

A Shoulder-Surfing Resistant Graphical Password Scheme - WIW

Shushuang Man
Department of Mathematics
and Computer Science
Southwest State University
Marshall, MN 56258, U.S.A.

Dawei Hong
Department of Computer Science
Rutgers University-Camden
Camden, NJ 08102, U.S.A.

Manton Matthews
Department of Computer
Science and Engineering
University of South Carolina
Columbia, SC 29208, U.S.A.

Abstract

We propose a new graphical password scheme. It is defined as a challenge-response identification. Hence, a password in our scheme is time-variant. User who knows the password is able to meet the challenge and to respond correctly. As a consequence, our graphical password scheme is shoulder-surfing resistant. An attacker still cannot tell what the password is, even if he/she has filmed a user's login process. Primary experiments on our graphical password scheme showed the scheme is promising.

1 Introduction

1.1 Existing password systems and schemes

Today, password is the most popular way to authenticate a user to login to computer systems. However, we all know that traditional text-based password systems are vulnerable to the shoulder-surfing attack. Through this paper we use the word "shoulder-surfing" in the following sense: A shoulder-surfing attack consists of a user being filmed during his/her login.

To protect customers' passwords, E-commerce vendors adopted various encryption techniques.

Text passwords are encrypted before they were sent across networks. A wire-tapping attacker cannot capture the passwords unless they have enough computing power and advanced decryption techniques. However, with a camcorder aiming at the screen of a computer and its keyboard, traditional text-based passwords will be captured with 100% accuracy.

Blonder [1] proposed a graphical password scheme in which a user is authenticated by clicking a sequence of points on a predetermined image. How secure the proposed scheme is was not discussed. Germyn [2] proposed DAS (draw-a-secret) scheme in which a password is a simple picture drawn on a 2-dimensional grid. The coordinates of the grids in which the picture touched are recorded in temporal order of the drawing. It gives users certain degree of freedom to tolerance their drawing during login process. As long as same cells are crossed with same order, a user is authenticated. Both [1] and [2] are vulnerable to the shoulder surfing. The problem is that every time a user login, he clicks or draws the same sequence of components that make up his password. That is, a password in either [1] or [2] is time-invariant. Thus, once a password has been filmed by an attacker, the attacker can

surely use the password to login. Perrig’s Map Authentication Scheme (cf. [5]) is based on navigation through a virtual world to a site. A user has to remember all passing sites. Therefore, the number of sites cannot be too many. It is hard for an eavesdropper to capture a password by a few times of observations. But any eavesdropper can easily break a password if he can take photos of a user’s login process.

Recently, a shoulder-surfing resistant graphical password scheme was proposed in [6]. The scheme works as follows: It displays h images one by one. In each image there are N distinguishable objects, such as symbols, flowers, animals, etc. Among the N objects there are K so called pass-objects which are pre-chosen by the user, and thus, only recognizable to the user. The h images have different contents. To “pass” an image the user must find the K pass-objects in the image and then make a mouse-click inside of the convex hull of the K pass-objects. To login the user must “pass” all the h images. From login to login, the N objects for each of the h images are randomly placed on a screen of a computer where the user is trying to login. The scheme has two drawbacks:

1. A technical drawback is the following: In order to make any attacker hard to guess the K pass-objects the total number N of objects was set to 1,000 in [6]. We used some best image processing packages to display 1,000 objects of the kinds as mentioned in [6] on a standard 19” screen. As a result, it was impossible to distinguish pass-objects from non pass-objects because they all were too small.

2. There is a theoretical complication. In [6] K is set to 10. It can be proved that (cf. [3]) *There is a constant $c > 1$, which depends only on the size of the screen used such that the probability of the center of the screen being in the convex hull of the K randomly placed pass-objects is greater than $1 - \frac{1}{c^{k-1}}$.* This implies that if the K pass-objects are randomly placed on a screen then an attacker can simply play wait-and-hunt:

For each image he may just click the center of the screen. The probability for him to login is $q = (1 - \frac{1}{c^{k-1}})^h$. For a screen of the standard size we have $c \approx 1.5$, and thus, we have $q \approx 0.77$ when $K = 10$ and $h = 10$; $q \approx 0.45$ when $K = 10$ and $h = 30$. Therefore, the K pass-objects must be moved as a group all over a screen. This complicates analysis of the scheme, since a mouse-click always gives an attacker some hints.

Taking a fundamentally different approach toward objects, we use small number of objects (200 ~ 300) so that we may exploit their structures. Moreover, our scheme does not require any mouse-click, which gives an attacker very little hints. Our idea is quite different from what in [6].

1.2 Our proposed password scheme

Our idea can be described as follows: Let a user choose an “alphabet” for his password. At each time of login our graphical password system randomly spells a “string” from the alphabet. A technical challenge is that it should be easy for the user to identify each of those “strings” and in the mean time, it must be difficult for an attacker to recognize any of those “strings”. How can we meet such a challenge? We propose a graphical password scheme. A password is formed via a few images. One after another, those images are displayed on screen in a fixed order. One image is used for one “letter”. In an image there are many objects among which there are a few so called pass-objects. Pass-objects are pre-chosen by the user as a part of his password. They are recognizable only to the user. A combination of appearances and locations of those pass-objects spells a “letter”. From login to login, in each image the locations of the pass-objects are randomly changed and their appearances are perturbed such that the “letter” spelled varies randomly.

The art of designing such a scheme is to make a user easily identify those “letters” and at the

same time to make any attacker hardly know those “letters” even if the attacker may film the user’s login process. In Section 2 we shall present a full description of our graphical password scheme. Analysis of the scheme is presented in Section 3. In Section 4 we describe an implementation of our graphical password scheme in primary experiments conducted by us.

2 The WIW graphical password scheme

2.1 The components

We call our graphical password scheme WIW, a name we borrow from a well-known puzzle game Where Is Waldo. In WIW for a password we have four parameters, h , n , k and m :

- h is the number of images used for the password. h is determined by user, which may vary from user to user. Different users may have different values for h . Since content of each of the h images changes from one login to another, in sequel we call an image a scene.
- n is the number of objects in each scene. n is fixed ranging from 250 – 300.
- k is the number of pass-objects for a scene. Pass-objects are chosen by a user as a part of his password. And pass-objects are recognizable only to the user.
- m is the number of perturbations on a pass-object.

We explain how to create a password. A user who is creating his password should make the following agreements with WIW: (1) Choose h images as h scenes, (2) in each scene choose k of the n objects in the scene as the pass-objects, and (3) for each pass-object determine m perturbations. Practically, we use h from 3 to 5, n from 200 to 300, k from 4 to 8, and m from 3 to 4.

Note that here we use the word “perturbation”. We mean that appearances of objects, including both pass and non-pass ones, may have small changes from login to login. For example, one pass-object is a kitten, and then perturbations are: The kitten wears a ribbon whose shape and color are variable. The range of this variable is determined by the user who has chosen the kitten as a pass-object. To confuse attackers, WIW generates perturbations on non-pass objects, which are similar to those perturbations on pass-objects.

A screen of a monitor can be viewed as a rectangle with width a and height b . Each scene is displayed on such a screen. For each scene WIW renders two small icons of eye shape at $(\frac{a}{3}, \frac{b}{2})$ and $(\frac{2a}{3}, \frac{b}{2})$, respectively. In the rest of this paper we shall call the two icons left and right eye.

2.2 The protocol

One by one, h scenes are rendered on the screen of a computer where a user is trying to login. Each of the h scenes is a challenge. A challenge-response process may be described using a $(k + 1)$ -vector (a_0, a_1, \dots, a_k) . The scene provides a vector whose all components are blank. And to pass the scene the user must fill out the vector correctly. To login the user must pass all h scenes.

Now, we explain how a user should fill out the $(k + 1)$ -vector (a_0, a_1, \dots, a_k) correctly.

- The first component a_0 represents one of the four cases:
 - C1 Both eyes are outside of the area surrounded by the k pass-objects.
 - C2 Both eyes are inside of the surrounded by the pass-objects.
 - C3 Only the left eye is inside of the area surrounded by the pass-objects.
 - C4 Only the right eye is inside of the area surrounded by the pass-objects.

- The component a_i , $1 \leq i \leq k$, represents one of the m perturbations on the i th pass-object.

The domain for each a_i , $0 \leq i \leq k$, is pre-agreed between the user and WIW. For example, in a trivial case we may take $\{1, 2, 3, 4\}$ as the domain for a_0 where 1, 2, 3, 4 represent C1, C2, C3, C4, respectively; and for a_i , $0 \leq i \leq k$, we take $\{1, \dots, m\}$ as the domain where $a_i = l$ means the i th pass-object wears the l th perturbation, $1 \leq l \leq m$.

At a scene WIW not only randomly allocates the n objects, but also perturbs their appearances. The challenge for the user is: Pick out the k pass-objects, and then according to the positions and appearances to fill out (a_0, a_1, \dots, a_m) . In total we have $4m^k$ vectors. Each vector can be viewed as a “letter”. This means that one scene can be understood as a “letter” that is randomly picked from an alphabet of size $4m^k$. Using h scenes, WIW generates random strings of length h from an alphabet of size $4m^k$. An implementation will be presented in Section 4. We analyze WIW in the next section.

3 Analysis

Suppose that a user has chosen his password. The password has h scenes. In each scene he uses k pass-objects, and for each pass-object he determined m perturbations. How many things he has to memorize for this password? It is totally $h \times k \times m$, which is polynomial in terms of h , k , and m . The trade-off here is that the user gets a time-variant password which is a set of random strings of length h from an alphabet of size $4m^k$. The number of total these strings is $(4m^k)^h$, which is exponential in terms of h , k and m .

3.1 Usability

In our primary experiments we found that for sophisticated users, who are computer profes-

sionals and have good memory, chose h from 3 to 5, k from 4 to 8, m from 3 to 4. This gives an average about $(4 \times 3^6)^4 \approx 52^8$ random strings in total. This shows that memorizing $4 \times 3 \times 4 = 48$ things achieves a set of random strings, which can be regarded as the set of all strings of length eight randomly and case-sensitively chosen from English alphabet. In one words, a sophisticated user’s password in our graphical password scheme is equivalent to using up all passwords of length eight in a traditional text-based password system. Hence, our graphical password scheme meets needs for extremely high security.

In the same primary experiments we required minimum on h , k and m respectively to be 2, 4 and 2. This means that at minimum, in WIW for one password a user needs to memorize two scenes. In each scene there are about 250 objects displayed. The user should be able to pick out 4 pass-objects from the 250 objects. Once the user recognizes the 4 pass-objects, he has no difficulty to distinguish cases C1 - C4. Each pass-object may wear 2 perturbations. The user needs to be able to tell which of the 2 pass-object wears what of the 2 perturbations. Most participants, students from high schools and colleges, had no difficulty to master WIW in which $h = 2$, $k = 4$, and $m = 2$. Our graphical password scheme WIW can be adopted by ordinary people. Notice that at the minimum requirement, for one password WIW uses $(4 \times 2^4)^2 = 64^2$ random strings in total, and the number 64^2 is more than the number of all possible passwords of length 2 in a traditional text-based password system.

A drawback of our graphical password scheme WIW is that it is built in application level in computer system because image processing for each scene. Compared with traditional text-based system this is quite expensive. Though progress of computer technology is cutting down the cost of using our graphical password scheme, the expense gap between using our scheme and traditional one is and will be there. But as shown

in next subsection the ultimate trade-off is that our graphical password scheme is more secure than any traditional ones.

3.2 Security

We consider a shoulder-surfing attack, in which a user's login has been filmed s times by an attacker. In other words, the attacker clearly knows s strings that can possibly be used for the password. Following [3] we analyze two aspects:

- (1) Assuming that the attacker is allowed to try the s strings, what is the chance of his success?
- (2) How much can the attacker learn about the password by analyzing the s strings?

Analysis of (1): To be precise, we introduce a parameter t , the number of times of trials of the s strings. There are totally $(4m^k)^h$ strings in which our graphical password scheme WIW randomly chooses one for each login. Thus, the probability that one of the s strings is chosen is

$$\frac{s}{(4m^k)^h}$$

which is also the probability of the attacker's success in one trial. Hence, the probability of his failure after all t trials is

$$\left(1 - \frac{s}{(4m^k)^h}\right)^t.$$

What does this formula tell us? Since the purpose of using this formula is to demonstrate the security of our graphical password scheme WIW, we take real values for h , k and m . We consider the case where $h = 2$, $k = 4$ and $m = 2$, the minimum requirements by WIW. Plugging in those numbers into the formula above, we have the probability of his failure after all t trials is

$$\left(1 - \frac{s}{64^2}\right)^t.$$

When $(st) \ll 64^2 = 4096$ we have

$$\begin{aligned} \left(1 - \frac{s}{64^2}\right)^t &= \left(\left(1 - \frac{s}{64^2}\right)^{\frac{64^2}{s}}\right)^{\frac{st}{64^2}} \\ &\approx e^{-\frac{st}{64^2}} \approx 1 - \frac{st}{64^2} \end{aligned}$$

which means that the probability of his success is about $st/64^2$. For example, let $s = 2$, i.e., a user's login has been filmed twice. The analysis above reveals the following fact: Suppose that with two strings observed on the films the attacker had chance to try to login 10 times by mimicking the two strings. Then the probability of the attacker's success is only about $20/4096 \approx 0.005$.

Analysis of (2): Since a password in WIW is guided by pass-objects, let us analyze the difficulty for the attacker to know the k pass-objects for one scene. We assume that the attacker knows the number k . As mentioned early, WIW places about 250 objects for one scene. Moreover, our graphical password scheme WIW is designed such that the letter spelled by a scene is determined by the positions and appearances of the k pass-objects for that scene. Thus, the attacker is forced to check out each combination choosing k from 250 objects. Consider the minimum requirements by WIW in which we have $k = 4$. This means that at least the attacker needs to check

$$C(250, 4) = \frac{250 \times 249 \times 248 \times 247}{4 \times 3 \times 2 \times 1} \approx 10^8$$

k -combinations. Furthermore, in the minimum requirements we have $h = 2$. Thus, the attacker needs to check totally 10^{16} pairs of k -combinations at very least.

The analysis given here is rather primary. A fine analysis which is quite involved will be presented in a separate paper. But the primary analysis in this subsection has clearly shown that our graphical password scheme WIW is much more secure than any traditional text-base password systems in terms of resisting shoulder-surfing.

4 An implementation of WIW scheme

Here is a sample WIW system used in our primary experiments. We take $n = 252$ small icons as small objects with size about 0.3×0.45 (square inches). Our graphical password system WIW randomly distributes them into 14 rows and 18 columns on a scene.

A junior high school student was one of the tens of volunteers for our experiments. To respect her privacy we use a nickname Alice for her in this paper. For her password, Alice chose $h = 3$ scenes, $k = 5$ pass-objects for each scene, and $m = 4$ perturbations for each pass-object. To illustrate her choices, we copy the 5 pass-objects that she chose for the first scene, and also copy the 4 perturbations for each pass-object (see Figure 1). The first pass-object is a moon-star, and the 4 perturbations are changes of relative positions between the moon and the star (see the first row of Figure 1). The second pass-object is a crying-eye, and the 4 perturbations are changes of positions of the tear (see the second row of Figure 1). The third pass-object is a fishing pole, and the 4 perturbations are changes of shapes of its string (see the third row of Figure 1). The fourth pass-object is a panda, and the 4 perturbations are changes of holdings in the panda's hands (see the fourth row of Figure 1). The fifth pass-object is a cat, and the 4 perturbations are changes of the cat's step on $+$, $-$, \times , or \div (see the fifth row of Figure 1).

In our experiments, WIW randomly scrambled 252 icons including the 5 pass-objects for each login, which generates four possible cases of positions of the 5 pass-objects. Figure 2, 3 and 4 respectively illustrate case C3, C2 and C1 (mentioned in Subsection 2.2). To respond the challenge by this scene Alice must fill out a vector $(a_0, a_1, a_2, a_3, a_4, a_5)$. Symbolically, we specify the domains as follows (cf. Figure 1, 2, 3 and 4):

Domain for a_0 : $\{c_1, c_2, c_3, c_4\}$.

Domain for a_i : $\{p_{i1}, p_{i2}, p_{i3}, p_{i4}\}$, $i = 1, 2, 3, 4$. In Alice's case for simplicity we let her use the trivial coding:

$$c_i = i, i = 1, 2, 3, 4;$$

$$p_{ij} = j, i = 1, 2, 3, 4, 5 \text{ and } j = 1, 2, 3, 4.$$

We refer reader to Figure 2, 3 and 4 to see how those values should be used.

We scored the performance of each volunteer who participated in our experiments. Alice's score was about the average. It was interesting for us to observe that Alice is able to pass this scene with 100% accuracy (total trials were 32).

All figures referenced in this subsection are in the next page.

Acknowledgement

We thank all people who participated in our primary experiments. Their comments turned out to be very helpful for this paper.

References

- [1] G. Blonder, *Graphical passwords*, United States Patent 5559961, 1996.
- [2] I. Germyn, A. Mayer, F. Monroe and M. Reiter, The design and analysis of graphical passwords, *Proceedings of the 8th USENIX security symposium*, 1999.
- [3] D. Hong, J-C. Birget and S. Man. The probability of a given point in a random convex hull. preprint.
- [4] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- [5] L.D. Paulson, Taking a graphical approach to the password, *Computer* 35 No.7 19-19, IEEE Computer Society. 2002
- [6] L. Sobrado and J-C. Birget, Graphical passwords, *The Rutgers Scholar, An Electronic Bulletin for Undergraduate Research*, Vol. 4, 2002.