# Graphical passwords

Leonardo Sobrado[*] and Jean-Camille Birget
[1]Department of Computer Science,
Rutgers University, Camden New Jersey 08102

[*]Rutgers Undergraduate Research Fellow

---

Passwords are the most commonly used method for identifying users in computer and communication systems. Typically, passwords are strings of letters and digits, i.e., they are alpha-numeric. Such passwords have the disadvantage of being hard to remember. We discuss graphical passwords, which consist of some actions that the user performs on an image. Such passwords are easier to remember, but are vulnerable to shoulder surfing (which consists of simply watching a user login). We present a few graphical password schemes that offer resistance to shoulder surfing.

## Alpha-numeric passwords

Alpha-numeric passwords were first introduced in the 1960s as a solution to security issues that became evident as the first multi-user operating systems were being developed. As the name indicates, an alpha-numeric password is simply a string of letters and digits. Although almost any string can serve as a password, these passwords only offer good security as long as they are complicated enough so that they cannot be deduced or guessed. Commonly used guidelines for alpha-numeric passwords are:

- The password should be at least 8 characters long.
- The password should not be easy to relate to the user (e.g., last name, birth date).
- The password should not be a word that can be found in a dictionary or public directory.
- Ideally, the user should combine upper and lower case letters and digits.

Since the best password would be a completely random one, people have devised ways to create pseudo-random passwords. One such method is to take a common word and perform certain actions on it. Using the word Dinosaur as an example, users often create passwords such as DiNoSaUr (by alternating upper and lower case), rUaSoNiD (by reversing the string), oSNaiUDr (by shuffling the string), D9n6s7u3 (combining numbers and letters). However, the better the password is, the harder it is to remember.

Another drawback of alpha-numeric password is the dictionary attack. Because of the difficulty in remembering random strings of characters, most users tend to choose a common word, or a name. Unfortunately, there are several tools that allow an

individual to crack passwords by automatically testing all the words that occur in dictionaries or public directories. This attack will usually not uncover the password of a predetermined user; but studies have shown that this attack is usually successful in finding valid passwords of some users of a given system.

## Graphical passwords

Because human beings live and interact in an environment where the sense of sight is predominant for most activities, our brains are capable of processing and storing large amounts of graphical information with ease. While we may find it very hard to remember a string of fifty characters, we are able easily to remember faces of people, places we visited, and things we have seen. These graphical data represent millions of bytes of information and thus provide large password spaces. Thus, graphical password schemes provide a way of making more human-friendly passwords while increasing the level of security.

### Other advantages of graphical passwords:

Dictionary attacks are infeasible, partly because of the large password space, but mainly because there are no pre-existing searchable dictionaries for graphical information. It is also difficult to devise automated attacks. Whereas we can recognize a person's face in less than a second, computers spend a considerable amount of time processing millions of bytes of information regardless of whether the image is a face, a landscape, or a meaningless shape.

## A simple graphical password scheme

The following example, while very unsophisticated, illustrates how a simple graphical password matches the security of its alpha-numeric counterparts. To login, the user is required to click within the 4 circled red regions in this picture. The user chose these regions when he or she created the password. The choice for the four regions is arbitrary, but the user will pick places that he or she finds easy to remember. The user can introduce his/her own pictures for creating graphical passwords. Also, for stronger security, more than four click points could be chosen.

Photo courtesy of Philip Greenspoon

Graphical passwords similar to the one we just described have been known since the mid 1990s, starting with [Blonder]. Various implementations of this idea appear in [Boroditsky]. Different versions are described or analyzed in [Perrig - Song], [JMMRR], [Passfaces], [Dhamija - Perrig], [Isaacson]. Detailed analyses can be found especially in [JMMRR] and [Isaacson]. Those systems, however, use predefined click regions, or click objects; they do not allow the user to choose arbitrary click disks in the image.

**Drawbacks**

Perhaps the biggest drawback for current graphical passwords is the shoulder surfing problem. Although graphical passwords are hard to guess, a person who gets to observe a few login sessions could, depending on the scheme, eventually figure out the password. The above example reveals the password to anybody watching the login session.

**Our goal**

Due to this vulnerability to shoulder surfing, it would appear that graphical passwords could never be used in environments where view of the screen is not exclusive to the person logging in. However, we have found that by applying the concept of challenge response it is possible to create schemes that counter the shoulder surfing problem.

## The shoulder surfing problem

As the name implies, shoulder surfing is watching over people's shoulders as they process information. Examples include observing the keyboard as a person types his or her password, enters a PIN number, or views personal information. Because of their graphic nature, nearly all graphical password schemes are quite vulnerable to shoulder surfing. Most of the existing schemes simply circumvent the problem by stating that graphical passwords should only be used with handheld devices or workstations set up in such a way that only one person can see the screen at the time of login.

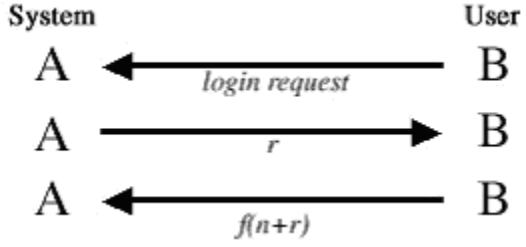
While it is usually possible to ensure that there are no people looking over one's shoulder at the time of login, the value of graphical passwords as an alternative to alpha-numeric passwords diminishes somewhat if they can only be used in environments set up to prevent shoulder surfing.

## Challenge response authentication

Challenge response authentication enables an entity (B) to prove to an entity (A) that (B) knows a secret shared by both (A) and (B). However, this proof of knowledge is done in such a way that the actual secret is not revealed to any third party who may be listening in.

### Typical challenge response session

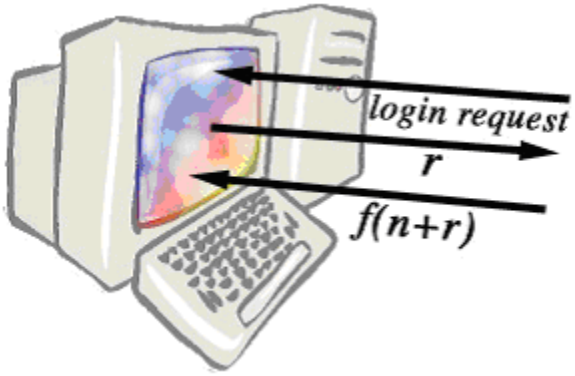
User (B) sends a login request to server (A), which in turn sends back a *random number* ***r***. The challenge for the user is to evaluate ***f(n+r)***. The user's identity is accepted if the last message received from (A) corresponds to ***f(n+r)***. A user who knows n can easily compute ***f(n+r)***. On the other hand, an eavesdropper who captures ***r*** and ***f(n+r***) cannot deduce n in a realistic amount of time. In addition, the use of a random number ***r*** prevents the reuse of previously recorded sessions.




***n*** is the secret shared by A and B.
***f*** is a public one-way function.

## Adapting challenge response to graphical passwords

The challenge response authentication that we just described is not intended to be used directly by humans to authenticate themselves to a system, because it requires many calculations to evaluate an a*lpha-numeric one-way function* for some random value. However, we can use the human ability to process graphical information. The goal is to create a *graphical one-way function* that will prevent an adversary from obtaining the secret even if he or she has full view of the value of the graphic one-way function.



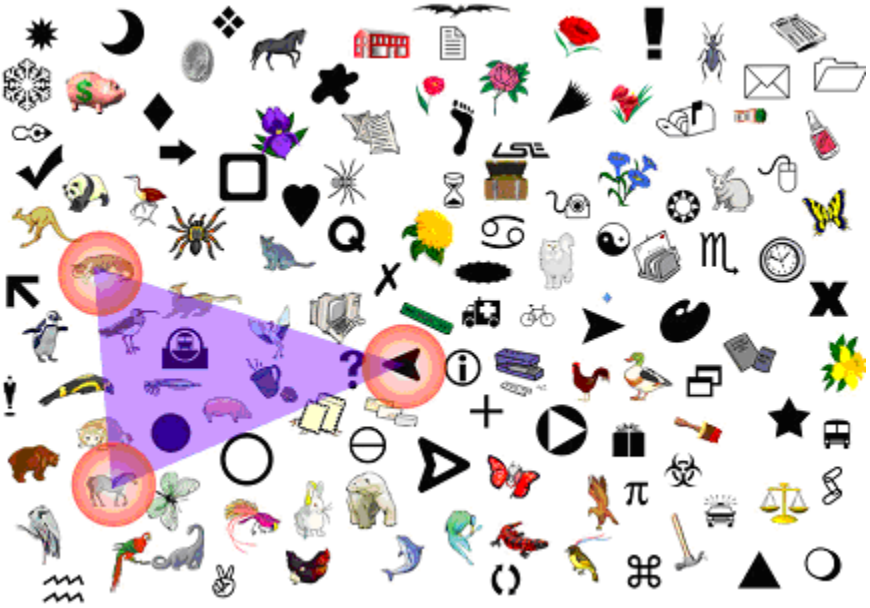

As the figure illustrates, all the adversary would see is ***r*** and ***r***. And although f is publicly available, the secret n is required to solve the next random challenge. However, unlike typical challenge response, the secret n is not alpha-numeric but rather a geometric pattern used to evaluate ***r***. Similarly, ***r*** and ***r*** are graphical. The evaluation of ***f(n+r)*** is done without any computation and can be easily performed by a user in a reasonable amount of time. Instead of sending a random number for each challenge, we can obtain the same functionality by performing certain random operations on an image (e.g., rotation, changes in position, perspective, shading).

## Solving the shoulder surfing problem

### Triangle scheme

The system randomly scatters a set of *N* objects on the screen. In practice, the number *N* could be a few hundred or a few thousand, and the objects should be different enough so that the user can distinguish them. In addition, there is a subset of *K pass-objects* (e.g., *K* = 10) previously chosen and memorized by the user. At login the system will randomly choose a placement of the *N* objects. However, the system first randomly chooses a patch that covers half the screen, and randomly places the *K* chosen objects in that patch. To login, the user must find 3 of the pass-objects and click inside the *invisible triangle* created by those 3 objects. This is equivalent to saying that the user must click inside the *convex hull* of the pass-objects that are displayed. In addition, for each login this challenge is repeated a few (e.g., 10) times

using a different display of some of the *N* objects. Therefore, the probability of randomly clicking in the correct region in each challenge is very low.



For clarity, this collection contains only a little over 100 objects. Typical screens can fit over 1000. All clip art images used to illustrate password schemes are © AAA Clipart.com and have been used with permission. http://www.aaaclipart.com/

The number of possible passwords is the "binomial coefficient" $\binom{N}{K}$ (choose any *K* objects among N). When $N = 1000$ and $K = 10$, the number of possible passwords is hence approximately $\binom{1000}{10} \approx 2.6 * 10^{23}$. This is a little more than the number of alpha-numeric passwords of length 15 ($36^{15} \approx 2.2 * 10^{23}$ ). Having $N = 1000$ objects is not unreasonable (compare with the "Where is Waldo" puzzles, where there are typically tens of thousands of little persons in a picture). Moreover, one can expect a user to choose the *K* objects fairly randomly; or, at least, an attacker (especially a computerized attacker) cannot predict much about which *K* objects a user will choose. On the other hand, the large number of possible alpha-numeric passwords ($36^{15} \approx 2.2 * 10^{23}$) is an illusion: users do not choose alpha-numeric passwords randomly at all.

After an attacker sees one click on the screen from the user, the attacker learns that the *K* pass-objects are such that their convex hull contains the click point. This rules out all the *K*-tuples that do not have the click point in their convex hull. However, when $N = 100$ and $K = 10$, the set of ruled-out *K*-tuples is at least $\binom{500}{10} > 2 * 10^{20}$, which is much too large to be remembered in any computer memory (compare e.g., with the Avogadro number $N_A \approx 6 * 10^{23}$ atom/mole) Hence the attacker can only remember a negligible amount of what he learns in each shoulder surfing

observation. As a consequence, the attacker cannot accumulate knowledge of the user's password. This shows that an exhaustive-search attack is physically infeasible; moreover, when passwords are chosen truly randomly, exhaustive-search attacks are the only possible attacks.

An improved version of this system would display only $N'$ objects (N / 2 $\leq N' \leq$ N) among which $K'$ are pass-objects (with 3 $\leq K' \leq$ K). This simplifies the login of the user, while making attacks harder.

**Movable frame scheme**

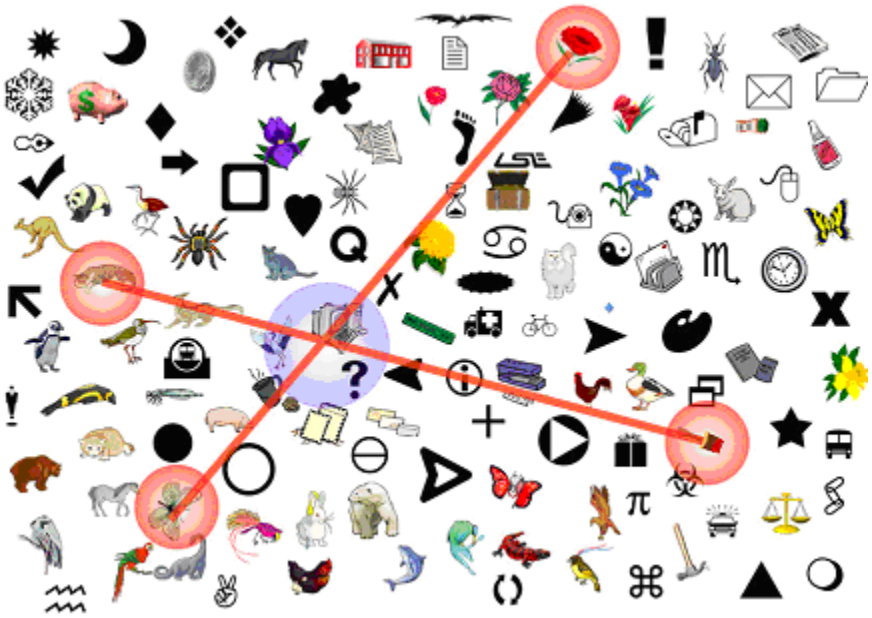
Using the same ideas and assumptions as in the previous scheme, the user must now locate 3 out of *K* pass-objects. This time however, only 3 pass-objects are displayed at any given time and only one of them is placed in a movable frame as depicted below. Which pass-object is displayed within the frame is completely arbitrary.

The task of the user is to move the frame (and the objects within it, like a tape) by dragging the mouse around the frame until the pass object on the frame lines up with the other two pass-objects. As before, this procedure is repeated a few more times to minimize the likelihood of logging in by randomly moving the frame.

**Other special geometric configurations**

Using the same ideas one can achieve more complex ways of telling the user where to click by increasing the number of pass-objects that are displayed at the same time. This scheme uses the intersection of the *invisible lines* formed by 4 pass-objects (out of *K* previously chosen pass-objects). The user must click near the intersection

of the two of these invisible lines, inside the convex quadrilateral formed by those 4 pass-objects. A similar analysis as for the triangle scheme shows that for $N = 1000$ and $K = 10$, the attacker cannot have enough computer memory to carry out an exhaustive-search attack.



## References

[Birget - Hong] J.C. Birget, D. Hong, "Robust discretization, with an application to graphical passwords" (in preparation).

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

[Boroditsky] M. Boroditsky, "Passlogix password schemes". http://www.passlogix.com

[Dhamija, Perrig] R. Dhamija, A. Perrig, "Déjà Vu: User study using images for authentication", 9th USENIX Security Symposium (2000).

[Isaacson] Brad Isaacson, "The password problem", Honor's project, Rutgers-Camden (June 2001).

[JMMRR] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, A. Rubin. "The design and analysis of graphical passwords", 8th USENIX Security Symposium (August 1999).

[Passfaces] "The science behind Passfaces", Real User Corporation (Sept. 2001) http://www.realuser.com

[Perrig - Song] A. Perrig, D. Song, "Hash visualization: A new technique to improve real-world security", Proc. 1999 International Workshop on Cryptographic Techniques and E-Commerce (CryTEC '99).