Screen oriented technique for reducing the incidence of shoulder surfing

Bogdan Hoanca
Computer Information Systems
University of Alaska Anchorage
3211 Providence Drive
Anchorage, AK 99508
Phone: (907) 786-4140
Fax: (907) 786-4115
afbh@uaa.alaska.edu

Kenrick Mock
Computer Science
University of Alaska Anchorage
3211 Providence Drive
Anchorage, AK 99508
Phone: (907) 786-1956
Fax: (907) 786-6162
afkjm@uaa.alaska.edu

Abstract – We propose an intuitive graphical password entry scheme that significantly reduces the ability of an attacker to obtain passwords by observing authorized users' authentication sessions. The scheme is resistant to dictionary attacks, and is likely to employ stronger and much more memorable passwords than conventional text or graphical schemes. We propose a procedure that makes the scheme resistant to guessing attacks, more resistant than other graphical schemes in the literature. Finally the scheme is designed to minimize the search time the authorized user needs to find the relevant information on a screen with multiple graphical objects. The scheme is currently being implemented and will be tested for usability.

Keywords: Shoulder surfing, graphical passwords, information security

1. Introduction

"Shoulder surfing" or "peeping attacks" refers to stealing information (especially authentication information) by looking over the shoulder of an unsuspecting user. In a general sense, shoulder surfing involves the unauthorized observing of an authorized user's session on an electronic device in order to gain access to information.

Historically, shoulder surfing concerns moved from telephone calling card fraud to automated teller machine (ATM) fraud, and more recently to mobile computer users. Low tech shoulder surfers are reported to sit in a car at a safe distance and use binoculars, telephoto lenses or more recently camera phones to record users entering their personal identification numbers (PINs).

The proliferation of laptop and personal digital assistant (PDA) usage, especially on now ubiquitous wireless networks, has greatly increased the danger of unauthorized observation of authentication procedures and even the theft of password information. Mobile computing users in a public place cannot be aware of all the activity in their surroundings, and are vulnerable to persons visually observing or even recording their authentication session, with the intent of extracting login information. Similar considerations apply to access points (security doors) where the user is expected to authenticate by entering a PIN. Most authentication methods involve pressing keys on a keyboard or selecting objects on a screen, and both the screen and the keyboard are visible to the authorized user as well as to the shoulder surfer. This paper will present and analyze the performance of a graphical screen oriented password entry system that greatly reduces the threat of shoulder surfing.

2. Related work on shoulder surfing

Most often computer security relies on using authentication credentials that the user i) is supposed to remember (password) ii) is born with (a biometric trait) or iii) was given as a physical authentication token (magnetic card, chip or other physical device). Often several of these techniques are combined to increase the security of a system.

Biometric identification techniques such as face scanners and particularly fingerprint readers are gaining in popularity, but some of these methods are still prone to false positive and false negative identification [1]. Some conditions, such as a cut finger, may result in further problems with biometric authentication. Privacy

groups have also expressed concerns over the potential theft or the gathering and sharing of personal biometric data; unlike other types of authentication, a stolen biometric is not easily replaced.

Authentication methods based on physical access cards are vulnerable to theft or loss. Unless the physical card is combined with other authentication means, a person who steals or finds a lost card would have full unrestricted access to the protected information.

Using passwords that the user must remember is not perfect either, but is widely used because of the low cost and the relative ease of use. Unlike biometrics or physical keys, passwords do not require special hardware (which reduces costs) and do not require any skills beyond those needed to use the computer itself (typing and mouse handling).

For all their advantages, passwords have several drawbacks. Much research has assessed the strength of various password schemes, starting with the early text based UNIX password [2] and more recently focusing on graphical passwords [3]. Passwords are likely to be either too simplistic (and easy to guess) [4,5] or if not simplistic they are difficult to remember and are likely to be written down. Passwords are also the most likely to be stolen without the owner's knowledge by a malevolent observer who watches or records the user entering a password. The danger of this information leak is compounded in the case of authentication from crowded public areas, where multiple observers may be watching or even recording several access sessions, then using this information to recover passwords.

Shoulder surfing attacks as a security threat to computer users have been discussed in the literature for more than a decade. Many authors mention shoulder surfing as a threat, especially for PDA's [6,5] but dismiss it by assuming small screens and limited viewing angles for mobile devices. Although true at the time, these assumptions are less valid today.

Probably the most relevant area of concern for shoulder surfing attacks is that of mobile computing. The combination of public places, mobile computers and wireless networks makes for a fertile ground for shoulder surfers to look for victims. Due to the potentially large number of people in the immediate vicinity of a wireless user, it would be difficult to assess which of the people are potential shoulder surfers and which ones are harmless bystanders.

Shoulder surfing attacks on a laptop user connecting to a corporate network from a public place could disclose not just user names and passwords to the attacker but also phone numbers, log on procedures, location of important documents on the network, and even the content of documents.

While the issue of shoulder surfing during a session remains open (where the attacker would be able to read confidential information on the screen), several solutions have been proposed to address at least the authentication phase. Hopper and Blum [7] identified the need to devise protocols that would even be useful for a user that is naked and under constant surveillance by malicious observers (extreme situation to make a point), when most of the existing authentication means would divulge the password or not allow the user to log on.

The class of measures that have been proposed for authentication under these conditions is known under several possible names: shared secrets, challenge-response protocols and zero-knowledge protocols. For all these protocols, the idea is to authenticate the user by questions that show that the user knows a secret, but where the answers do not explicitly reveal the secret. Several schemes have been proposed that require modest mathematical calculations, such as computing the modulus or parity of numbers [8].

For a simpler and more intuitive interface, Sobrado and Birget [9] present several graphical schemes that are resistant to shoulder surfing. The idea is to use a challenge response mechanism that authenticates the user without revealing the password to an observer. Sobrado and Birget propose three different schemes that can achieve this, all of them applied to graphical password spaces. The user is presented with a screen full of hundreds or more graphical objects, randomly distributed each time they are displayed, and must identify the objects that compose the password. To avoid disclosing the password information to an observer, the user does not select the password objects themselves, but instead must click anywhere within the convex hull of the triangle formed by a triplet of objects. Because the number of possible triangles far exceeds the number of objects on the screen, the observer has obtained very little, if any,

useful information about the actual password just by knowing the clicked location. More recently, Sobrado and Birget realized that the center of the screen has a high likelihood of being within the convex hull, which leads to a high risk of false positive, so they propose methods to deal with this threat [10]. The other two challenge response ideas presented in Sobrado and Birget's work include clicking at the intersection point of the diagonals of a set of four points, or moving a frame to align one object on a mobile frame with two others that are fixed on the screen. This last method reportedly has the advantage that it discloses even less information, since the user does not need to click anywhere on the screen.

Several challenges are apparent in Sobrado and Birget's scheme. First, if the number of graphical objects is large, the user will spend a long time searching for the objects that compose the password. Objects are even more difficult to locate because the distribution must be randomly changed every time the user authenticates in order to change the locations the user clicks on successive logins. A secure password requires either more objects on the screen or a longer password, either scenario resulting in a higher search cost for the user to find the appropriate objects on the screen. An eight screen logon would require locating three times this many objects, potentially a lengthy process.

As a second drawback of the scheme, if the randomly distributed objects that make up a given password screen are all clustered in a corner of the screen, the convex hull of the triangle will be small and relatively close to the corner of the screen. Additionally, the number of possible triplets associated with that clicked location may be small, which would make it easier to guess the triplet. We refer to this problem as edge and corner effect. Sobrado and Birget discuss ways to address this using a relatively complex algorithm.

In this paper we propose to modify the Sobrado and Birget scheme to address both of these problems. First, we require the user to click in the center of the triangle, rather than anywhere in the convex hull. This greatly reduces the likelihood of false positives, although it requires the system to account for user error.

Furthermore, in the next section we propose a scheme that allows the user to quickly and efficiently find objects on the screen, and we also show how the procedure could be modified to avoid edge and corner effects.

3. Scheme to reduce the risk for shoulder surfing

To allow the user to quickly locate the password symbols on the screen, we propose to use an alphanumeric pass phrase as the password, for example a sentence including letters, digits, and punctuation marks. The available set of characters is printed on the screen, and the user must select groups of three symbols from the pass phrase by clicking in the center of the associated triangles. For example, if the password is "I carry $23.5 with me, but not today," the user would have to click in the center of the triangles formed with alphanumeric characters /I c/arr/y $/23./… where we have used the forward slash "/" to separate triplets of points. A special nonblank character would have to be used for space (for example the string "sp"). For a sentence of N characters (including spaces and punctuation marks), the user would have to click through $N/3$ screens to authenticate.

The key to being able to quickly find the characters of interest on the screen is to always keep the alphanumeric characters in lexicographic order. At the same time, we need to maintain an element of randomness, so that each login would require the user to click in a different area of the screen. To achieve this, we randomly vary the direction and the starting point for the string of characters, while maintaining the lexicographic order. The user will be able to quickly understand the spatial distribution of the characters, and find the characters required for the password, but the center of a triangle formed with any triplet of characters would be in a very different location depending on the particular type of spatial arrangement randomly selected for that particular authentication session.

As shown in Fig. 1 for the simple case of letters only on a screen with 4x4 positions, the direction of the text could be vertical, horizontal, alternating horizontal left and right (or up and down), spiraling in (or out), diagonally (same direction or alternating), or other more complex patterns. An additional degree of freedom with all of the schemes in Fig. 1 is that the starting point (the letter "a") could be anywhere on the screen.

The space of possibilities would have to be designed to ensure that shift-equivalent patterns (those that differ only in a uniform shift) would not be used. For example, for the horizontal run

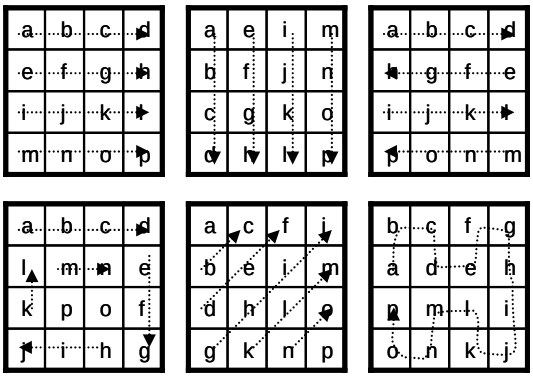

Figure 1. Possible arrangements for sixteen characters on a 4x4 screen to randomize the overall distribution while preserving the lexicographic order

pattern (the pattern shown in the upper left corner in Fig. 1), all patterns with the letter "a" in the leftmost column differ from each other in just a simple shift (with the lines wrapping around the top and bottom of the screen). A shoulder surfer that sees the user click for example on the "g" will have a high likelihood of success if she clicks on the "g" in an attempt to log on. Except for a wraparound effect, many of the possible triplets that have their center on "g" will still be centered on that character if the pattern is shifted up or down. The solution to this is to only use one of the patterns in the set that are shift-equivalent (for example, only the pattern with "a" in the upper left corner would be used among the patterns with "a" in the left column).

Since most of the pattern types displayed here are not prone to shift invariance, they will generate a number of possible configurations of the order of the number of characters on the screen. For example, the spiral pattern (lower left corner in Fig. 1) could generate sixteen different (not shift-related) patterns, depending on which of the letters is in the upper left corner of the grid. In general, this spiral pattern would generate N^2 possible patterns on an NxN screen, one for each possible position for the letter "a."

By using upper and lower case characters, digits, punctuation marks and spaces, the complexity of the password could approach that of a well designed regular password. With 26 characters (both upper and lower case), punctuation marks (including parenthesis), digits, space, the number of possible choices would be approximately 70 per screen. This in fact amounts to a cryptographic complexity of 70^8 (for an eight screen password), which is equivalent to 49 bits, not overly impressive, but certainly more than the expected complexity of a typical user password [2].

The second problem we identified with the Sobrado and Birget scheme is that of corner and edge effects. Figure 2 shows a plot of the number of possible triangles that have the center located at a given position on the screen for a 20x20 array of symbols. It is apparent that most triangles have their center close to the center of the screen. These points are robust to shoulder surfing attacks, because the attacker must guess which triplet out of a *large number* of possibilities corresponds to the user password. On the other hand, if the chosen point lies close to an

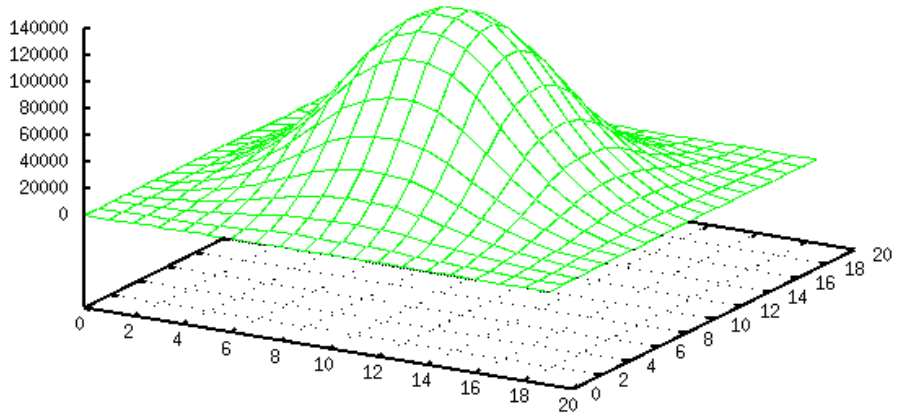

Figure 2. Mesh plot of the number of possible triplets of screen points defining triangles with center at a particular location on a 20x20 screen. Most triangles are centered somewhere close to the center of the screen. The points in the corners and close to the edges of the screen correspond to a much smaller number of possible triplets.

edge or a corner of the screen, the number of possible choices for the vertices of the triangle is greatly reduced. A shoulder surfer would have an easier task in determining which triangle the user was targeting, because the number of possible triplets determining the triangle would be relatively small. The distribution of occurrences of centers assumes that users click in the exact geometric center of the triangle. This paper does not consider the implications of the positioning errors between the perceived and the actual center. These errors would as a first order act as a noise on the plot in Fig. 2, and would have an averaging effect that would further flatten this curve, making shoulder surfing more difficult.

We propose to use a border of repeated tiles around the main *N*x*N* screen area. Within the main *N*x*N* area, the characters are arranged as described before, in lexicographic order and in a new and random pattern for each login session. The pattern area is surrounded by a one tile wide border, where the character pattern repeats such that each character appears periodically in both horizontal and vertical directions with a period of *N*. The idea now is to use points that are as far away from each other as possible when defining the triangle associated with a triplet of points. To avoid forcing the user to decide which triangles are the largest, we suggest using the following algorithm to select the triplet with the largest associated triangle.

The algorithm includes two phases. In the first phase, the computer grays out some of the characters displayed on the screen, to give the user guidance in how to choose appropriate triplets. In the second phase, the user is instructed to choose a combination of black and gray characters, which will lead to choosing the triangles with the largest areas.

The *computer algorithm* is as follows:

1. *Generate the center area (the area with gray background in Fig. 3) and the one tile border around it, based on the pattern type and the initial starting position (where the letter "a" in Fig. 1 would be located).*
2. *Print in black all the characters in the center area and in gray all the characters on the border.*
3. *For each tile that is on the edge of the center pattern but not in a corner, print in black the corresponding tile with the same character that is on the outer border (there should be exactly one such occurrence for each character on the edge).*
4. *For each tile that is in a corner of the center area, print in black the tile that is in the corner diagonally across the screen on the border area.*

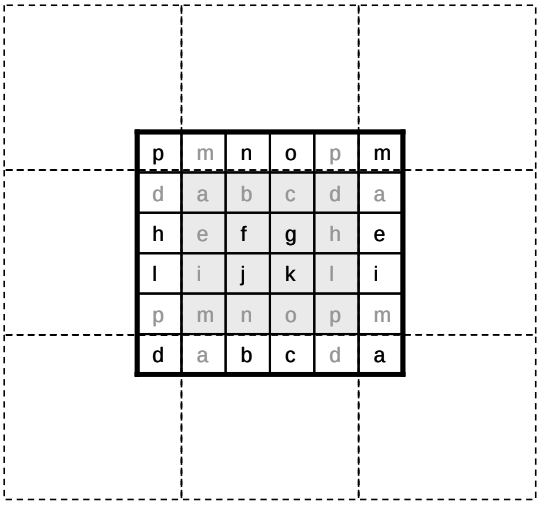

Figure 3. Tiled distribution of one of the random patterns showing how adjacent tiles create a border around the initial configuration. Dotted pattern areas each contain the same pattern as the grayed out area in the center. The screen displays the central pattern area, and a one tile wide border around it, making characters occur periodically at multiple locations around the edge.

This same procedure would apply to display the border and center area, regardless of how the center area characters are arranged (horizontally, vertically, in spiral etc). For each triplet of characters in the password, the *user* is instructed to select the triangle corresponding to the triplet such that

1. *Choose all black characters in the triplet*
2. *If all characters are on the same edge of the center area, replace one of them with a grey character*

This way, the user is forced to choose triangles of the largest area, ensuring that the center of those triangles is close to the center of the screen. We show in Fig. 4 the effect of this one tile border on the number of possible triplets that are associated with each position on the screen. In particular, the corner points on the screen without border correspond to only one possible triplet, but when using the one tile

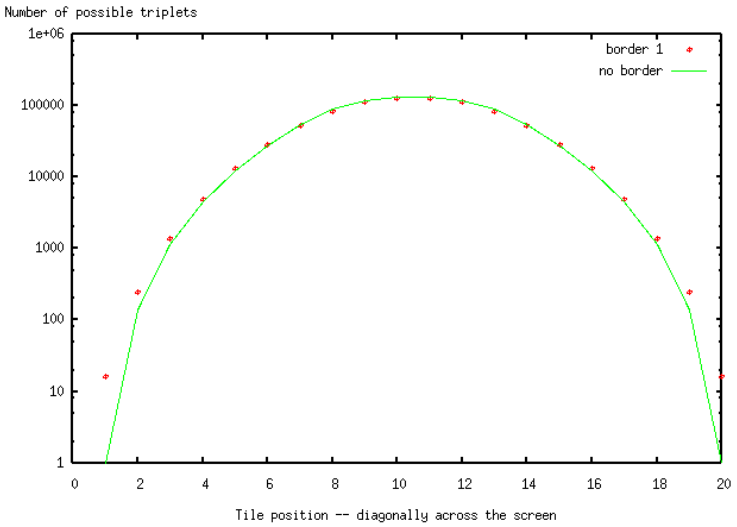

Figure 4. Cross section (diagonal) through the mesh plot for Fig. 2 (solid line, labeled "no border"), compared with a cross section (diagonal) plot of the same mesh when a one tile border is applied around the initial screen area (dots, labeled "border 1"). The corner positions for the "no border" are generated by exactly one possible triplet, while corners for the "border 1" case correspond to any one out of sixteen possible triplets.

border, they could be any one of sixteen possible triplets.

The discussion above has focused on the worst case scenario from the point of view of a shoulder surfer attack that is attempting to guess the user password based on the triangle center the user is entering. A more global view of the performance of the scheme is to look at the informational entropy of the distribution of screen locations. In the absence of edge and corner effects, all screen areas would be equally probable as a position of the center of the triangle. In that case, the entropy (ideal case) for an *N*x*N* screen would be

$$H = -\sum_{all\_character\_locations\_i} p_i \cdot \log_2(p_i),$$

where p_i are probabilities of occurrence of the center of the triangle at locations on the screen, $p_i = \frac{1}{N^2}$. We calculate the actual entropy including edge and corner effects by exhaustively evaluating all possible triplets of points. The comparative results of the ideal case, the case with edge and corner effects (without border) and the case with a one tile border are shown in Fig. 5. The situation when no border is used leads to a loss of about one bit of complexity, relatively uniform across all screen sizes from 3 to 20. Using a border reduces this loss somewhat, but the reduction is higher for smaller screen sizes. For screen sizes exceeding 15x15, the improvement when using a border is less than 0.1 bits. Based on the earlier discussion of the local edge and corner effects, using a border is still beneficial in making the distribution of triangle centers more uniform, even though the entropy is almost unaffected.

If needed, the border could be expanded to include two or more tiles all around the center area, which would further reduce edge and corner effects. This could eventually lead to equal probability occurrences (what we described above as the ideal case), but at the price of a much more cluttered screen and a much more complex algorithm for the user to

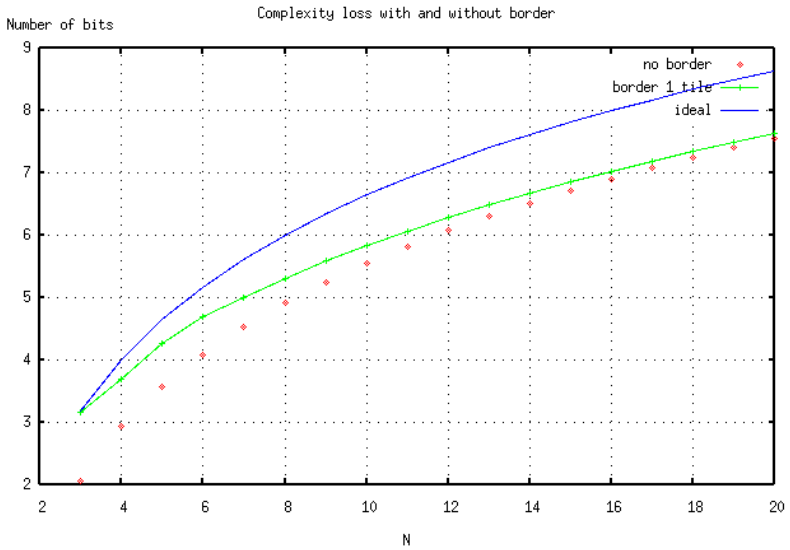

Figure 5. Complexity loss (number of equivalent bits), relative to the ideal case for the case with and without border effects. The dotted curve shows the case of a screen with *N*x*N* tiles and no border. The dotted + solid curve shows the case for a one tile border around the *N*x*N* screen.

choose optimum triangles. For this reason we limit this idea to borders one tile wide only.

As an added benefit of the proposed scheme, clicking in the center of triangles defined by triplets of characters results in averaging of the natural frequencies of letters in the English alphabet. If a shoulder surfer were to use any heuristics to try to determine the relative likelihood for triangle centers, she would face the daunting task of needing to combine information about the natural frequencies of individual characters in the triplet. This would most likely not lead to any information gain.

Following this analysis, the expected strength of the proposed graphical password scheme would reduce to approximately 40 bits instead of 49 bits when using upper and lower case letters, digits and punctuation marks with eight validation screens and including edge effects. This may seem low, but considering that a 24 character phrase is more memorable than a strong (but cryptic) password including eight characters, the users are less likely to write the down their password, and also less likely to have to call customer service to reset their password.

4. Conclusions and further work

This paper introduced and analyzed a scheme that is resistant to shoulder surfing, and that expands and improves on the work of Sobrado and Birget [9]. Our proposed scheme allows users to find the password quickly on a screen with a relatively large number of symbols, and is also robust to edge and corner effects, which would make it applicable to small keypad security systems (for example ATM's and secure doors).

As novel authentication technologies improve in performance, decrease in price, and gain more acceptance in the user base, the threat of shoulder surfing will decrease. In the meantime, as people become aware of the dangers of shoulder surfing, the threat will reduce. Simply by paying more attention to their surroundings and by using caution to cover the keypad or other input device, users could greatly reduce the risk of their information being stolen by a shoulder surfer. At the same time, shoulder surfers are expected to become savvier and to make use of increasingly more sophisticated technologies as they become available in the future. The schemes proposed in this paper should be useful to design systems that thwart shoulder surfing attacks in the near future.

5. References

[1] S. Tseng, "Comparison of Holistic and Feature Based Approaches to Face Recognition," Masters of Applied Science in Information Technology thesis, School of Computer Science and Information Technology, Faculty of Applied Science, Royal Melbourne Institute of Technology University, Melbourne, Victoria, Australia, July 2003

[2] R. Morris and K. Thompson, "Password Security: A Case History", Communications of the ACM, vol. 22(11), pp. 594—597, 1979

[3] G. Blonder, "Graphical Passwords". United States Patent 5559961 (1996).

[4] J. Thorpe and P. van Oorschot, "Graphical Dictionaries and the Memorable Space of Graphical Passwords," in *Proceedings of the 13th USENIX Security Symposium*, San Diego, USA, August 9-13, 2004, pp. 135–150

[5] L. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin, "The Design and Analysis of Graphical Passwords," in the *8th USENIX Security Symposium*, 1999

[6] W. Jansen, "Authenticating Mobile Device Users through Image Selection," in *The Internet Society: Advances in Learning, Commerce and Security*, K. Morgan & M. J. Spector (Editors), vol. 30, 2004, 10 pp

[7] N. Hopper and M. Blum, "Secure Human Identification Protocols," *ASIACRYPT 2001*, pp. 52–66

[8] N. Hopper and M. Blum, "A Secure Human-Computer Authentication Scheme" CMU Tech Report CMU-CS-00-139

[9] L. Sobrado and J.-C. Birget, "Graphical passwords," *The Rutgers Scholar*, vol 4, 2002, retrieved online on April 14, 2005 at http://rutgersscholar.rutgers.edu/volume04/sobrbirg/sobrbirg.htm

[10] L. Sobrado and J.-C. Birget, "Shoulder surfing resistant graphical passwords," retrieved online on April 14, 2005 at http://clam.rutgers.edu/~birget/grPssw/srgp.pdf