A Ogle Sign-Based User Authentication System to Counter Shoulder-Surfing Attacks

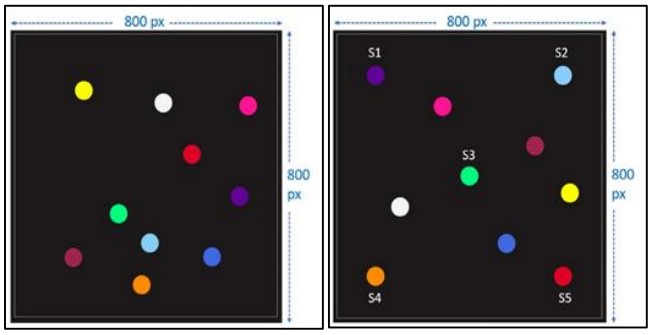
Pooja N[1] Nikshita[2] Priyanka S N[3] Tharunya U P[4] Prof. Sharan L Pais[5]
[1,2,3,4]Student [5]Assistant Professor
[1,2,3,4,5]Department of Information Science and Engineering
[1,2,3,4,5]Alvas Institute of Engineering and Technology, Mijar, India

Abstract— Shoulder surfing helps to gain the authentication details of a victim to attacker through observations and nowadays it is becoming a very big threat to visual privacy. We present DyOglePass: Dynamic Ogle Passwords, an authentication strategy that uses dynamic ogle sign. Here we present two authentication interfaces, a dynamic and a static-dynamic interface, that helps this strategy to counter shoulder surfing attacks. The main idea is to authenticate the user by following uniquely colored circles that move along random paths on the screen. After so many evaluations, we discuss how the authentication accuracy varies with respect to transition speed of the circles, and the number of moving and static circles. Furthermore, we evaluate the resiliency of our authentication method by comparing it to a ogle- and PIN-based authentication system. Finally, we found that the static-dynamic interface with a transition speed of two seconds was the most effective authentication method with a greater accuracy.

Keywords: Shoulder-Surfing, Ogle, Authentication, Static, Dynamic

I. INTRODUCTION

Shoulder surfing generally refers to unsolicited access to a user's confidential information (e.g., interests, hobbies, sexual preferences, login credentials, etc.) by an observer. In this work, we focus on preventing shoulder surfing attacks on a knowledge-based authentication method, i.e., using passwords specifically in public and semi-private spaces. Keypad monitoring commonly occurs at public places like ATMs, kiosks, airport lounges, coffee shops, airplanes, and semi-private spaces like offices, aimed at stealing the login credentials of users. A report on global visual hacking, presented by Ponemon Institute in 2016, found that in business office environments the attacks happen on laptops, tablets, smartphones, etc. They conducted shoulder surfing attacks in eight countries, and a staggering 91% of visual attacks were successful, resulting in 613 units of breached data of various types. Furthermore, 11% (69 units) of the breached data were login credentials. To prevent shoulder surfing, we focus on ogle-based authentication, which has been previously explored. The existing solutions are limited by low accuracy, the need for precise ogle input, accurate recall of the sign by users, and susceptibility to video analysis attacks.



(a) (b)

Fig. 1: a) Dynamic authentication interface with 10 uniquely colored circles placed at random positions. b) Static dynamic authentication interface comprising 5 static (S1, S2, S3, S4, S5) and 5 dynamic circles.

We present, a ogle sign-based approach to addressing shoulder surfing on user authentication. The central idea of our authentication system is, the interface comprises of 10 uniquely colored circles which move simultaneously along random paths during an animation of N seconds. An animation is a time interval where all the circles move from their source to destination locations. Analogous to a four digit PIN, a user selects a set of four colors (out of 10) as her password. To authenticate, the user follows the path of the circle, colored with her password color, during an animation. This animation is repeated four times allowing the user to enter all four colors. For example, if the user's password is "red-blue-yellow-green" the user follows the red colored circle during the first animation, blue during the second, and so on. For a successful authentication, the scan-path of the user's ogle should match with the path of the colored circle selected as the password, in each animation, for all the four animations. We hypothesize that people would be able to remember their password by associating the password colors with their favorite colors, the colors of the objects they frequently use (car, cloth), etc. The two authentication interfaces we have developed are the "dynamic interface" shown in Figure 1a and the "static dynamic interface" show in Figure 1b. Since we targeted our authentication system to be deployable at ATMs, kiosks, laptops, or in general, devices with smaller screens, we surveyed screen dimensions 2 of ATMs by various vendors. There is no single standardized size of the ATM screen, but commonly used dimensions include 8", 10.1", 12.1", 15". We chose a median size of 11.5" which approximately translates to a dimension of 800×800 pixels on a screen with 98.44 PPI (screen size 1900×1200 px, 23") used in our experiments. We evaluated our solutions through a two phase user study. In the first phase, both dynamic and static dynamic interfaces were tested for their accuracy under two animation speeds: 3 and 2 seconds. Since the static-dynamic interface was found to be a more practical

solution, we tested its susceptibility to video analysis attack in the second phase.

II. LITERATURE SURVEY

Kumar at al., [1], presented "Eye Password," an authentication method where the user enters sensitive input like password or PIN by selecting from an onscreen keyboard using their gaze.

Khamis et al., [2], presented "Gaze Touch Pass," that allows authentication on mobile phones though multiple switches between gaze and touch input modalities.

De Luca et al. [3], presented "Eye-Pass-Shapes", where a user authenticates by drawing one of the eight gestures with their eye movements. The system evaluations showed that the eye gestures significantly increases security while being easy to use.

Best et al. [4], presented a rotary interface for gaze-based PIN code entry. Their solution is based on the weighted voting scheme of numerals whose boundaries are crossed by the streaming gaze points. The authentication accuracy was found to be 71.16% with the PIN interface and 64.20% with the rotary interface.

Rajanna et al. [5] presented an intelligent gaze gesture based authentication system to counter shoulder surfing attacks. A user authenticates by her unique gaze patterns onto moving geometric shapes.

Vidal et al. [6] presented the idea and the design of authentication using eye pursuits. The authors proposed a pursuits-enabled screen that displays an animation of fishes swimming in the fish tank. The user can authenticate by looking at four specific fishes in the precise sequence.

Cymek et al. [7], presented an authentication method, where the user follows the digits moving in vertical and horizontal directions to authenticate. The system accuracy was 97.57% in recognizing the digits entered. As discussed before, the common limitation with ogle-based authentication systems is having a low accuracy.

As evaluated by De Luca et al. [8], the error rate of various well known gaze-based authentication methods varied from 9.5% to 23.8%. Also, gaze authentication is susceptible to video analysis attacks.

III. SYSTEM ARCHITECTURE

Our ogle sign-based authentication system consists of two main modules: 1) Ogle Tracking Module, and 2) Authentication Engine. A working model of the system is depicted in Figure 2a.

Ogle Tracking Module: The system uses "The Eye Tribe" tracker, which is a table mounted eye tracking sensor that provides (X,Y) coordinates of the user's gaze on the screen at 60 Hz. For the eye tracker to work efficiently, the user is positioned such that the face is centered in front of the monitor at a distance of 45 - 75 cm.

Authentication Engine: The authentication engine is the central module that runs on a computer and receives (X,Y) gaze-coordinates from the eye tracker. This module is responsible for positioning the circles at random locations on the interface, and generating a random path for each circle. The module also implements the scan-path matching algorithm to authenticate a user.

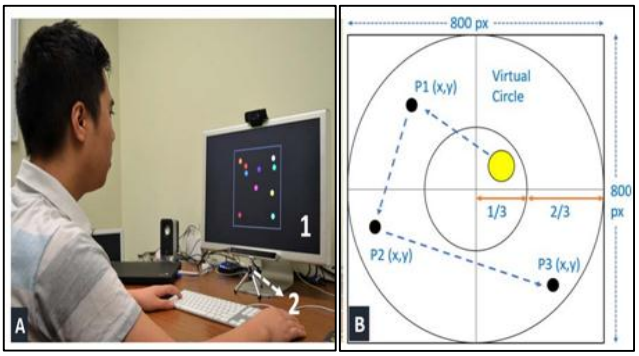


Fig. 2: a) A user is authenticating by following the paths of four uniquely colored circles chosen as the password (1 - Authentication interface, 2 - Eye tracker) b) Distribution of Random Points: distribution of random points (P1, P2, P3) for the path of a circle (yellow). The random points are beyond 1/3 distance from the center along the radius of the virtual circular boundary.

A. Authentication Procedure

The authentication procedure comprises of two processes: 1) one-time password selection, and 2) password entry. The user selects four colors from a password selection window, one for each of the four animations that form the password of the user. The user controls the authentication interface through a set of hot-keys on a keyboard: 'A' to initiate movement of the circles (start animation) and record ogle data, 'Z' to recover from user mistakes (blink, sneeze, losing the path, etc.) and discard recorded ogle data, and 'M' to submit the password after following four circles.

IV. AUTHENTICATION INTERFACE DYNAMICS

Both the dynamic and static-dynamic interfaces have 10 colored circles and have dimensions of 800×800 px, but they differ in the number of moving circles during an animation. The two mechanisms that are common to both the interfaces are: 1) random point generation, 2) generation of the animation path and template for each colored circle.

A. Random Point Generation Algorithm

To generate n number of random points that are uniformly distributed inside a circle with radius Rc and area A, we employ the method proposed by Leon-Garcia et al., The joint probability distribution function (PDF) of the random points inside the circle, i.e., the joint distribution of random variable X and random variable Y representing the x and y coordinates of a random point.

1) Initial Points for Circles

The initial locations of the circles on the interface are random but uniformly distributed within the radius of 100 pixels from the center of the interface (with dimensions of 800×800). To generate the points, we use the Random Point Generation.

B. Generating Animation Paths and Templates

Once the initial points are generated for the 10 circles, we generate a random path for each circle. Each random path is a set of three points that the circle traverses through from its initial point. To generate the three points for a random path, we use the same method of generating uniformly distributed random points.

C. Scan-path Matching Algorithm

We match the user's scan-path against a circle's traversed path through the "Template Matching" algorithm, where we compute the root-mean-square distance of the candidate path (user's scan-path) from all the template paths (circles' traversed paths). The template path of a circle that is at the least root-mean-square distance from the candidate path is chosen as the circle (color) followed by the user.

V. Ogle- and PIN-based Authentication

The PIN-based authenticating interface uses a standard layout of numbers arranged in a 4×3 grid as seen on most ATMs. For consistency in comparison, the numeric grid was also placed in the space of a 800×800px square. A user authenticates with a PIN of 4 digits, and to enter the PIN with ogle, the user first looks at the digit and presses a hot key (A). We chose a hot key as we wanted to be consistent with the activation method on the colored circles interface where an animation is initiated by pressing a hot key(A).

A. PIN Recognition

PIN recognition is a simple process that uses Euclidean distance between the points. For each recorded gaze point, we compute the Euclidean distance to the center of every digit on the interface. The digit at the least distance from the gaze point is selected as the digit entered by the user. If all the 4 digits entered by the user match with the PIN, the user is authenticated.

VI. Limitations

Few limitations of our approach are that users with color blindness will have limited set of colors to choose from as they cannot distinguish few colors. A solution would be to have different shapes with different colors, while creating no visual overload. Furthermore, an authentication time of ~ 8 seconds would not be appropriate in cases where the authentication is performed frequently.

VII. Conclusion and Future Work

We presented a dynamic ogle sign-based authentication system to counter shoulder surfing attacks. We also explored two authentication interfaces: 1) a dynamic interface, and 2) a static-dynamic interface that leverage ogle signs. Through system evaluations, we found that the static-dynamic interface with an animation speed of 2 seconds is the most practical solution of the two interfaces. As future enhancements, we will try to reduce the authentication time and improve the password memorability through interface modifications.

References

[1] D. S. Best and A. T. Duchowski. A rotary dial for gaze based pin entry. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ETRA '16, pages 69–76. ACM, 2016.
[2] A. Bulling, F. Alt, and A. Schmidt. Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12. ACM, 2012.
[3] D. H. Cymek, A. C. Venjakob, S. Ruff, O. H.-M. Lutz, S. Hofmann, and M. Roetting. Entering pin codes by smooth pursuit eye movements. Journal of Eye Movement Research, 2014.
[4] A. De Luca, M. Denzel, and H. Hussmann. Look into my eyes!: Can you guess my password? In Proceedings of the 5th Symposium on Usable Privacy and Security, SOUPS '09, pages 7:1–7:12. ACM, 2009.
[5] A. De Luca, R. Weiss, and H. Drewes. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In Proceedings of the 19th Australasian Conference on ComputerHuman Interaction: Entertaining User Interfaces, OZCHI '07, pages 199–202. ACM, 2007.
[6] M. Eiband, M. Khamis, E. von Zezschwitz, H. Hussmann, and F. Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems, CHI '17. ACM, 2017.
[7] P. Institute. Global Visual Hacking Experimental Study: Analysis. 2016.
[8] M. Khamis, F. Alt, M. Hassib, E. von Zezschwitz ,R. Hasholzner,and A. Bulling. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '16, pages 2156–2164. ACM, 2016.
[9] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In Proceedings of the 3rd Symposium on Usable Privacy and Security, SOUPS '07, pages 13–19. ACM, 2007.
[10] A. Leon-Garcia and A. Leon-Garcia. Probability, statistics, and random processes for electrical engineering. Pearson/Prentice Hall 3rd ed. Upper Saddle River, NJ, 2008.
[11] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady, volume 10, pages 707–710, 1966.
[12] J. Long. No tech hacking: A guide to social engineering, dumpster diving, and shoulder surfing. Syngress, 2011.
[13] V. Rajanna and T. Hammond. Gawschi: Gaze-augmented, wearable-supplemented computer-human interaction. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ETRA '16, pages 233–236, New York, NY, USA, 2016. ACM.
[14] V. Rajanna, P. Taele, S. Polsley, and T. Hammond. A gaze gesture-based user authentication system to counter shouldersurfing attacks. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '17. ACM, 2017.
[15] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13. ACM, 2013.

[16] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07, pages 159–168. ACM, 2007.

