developed to efficiently create dictionaries based on heuristics such as click-order patterns (e.g., five points all along a line). Some of methods combine click-order heuristics with focus-of-attention scan-paths generated from a computational model of visual attention, yielding significantly better automated attacks than previous work. One resulting automated attack finds 7%-16% of passwords for two representative images using dictionaries of approximately 2^{26} entries (where the full password space is 2^{43}). Relaxing click-order patterns substantially increased the attack efficacy albeit with larger dictionaries of approximately 2^{35} entries, allowing attacks that guessed 48%-54% of passwords (compared to previous results of 1% and 9% on the same dataset for two images with 2^{35} guesses). These latter attacks are independent of focus-of-attention models, and are based on image-independent guessing patterns. This method uses multiple images require serious consideration when deploying basic Pass Points-style graphical passwords.

Index Terms - *Cued Click Points, Pass Points heuristics*

.

I. INTRODUCTION

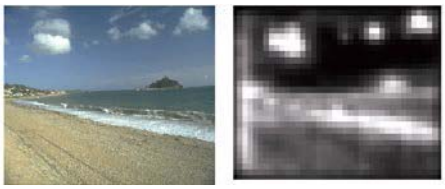
Users preferred Cued Click Points (CCP) to Pass Points selecting and remembering only one point per image was easier, and that seeing each image triggered their memory of where the corresponding point was located. Purely automated attacks against Pass Points-style graphical passwords are introduced and evaluated. Graphical passwords are an alternative to text passwords, whereby a user is asked to remember an image (or parts of an image) instead of a word. They are motivated in part by the well-known fact that people have superior memo ability for images and the promise of their suitability for small devices such as smart phones.

Manuscript received, 2011.

K.Dinesh Kumar [a,*]
BE-ECE-Fina Year
Electronics and Communication Engg..
Kings College of Engineering,
Punalkulam, pudhukottai (DT)
E-mail:vkjdinesh@gmail.com

II. GRAPHICAL PASS POINTS

Graphical passwords have become an active topic of research with many new proposals. One proposal of interest, Pass Points involves a user creating a 5-point click sequence on a background image. Usability studies have indicated that these graphical passwords have reasonable login and creation times, acceptable error rates, decent general perception and less interference between multiple passwords when compared to text passwords. Our research improves our understanding of the security of Pass Points-style graphical passwords, i.e., schemes closely resembling Pass Points, wherein a user creates a click sequence of r points (e.g., r = 5) on a single background image. Pass Points-style graphical passwords have been shown to be susceptible to hot-spots, which can be exploited in human-seeded attacks whereby human-computed data (harvesting click-points from a small set of users) is used to facilitate efficient attacks. These attacks require that the attacker collect sufficient "human-computed" data for the target image, which is more costly for systems with multiple images. This leads us to ask whether more scalable attacks exist, and in particular, effective fully-automated attacks.



Our attack method is based on the hypothesis that users are more likely to choose click-points relating to predictable preferences, e.g., logically grouping the click-points through a click-order pattern (such as five points in a straight line), and/or choosing click-points in the areas of the image that their attention is naturally drawn towards. To find parts of the image that users are more likely to attend to (salient parts of the image). Method examines click-order patterns both alone and in combination with these more salient parts of the image. Attacks employ sets of graphical passwords that we hypothesize are "more likely" to be chosen than others; these sets naturally define dictionaries for use in a dictionary attack. A successful such attack must be able to efficiently generate a dictionary containing highly probable passwords. In existing literature, the size of a dictionary is normally considered the most important cost for a dictionary attack, whereas the cost of dictionary generation is often neglected; the latter is reasonable if a one-time precipitation can be reused. Alternately, if the dictionary must be generated on-the-fly, or recomputed each time (e.g., for a different background image), then the cost of dictionary generation may become as or more important than the size of the dictionary itself. A graph-based algorithm is developed for attack dictionary generation whose computational cost is on the order of the number of dictionary entries. Given an alphabet, it can effi- cogently generate r-permutations that also satisfy a predefined set of conditions (e.g., click-order heuristics). This method is more efficient than

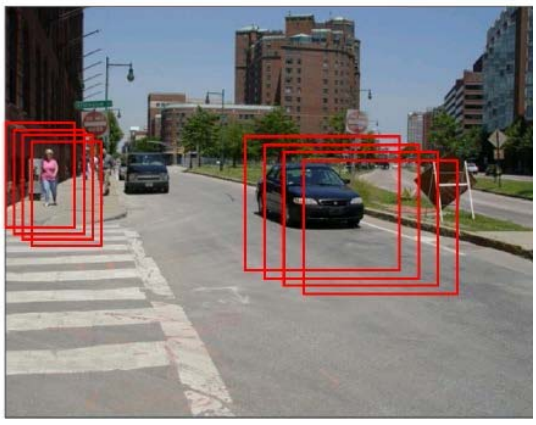
generating all possible r-permutations from the alphabet and then checking which ones satisfy a predefined condition. Note that each r-permutation is a single ordered arrangement of r points, and that we use the term “all possible r-permutations” to describe all possible ordered arrangements of r points. Our methods are substantially more successful than previous purely automated attacks. Some of our dictionaries find 19-83% as many passwords as human-seeded attacks (when based on independent probabilities), with only about 3% as many dictionary entries.

III. RELATED WORK

On click-based graphical password schemes, wherein a user clicks on a set of points on one or more presented background images, and work related to guessing attacks on graphical passwords. One way that an attacker could predict hot-spots is by using image processing tools to locate areas of interest Basic click-order patterns were first introduced and evaluated in combination with human-seeded attacks the only pattern in common with the present work is regular DIAG (i.e., without any “laziness" relaxation). Analyzing a set of patterns for three click-based graphical password schemes: Pass Points and two variants named Cued Click-Points (CCP) and Persuasive Cued Click-Points (PCCP). In CCP and PCCP, a user clicks on a single point on each of five images, where each image (except the first image) is dependent on the previous click-point. They show that the design of the interface impacts whether users select click-points in some predictable patterns, and implied that such patterns in user choice might reduce the effective password space.

The present work mathematically models click-order patterns and uses them to mount purely automated attacks, demonstrating and experimentally quantifying the degree to which certain patterns can be used to efficiently search the password space. Human-seeded attacks introduced and demonstrate their efficacy against Pass points-style graphical passwords. Human-computed data sets (harvesting click-points from a small set of users) were used in two human-seeded attacks against passwords from a field study on two different images: one based on a first-order Markov model another based on an independent probability model .Using their human-computed data sets (harvested from a single-session lab study), a dictionary based on independent probabilities contained 231.1−233.4 entries and found 20-36% of field study passwords, and a dictionary based on the first-order Markov model found 4-10% of field study passwords within 100 guesses. These attacks require the attacker to collect sufficient click-points for each image, and are image dependent, thus requiring per-image costs for systems with multiple images.

IV. MODELS OF VISUAL ATTENTION

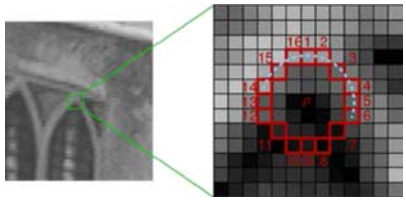


The general idea is that areas of an image will be salient (or visually “stand out") when they differ from their surroundings. Given an input image, Nitti’s model outputs a focus-of-attention scan-path to model the locations and the order in which a human might automatically and unconsciously attend these parts of the image. The model first constructs a saliency map based on visual features. Then it uses a winner-take-all neural network with inhibition of return to define a specific focus-of-attention scan-path, intended to represent the order in which a user would scan the image. In stage 1, the saliency map is created by decomposing the original image into a set of 50 multi-levels “feature maps", which extract spatial discontinuities based on color opponency (either red-green or blue-yellow), intensity, or orientation. Each level defines a different size of the center and its surround, in order to account for conspicuous locations of various sizes. All feature maps are then combined into a single saliency map. In stage 2, the neural network detects the point of highest salience (as indicated by the intensity value of the saliency map), and draws the focus of attention towards this location. Once an area has been attended to, inhibition of return will prevent the area from being the focus again for a period of time. Together, the neural network with inhibition of return produces output in the form of patio-temporal attention scan-paths, which follow the order of decreasing saliency as defined by stage 1. Two different normalization types (producing different scan-paths) can be used with the model: Local ax and Iterative (cf. Figure 1). In Local ax normalization, the neural network will have a bias towards those areas that are closer to the previously attended location. In Iterative normalization, the neural network will find the next most salient area that has not been inhibited. We use Local ax herein.

V. IDENTIFYING DISTINGUISHABLE POINTS

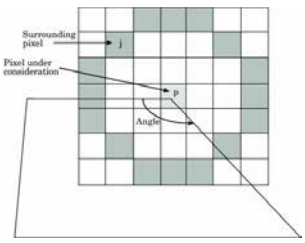
Overall method is based on the hypothesis that users are more likely to choose passwords insisting of click-points, each of which is a distinguishable point, defined as a point on a digital image that can be easily distinguished and located again by a user – e.g., by using reference able points on the image (such as a corner), or calculable points based on other reference able parts of the image (such as object centers). corner detection to find reference able points, previous work used cancroids to find calculable points.

5.1 Corner Detection



A corner is defined as the intersection of two edges, where an edge is defined by the points in a digital image where there are sharp changes in intensity. Harris corner detection used as implemented. This first identifies the edges. Those edges are then blurred to reduce the effect of any noise. Next, based on the edges, an energy map is generated, containing local maxima and minima.

5.2 Centric Detection



To find the centers of objects, first partition the digital image into segments using image segmentation, by the mean-shift

segmentation algorithm which takes a feature (range) bandwidth, spatial bandwidth, and a minimum region area (in pixels) as input. Set these parameters to 7, 9, and 50 respectively, which we found empirically to provide an acceptable segmentation with the smallest resulting number of segments.

(a) Local ax normalization (b) Iterative normalization

Figure 1. Pool image with the first 7 steps in the scan-path.

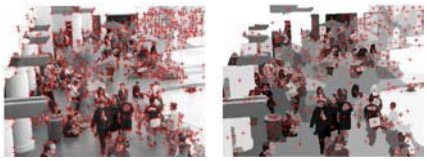


Figure 2. Corner detection (left) and center detection (right) for pool image.

VI. WINDOW CLUSTERING ALGORITHM

A window cluster is a square region of n× n pixels for some positive integer n. A cluster is a set of one or more points that lie within a window cluster. The geometric center of a window cluster is used as the representative of all the points within the window cluster. An alphabet is a set of window centers. Assume that an attacker's goal is to guess the largest number of passwords with the fewest guesses. After creating a set of points for a guessing alphabet (which might be used in passwords in any ordering of five clicks), those within the same tolerance region could be redundant (effectively guessing the same point).

The term clustering to mean normalizing a set of points to a single value. The intuition behind clustering is that given the system error tolerance, one point would be accepted as a correct entry for all others within its tolerance region. A clustering algorithm (window clustering) introduced, based on setting a window of fixed size (not necessarily the same size as the tolerance region) over the largest number of points it can cover.

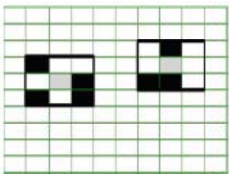


Figure 3. Window clustering.

Figure 3 shows an example set of candidate points with black squares, where each square represents a pixel. These 7 candidate points are covered with two 3×3 windows and will be represented by the centers of the two windows illustrated with grey squares. Replace those candidate points inside the window with the geometric center of the window. Thus, the center of the cluster is not necessarily one of the original input points (in contrast to a previous clustering algorithm).

More precisely, window clustering is a greedy algorithm with a fixed window size. Starting with all candidate points, find the next position for the window that covers the maximum number of remaining points (ties are broken arbitrarily). Then store the center of the window to represent the points in the window, and erase the corresponding points. Continue this process until no candidate points remain. The candidate points we use are the points with value 1 in Bi of Section IV-B1, and the window size is set to 19 × 19.

VII. Conclusion

Finally, our attacks could be used to help inform more secure design choices in implementing Pass Points-style graphical passwords. Proactive checking rules for Pass Points-style graphical passwords might be created based on the click-order pattern attacks herein; for example, disallowing LINE or DIAG patterns (for all laziness modes), and disallowing passwords where too few click-points are further than 150 pixels away from the previous click-point. Of course, any such proactive checking rules would need to be tested to ensure that the usability impact is acceptable and that security is not impacted in other unexpected ways.

REFERENCES

[1] P.C. van Borscht, Airmail Salehi-Abari, Julie Thorpe. Purely Automated Attacks on Pass Points-Style Graphical Passwords School of Computer Science, Carleton University

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

[3] S. Chanson, A. Forget, R. Biddle, and P.C. van Oorschot. Influencing Users Towards Better Passwords: Persuasive Cued Click-Points. In Proceedings of HCI, British Computer Society, 2008.

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

[5] S. Chiasson, A. Forget, E. Sober, P.C. van Oorschot, and R. Biddle. Multiple Password Interference in Text Passwords and Click-Based Graphical Passwords. In 16th ACM Conference on Computer and Communications Security (CCS), 2009.

BIOGRAPHY

Mr.K. Dinesh Kumar, from Periyakottai, Needamangalam, Thiruvarur, doing BE(ECE) at Kings College of Engg.. Punalkulam, Pudukottai. With a CGPA of 8.1 (upto 6th sem). His area of interest is Computer Networks, He has presented more than 6 papers in various colleges technical symposiums. And also published in one international & National conference paper publish.