

A Password Scheme Strongly Resistant to Spyware

Dawei Hong*
Dept. of Computer Science
Rutgers University-Camden
Camden, NJ 08054
Email: dhong@camden.rutgers.edu
Telephone: (856) 225-6699
Fax: (856) 225-6624

Shushuang Man[†] Barbra Hawes
Dept. of Math./Computer Science
Southwest Minnesota State University
Marshall, MN 56258
Email: {mans, hawes}@southwestmsu.edu
Telephone: (507) 537-6168
Fax: (507) 537-6151

Manton Matthews
Dept. of Computer Science/Engineering
University of South Carolina
Columbia, SC 29208
Email: matthews@cse.sc.edu
Telephone: (803) 777-3285
fax: (803) 777-3767

Abstract

Spyware is now serious threat to computer security. In particular, the Internet is used to remotely login to servers on which sensitive data and applications are stored. Spyware may well steal passwords for login to such a server when users login to the server through the Internet. We propose a new password scheme which is strongly resistant to spyware. In theory, a password in the scheme is a set of random strings. We programed the scheme and conducted experiment. The result is promising.

1 Introduction

A variety of spyware has become serious threats to computer security. In general, spyware is any technology that aids gathering information about users and their computer systems

*DH supported by NSF grant CCR 0310793

[†]SM supported by NSF grant CCR 0310571

without their knowledge. In particular, on the Internet, spyware is program installed in a user's computer (without the user's knowledge) to secretly gather information and relay it to other parties. There are a number of ways for spyware to get in a computer system, for example, as a software virus or as the result of installing a new application.

One of the attacks spyware in a computer may launch is to steal users' passwords, who use the computer to login. There are many such cases have been reported. One can have counter-spyware installed in a computer. Running counter-spyware requires much resources. Moreover, newly developed spyware may overcome the counter-spyware being used. We would naturally ask: *Is it possible to have a password scheme that is strongly resistant to spyware?*

To answer this question we need to technically define what spyware can do in order to steal passwords. Different spyware may function differently. We take the maximum into account: Spyware installed in a computer can detect and record a user's every move and its response by the system. For example, when a user strikes a key, spyware knows which key it is; when a user clicks the mouse, spyware knows what the scene being displayed on the screen is and the location where the click being made. We call a password scheme strongly resistant to spyware if it can overcome spyware described above.

The traditional text-based password scheme is totally vulnerable to spyware. Here, "text-based" we mean that a password associated with a user is a string of characters. This string, as a password, is a shared secret between the user and system. To gain access to a system resource, using a keyboard the user types a user-ID (which is typically a string too) paired with the password. A user-ID is a claim of identity and a password is the evidence supporting the claim. This password scheme has been used for about four decades. There have been much efforts to make this scheme secure (cf. [6]).

Recently, efforts have been made to have graphical password schemes more secure than the the traditional text-based one. Blonder [1] proposed a graphical password scheme in which a password is a sequence of mouse-clicks at points in a predetermined image. Jermyn *et al* [4] proposed DAS (Draw-A-Secret) scheme in which a password is a 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. As long as same cells are crossed with same order, a user is authenticated. Let-Me-In is a graphical password interface by Microsoft, which works in a similar fashion to DAS [7]. The Map Authentication scheme [2] is based on navigation through a virtual world consisting of several sites. A user memorizes all sites. "Passfaces for Windows" is a commercial product by the Real User Corporation [8]. It adds one more layer to the traditional text-based password. To complete a login, a user first must correctly type in his user-ID and password (as in the traditional text-based password scheme), then in a list of faces displayed on the screen the user must correctly click on some of the faces (which are usually user's friends' faces). All schemes above are vulnerable to spyware that can detect and record a user's every move and its response by the system.

What are weak spots in existing graphical password schemes? They are mouse-clicks and contents of images used. Since the first graphical password scheme was proposed, mouse-clicks have been always used as the mean of entering a password. A mouse-click usually indicates key components in this password. For example, in "Passfaces for Windows", when

a user clicks some faces he clearly shows that which are pass-faces. This makes the scheme vulnerable not only to spyware but also to shoulder-surfing attack.¹ Though the content of an image (used for a password) varies from time to time, the components in the content which are used as the password are fixed. For example, in “Passfaces for Windows”, the list of faces is randomly scrambled at each login, but pass-faces are fixed (predetermined by the user). Once spyware detected and recorded which pass-faces are, the password is cracked.

To overcome those weak spots in existing graphical password schemes, we propose a new graphical password scheme. Our idea is:

- (I1) Not only let the content of an image (used for a password) vary, but also let each component in the content (which is used as a part of the password) vary.
- (I2) Make the way of a user’s entering his password not directly tightened to an image (used for the password).

In one words, our scheme is designed as a challenge-response system that can confuse spyware. This paper is organized as follows. In the next section we present the scheme and analyze its security. Then in the section after that we demonstrate our experiment of the scheme.

2 The scheme and its security

Formally, in our scheme a set of random strings is used as a password. A part of this idea appeared in one of our early work [5]. In this paper we much further develop the idea in a practical way. Also, we programed this new scheme and run experiment. The result is promissing. In this section we first present a detailed description of our scheme, and then analyze its security. In next section we shall describe our experiment.

2.1 Description of the scheme

At each time of login, the system challenges a user who wants to login. The challenge is an image displayed on the screen. We shall call this image a login screen. A login screen is divided into n grid. Each grid contains an icon. An icon may be a symbol or a small figure. Among the n icons on a login screen, there are k icons pre-chosen by the user who owns the password. We shall call the k pass-icons, and the other $(n - k)$ non pass-icons. Each of the n icons has m variations which are designed easy for human to recognize and hard for computer vision. (We suppose that spyware has computer vision.) In particular, the m variations for each of the k pass-icons are a part of the shared secret between the system and the user (owner of the password). In Figure 1 we list ten icons each with four variations.

For each of the k variations of each pass-icon, the user (pre-)defines a string S_{ij} , $i = 1, \dots, k$, $j = 1, \dots, m$. There are total $k \times m$ strings. We shall discuss how to help user to

¹A shoulder-surfing attack consists of a user’s login process being watched or even taken photos.

define those strings so that he/she can use icons as hints to easily remember them in next section when we present our experiment.

At each time of login, the system randomly places the n icons (pass and non pass ones) into the n grid, and randomly assign each icon with one of its m variations. Then the system displays all n icons as a login screen. Figure 7 is a sample login screen used in our experiment. To login the user must correctly respond to the login screen as follows: He/she should apply two steps. Step 1 is to recognize the k pass-icons and their variations. Step 2 is to enter strings $S_{1,j_1}, S_{2,j_2}, \dots, S_{k,j_k}$ (in order), if the i th icon is exhibiting its j_i th variation. Here, the concatenated string $S_{1,j_1} S_{2,j_2} \dots S_{k,j_k}$ is used as the password for the user's login at that time. We call such a concatenated string a pass-string. Thus, depending on which variation of each pass-icon is exhibiting, for k pass-icons each with m variations, we have total m^k pass-strings can be used as the password at different times.

Is it possible for a user to handle this kind of password? As matter of fact, this was the question we asked ourselves. A password scheme, no matter how strong it is in theory, must be adaptable for user and system. We carefully chose $n = 121$, $k = 4$ and $m = 4$. Then we conducted experiment on ourselves. We were amazed how quickly we could handle the scheme. We shall describe the experiment in Section 3. We strongly feel that to analyze why people (at least us and our friends) are able to handle the scheme in such a fast and smooth way is an interesting topic in human-computer interaction. We planed research along this line and shall carry it out soon.

2.2 Analysis of the security

Since our experiment showed that $n = 121$, $k = 4$ and $m = 4$ are adaptable choices. We shall analyze the security based upon these numbers. In this case a set of $m^k = 4^4 = 256$ random pass-strings is used as a password.

In theory (cf. [6]), when analyzing security we should suppose that adversaries know the scheme. Now, we assume that spyware functions well so that it knows all 256 pass-strings for the password and relay all these strings to a third party. It is still hard for the third party to use these strings to login. The system randomly places the 121 icons on the screen, which results in that with equal probability ($\frac{1}{256}$) one of the 256 pass-strings is used. The third party may randomly choose a pass-string. Since the system randomly chooses a pass-string, the probability for the two strings match is only $\frac{1}{256} \times \frac{1}{256} =$. Suppose that the system allows anyone to try three times for one login. Then the probability for the third party to login is only

$$1 - \left(1 - \frac{1}{256^2}\right)^3 \approx 4.6 \times 10^{-5}$$

The third party can try another login, but he/she will have only the same extremely small chance.

Spyware can try to figure out what the four pass-icons are. There are total 121 icons (pass and non pass ones). Spyware has only computer vision, and hence, it will be greatly confused by variations of the icons. Despite this difficulty, at very least spyware has to

correctly pick 4 from 121 icons, which yields choices as many as

$$\binom{121}{4} = 8,495,410$$

Each of these choices involves 4 icons (as small images). Thus, to enumerate all these choices is not easy for any computer program.

3 Our experiment

To design our experiment we suppose that the password system is installed on a server. The server is highly secure, and hence, is spyware free. Passwords are stored on the server using standard techniques (cf. [6]). Users login to the server from network (e.g. the Internet). The security concern is that over network some user-end computers may have spyware installed. When a user uses one of these computers to login the server, he may lose his password to spyware. An important remark is that we need to assume that creation and change of passwords are secure, that is, during creating or changing a password the computer used is spyware free. One way to guarantee this is to have creation and change of passwords carried out on specified computers which are surely spyware free.

We programed the scheme using Visual Studio.Net 2003. Every time when a user tries to login to the server, our password system deploys a login screen on the user-end computer. A login screen is divided into 121 grid, 11 rows and 11 columns.

When a new user creates his password, he chooses all 121 icons from an icon library on the server. Also, he determines 4 pass-icons. Each icon has 4 variations. The user does not need pay attention to variations of non pass-icons, but the password system will lead the user going through the 4 pass-icons to set up his password. Let us use an example to show how this procedure is done. A user (one of the authors) chooses four icons, Face, Pin, Chain and Cube, as pass-icons. Figure 2 shows the four variations. The user (she) determines a string that corresponds to a variation and enter the string beneath the variation. In Figure 2, the way she chooses the strings is to relate the variation to some events in her life. The first variation is a happy face and she entered “99dc”. In 1999 she was granted her doctorate degree. For the second variation, a puzzled face, she entered “93gre”. She took GRE exam in 1993. The test on vocabulary was tough. The third variation is a face wearing a pair of glasses and she entered “80sd”. In 1980, she went to Shandong university and it was the time when started wearing glasses. The fourth variation is a sad face with two drops of tears. She entered “3fvr”. She was seriously ill when she was three. Strings for variations of the other three icons are set in a similar manner (see Figure 3, 4 and 5).

Once the password is created, the system deploys a summary as shown in Figure 6, which can be printed out for case where the user cannot remember the password. Now, the user can use this password to login to the server through any connected computer. Even in case where the computer has spyware, that spyware will have very hard time to steal the password. Figure 7 and 8 show two login screens for the same user at two different times.

All of us tried the password system. In average, it took one person fifteen minutes from creating a password to using it fluently. The trade-off is to have the password strongly

resistant to spyware. *A server, which has sensitive data and applications, this type of security is necessary.*



Figure 1: Samples of variations. There are eight icons each with 4 variations.

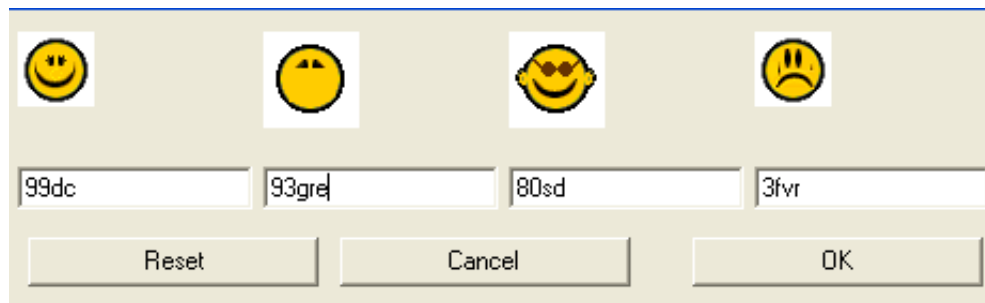


Figure 2: Face icon with 4 variations



Figure 3: Pin icon with 4 variations

References

- [1] G. Blonder, Graphical passwords, United States Patent 5559961, 1996.

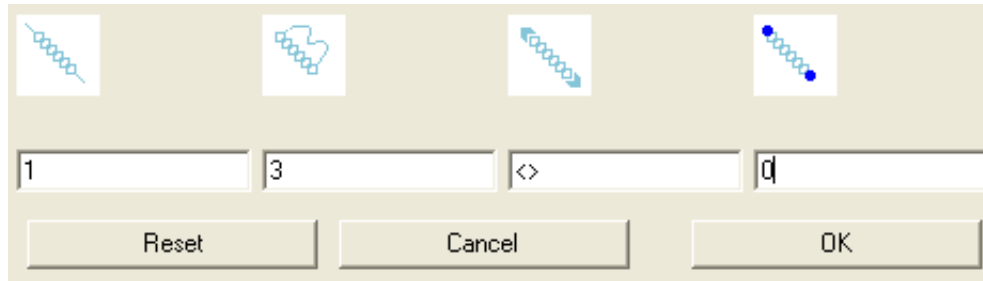


Figure 4: Chain icon with 4 variations

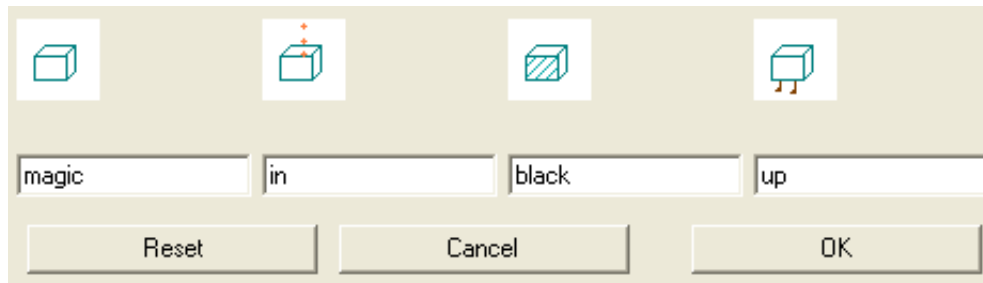


Figure 5: Cube icon with 4 variations

- [2] R. Dhamija and A. Perrig. Déjà Vu: a user study using images for authentication, *Proceedings of the 9th Usenix Security Symposium*, August 2000.
- [3] I. Germyn, A. Mayer, F. Monrose and M. Reiter, The design and analysis of graphical passwords, *Proceedings of the 8th USENIX security symposium*, 1999.
- [4] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter and A. D. Rubin. The design and analysis of graphical passwords. *Proceedings of the 8th USENIX security symposium*, 1999.
- [5] S. Man, D. Hong and M. Mathews, A shoulder-surfing resistant graphical password scheme, *Proceedings of 2003's international conference on security and management*, Vol. I pp. 105-111, Las Vegas, June 2003.
- [6] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- [7] Microsoft Corporation, Let Me In: Pocket PC user interface password redirect sample, July 2003. <http://support.microsoft.com/default.aspx?scid=kb;en-us;314989>
- [8] Real User Corporation. 2003. <http://www.realuser.com>

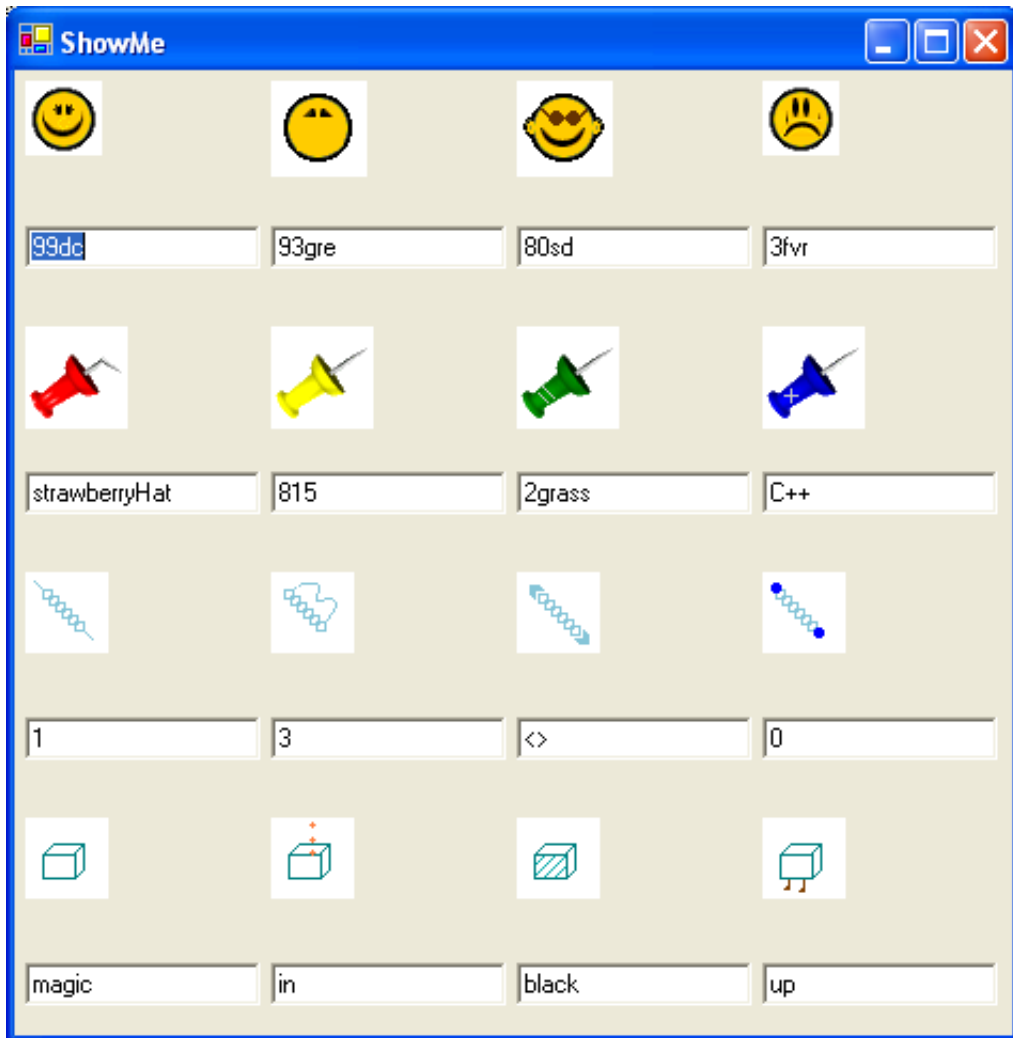


Figure 6: Show Me My Passwords



Figure 7: Login Screen 1. The pass-string is: 99dc8151up

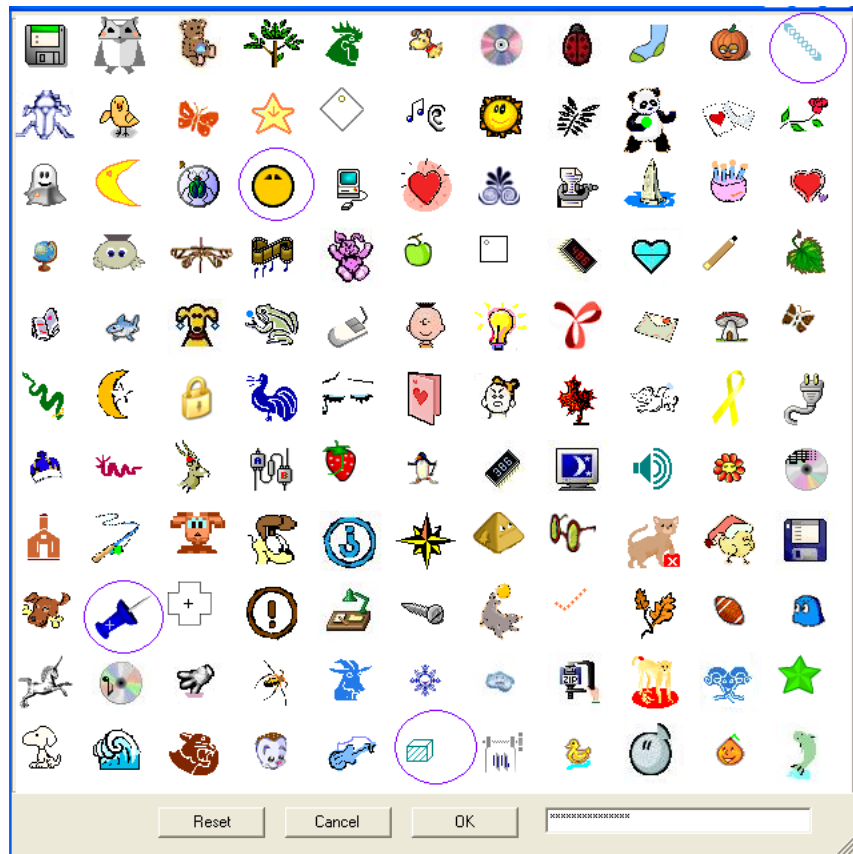


Figure 8: Login Screen 2. The pass-string is: 93greC++<>black