My Authentication Album: Adaptive Images-Based Login Mechanism

Amir Herzberg and Ronen Margulies

Dept. of Computer Science, Bar Ilan University
{herzbea,margolr}@cs.biu.ac.il

Abstract. We present the design and user study of an adaptive authentication mechanism based on recognition of user-custom images. The mechanism relies on memorizing the custom images on each primary login, and adaptively increasing the authentication difficulty upon failures (suspected impersonation attempts). The constant memorization of the images allows fallback authentication by recognizing all/most of the user's custom images. Our mechanism can be used for multiple authentication scenarios; in particular, it can provide effective phishing protection for primary and/or fallback web login. The mechanism features quick authentication times and low guessing probabilities.

Keywords: web authentication, phishing, fallback authentication, password reset, memorability, human factors, user study, security by design.

1 Introduction

In today's world, users have accounts to many websites, devices and systems which require them to remember a password in order to identify. As seen on [11] and other studies, most users can only remember a limited amount of passwords, especially if those passwords are strong and hard-to-guess. Therefore, many users often forget their passwords. To deal with passwords forgetfulness, two families of countermeasures were suggested:

- Automatic fallback authentication techniques, aiming to allow authentication even if the password is forgotten.
- Memorable alternatives to textual passwords, aiming to *prevent users from forgetting their passwords* in the first place.

Each authentication technique, primary or fallback, should deal with the following security and usability parameters:

Susceptibility to guessing attacks which includes both the password space (which should be large enough to effectively prevent brute-force attacks), and the susceptibility for educated guesses which could significantly reduce the password space.

Susceptibility to phishing primary/fallback authentication techniques should prevent users from submitting their identifying information to a spoofed site.

D. Gritzalis, S. Furnell, and M. Theoharidou (Eds.): SEC 2012, IFIP AICT 376, pp. 315–326, 2012.

Memorability of the identifying information provided by the user.
Usability the authentication times and ease of use.

Fallback authentication techniques ought to be (almost) as secure as the primary authentication in their resistance to guessing and phishing attacks, or else attackers will focus on attacking the fallback authentication mechanism, instead of the primary. In addition, since the fallback authentication process is not used frequently, the memorability of the identifying information has an increased importance. Since *easily-remembered* information might be *easily-guessed*, it is a true challenge to preserve its memorability and prevent its predictability. On the other hand, the fallback authentication could be somewhat slower and less convenient in comparison to the primary login.

Like fallback authentication, alternatives for the primary login also need to be *as secure as the login ceremony they intend to replace*, and should not have worse authentication times.

Textual passwords encounter some additional problems: users often choose passwords that are easily guessed, keyloggers can transfer them to an attacker, and sometimes typing is inconvenient/infeasible.

1.1 The Fallback Authentication Challenge

Today's most common mechanisms for automatic fallback authentication/password recovery are email-based mechanisms and security questions (a.k.a. challenge questions). Email-based mechanisms require access to a preconfigured email address, in order for the site to send back the password or a link to a password reset page. Security questions are used as identifying information, provided by the user during registration and supposed to be known only to the user.

Email-based password recovery reduces the security to that of the email account, which is often unacceptable. Security questions suffer from a few problems. First, they are susceptible to phishing attacks if no effective security indicators are used to help the user identify she reached her target site. This is due to the fact that security questions, like login forms, stimulate an automatic response among users causing them to answer the questions mindlessly. Karlof et al. found 92.7% spoof rate in their user study of security questions [5].

In addition, early and recent studies of leading websites have shown that users tend to forget ∼20% of their answers and they are easily-guessed due to inherently *low entropy* [12, 7, 8]. If a few and/or trivial questions are used, it is easy to find the answers, especially in today's social-networked world, where attackers can harvest all kinds of 'secret' information about users. Using many and/or non-trivial questions makes it harder for users to remember their questions for a long period of time.

Finally, the security questions mechanism does not prevent attackers from accessing the fallback authentication page, where they can try guessing the user's answers, possibly by presenting the same questions to the user and using the answers.

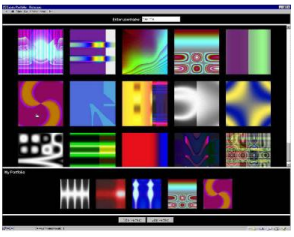

Fig. 1. Dèjá Vu's abstract images selection

1.2 Graphical Passwords

Graphical passwords, a general name for authentication methods where images are used for authentication, were originally proposed as an *alternative to textual passwords* to deal with the problems textual passwords encounter:

- Ease of use on devices where typing is inconvenient (e.g. mobile devices with a touch screen, ATMs).
- Increased memorability – Images are better remembered than textual phrases [9].
- It is hard to write down a graphical password, or tell it to someone orally.
- Resistance to keyloggers.

An example for a graphical password scheme is Dèjá Vu, which was suggested by Dhamija and Perrig[2]. Dèjá Vu uses randomly created abstract images, which are unlikely to be guessed. On the authentication screen Dèjá Vu displays a grid with n images, k of whom are the user's custom images, which the user should identify (by 'clicking') to login; see figure 1.

Graphical passwords seem like a very nice and appealing alternative to textual passwords, and indeed several additional graphical password schemes were proposed [10, 13]. Yet, due to their rather long authentication times ($\sim$30 seconds in [2]), graphical passwords were not widely deployed as a substitute for textual passwords. In fact, for that reason they are *better suited for fallback authentication*.

In particular, a fallback authentication technique which require users to recognize items (as text phrases or images) they like or dislike was suggested by Jakobsson et al. [4]. Jakobsson et al. argue that this method is preferably, since it has higher entropy, and since they believe that preferences don't change much over the years, hence are better remembered.

Yet, in today's social-networked world users often publish their preferences in many areas, so (part of) the fallback authentication information can be found. In addition, as discussed in section 3, this method is susceptible to educated guesses. Finally, in their evaluation Jakobsson et al. found that 16 items (8 likes and 8 dislikes) are needed for sufficient security. We believe that recognizing such a large set of images/items after a long period of time from the initial setup, may prove hard, especially without periodically refreshing the user's memory.

The poor authentication times for primary authentication and poor memorability for fallback authentication raises a dilemma for the usefulness of graphical passwords. We propose a "learn-by-use" login mechanism which provides memorization of custom-images on the primary login while preserving its quick authentication times, and requires the user to recognize all/most of her images when fallback authentication is requested.

1.3 Contributions

We have designed and implemented a learn-by-use authentication mechanism, which is comprised of the following elements:

Single stage authentication requires the user to recognize and click one/few of her custom images from a small set of images on each primary login.

Multiple Stages Authentication requires the user to recognize all/most of her custom images in multiple stages. The multiple-stages authentication is used for fallback authentication or as a difficult authentication scheme when an impersonation attack was detected. The use of multiple stages allows gradually increasing the authentication difficulty by increasing the number of stages.

Detecting impersonation attacks and handling them with different levels of certainty.

Adaptivity using different authentication difficulties on different scenarios – weak authentication when additional identifiers are provided (passwords, cookies) and increasing the difficulty when no identifiers are provided and upon impersonation attacks.

One contribution of our mechanism is the constant memorization of the identifying information (custom images) it provides to the users, which can be used when fallback authentication is needed. Another contribution is the adaptive nature of our mechanism, which detects impersonation attacks and increases the authentication difficulty gradually as the certainty that an attack is witnessed increases. We also present the new concept (to the best of our knowledge) of detecting that part of the authentication secret is exposed to an attacker and asking the user to replace it upon high attack-certainty.

Another contribution is the flexibility of our mechanism, which allows different levels of security and usability. We present two main authentication usages; one that is more stringent and provides two-factor authentication and phishing protection, hence more suitable to sensitive sites; and one that is more alleviate, hence more suitable for less-sensitive sites. In both usages, our mechanism increases the security level of current login ceremonies without increasing the authentication times, provides a secure fallback authentication ceremony and handles impersonation attacks.

Finally, we conducted a long-term user study, whose results confirmed our mechanism's long-term memorability, quick authentication times and effective phishing-resistance.

1.4 Paper Layout

The next section describes our mechanism. Section 3 provides a security analysis of our mechanism in comparison to related work. Section 4 discusses its different applications and section 5 describes our user study and its results.

2 Mechanism Description

Registration. During registration the user chooses (or assigned) a small number of custom images (k) from a large collection of images.

Single Stage Weak Authentication. On the primary login, the system displays a small set of L images, one of whom is one of the user's k custom images which she has to click in order to login. After each *successful* login, a new set of L images is chosen randomly for the next login.

Since the user's custom images are displayed and clicked on each login, the primary login ceremony *constantly refreshes the user's memory* of her custom images, so they are better remembered when fallback authentication is needed. The single stage authentication provides only weak authentication (1/L guessing probability) and in most applications will be used in conjunction with other identifiers (passwords, cookies, etc.). In addition, our mechanism increases its authentication difficulty adaptively upon suspected impersonation attacks, thus the weak authentication can rapidly become strong.

Multiple Stages Authentication. On the multiple stages authentication, which is used for fallback authentication or upon suspecting attacks, the user is displayed with a series of k stages. Each stage displays n images, only one of whom is a user-custom image, and the user has to correctly click all/most of her custom images. If the user has chosen a wrong image in any of the stages, the ceremony continues until her k-th click and only then announces an unsuccessful authentication attempt. See figure 2.

It is important to note that in all stages of *different* authentication attempts, the *same images* are displayed on each stage, or else an attacker will be able to isolate the user's custom images from the non-custom images.

Detecting and Handling Impersonation Attacks. Since one of the user's custom images is displayed on the primary login page, an attacker can find that image by accessing the primary login page and providing an initial identifier. After each successful login, the custom image will differ the next time the user (or the attacker) reaches the login page. This way, the attacker could constantly monitor the login page and check if any new images are displayed, until he finds most/all of the user's custom images.

An impersonation attack includes both a monitoring attack, and guessing attempts of the user's password or custom images. The first stage is to *detect* such attacks, and there are several ways for doing so. The system can notice when a user reaches the login page and does not try to authenticate at all, or

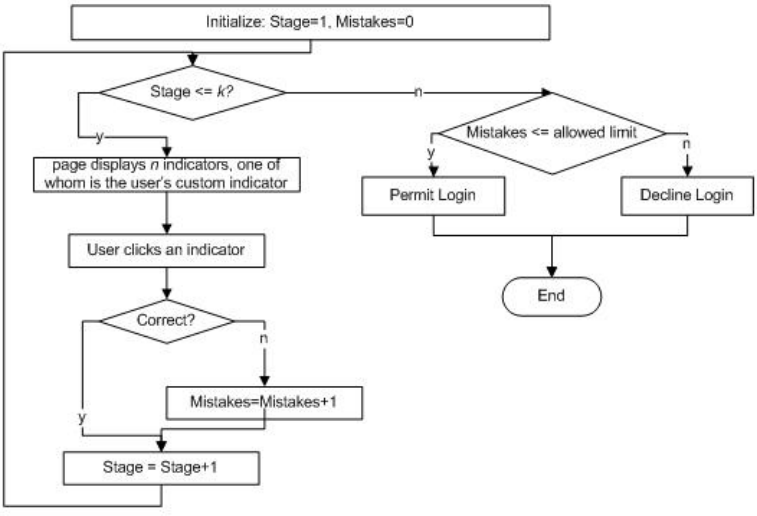

Fig. 2. The multiple (k) stages authentication scheme. The user must click all/most of her custom images.

mistakenly trying to guess the user's correct image (out of L), especially if this happened several times within a pre-determined period of time. Guessing attacks are detected by failed authentication attempts, i.e. submitting a wrong password or clicking incorrect images on the fallback authentication page.

After the system detected a suspected attack it increases the authentication difficulty adaptively to prevent the attacker from gaining more of the user's custom images or apply guessing attacks. As the difficulty increases, the attacker is likely to make more mistaken clicks or disconnect without trying to authenticate, which increases the attack's certainty even more.

First, the system can increase the number of images displayed on the login page (for example, double them) after reaching some certainty threshold. Second, the system can apply the multiple stages scheme for increased difficulty. Third, the amount of stages can also be increased to $m>k$ stages, each containing a "none of my images are here" choice, and only k of the stages display a user-custom image. The authentication difficulty can be further increased by using hindering measures such as CAPTCHAs between the stages.

After a pre-configured level of certainty was reached, the system can ask the user to replace her initial identifier (discussed on next section), and by logging the images that the suspected attacker correctly clicked, can also ask the user to replace those images. The site can also deal with guessing attempts of the textual password; if the guessing attempts were close enough to the user's correct textual password (for example, a small edit distance value), the site can ask the user to replace her password.

On [6] we provide an algorithm for detecting and handling impersonation attacks, with varying levels of certainty, by using all the above-mentioned measures.

3 Security Analysis

In this section we analyze the security strength of our method in comparison to Dèjá Vu [2] and preferences-based security questions [4] we discussed previously. The theoretical password space of our mechanism is n^k and Dèjá Vu's is $\binom{n}{k}$ (390,625 and 12,650 respectively for reasonable values of n=25 and k=4). The theoretical password space of preferences-based security questions is $\binom{16}{8} = 12,870$, similar to Dèjá Vu's.

As seen on Davis et al.'s user study [1], user chosen images/items can be highly predictable, especially among certain populations (based on gender, geographic location, age, etc.). It was evidenced in [1] and in [2] that more appealing images and popular items (objects, places, food, etc.) have higher probabilities to be selected by users. In addition, correlations between images/items also exist. Therefore, basic prior knowledge of the users' background and preferences (taken for example from social networks) as well as knowledge of the selection probabilities and correlations (among the general population or different populations) can significantly increase the guessing probabilities in all three methods.

With preferences-based security questions, a unique correlation also exist: category-based correlation. Since users are requested to select their items from different categories, a person who likes sports, for example, might choose most of his 'likes' from the sports category. The mere fact that a person likes one of the categories in general could be quite easily drawn from social networks. If users are allowed to choose only few likes/dislikes from each category, then they won't choose their most liked and most disliked items, resulting in weaker memorability.

To avoid different probabilities and correlation as much as possible, the images used by a site implementing images-based authentication should:

- Be as general as possible (e.g., abstract images) and/or taken from the same category/topic. The site's topic can be a good idea for the images portfolio.
- Be unfamiliar. For example, a traveling agency should use general images of landscapes and urban views, instead of popular sites.
- Have the same level of appeal.

Obviously, preferences-based security questions cannot follow those guidelines. Therefore, it is likely that our method and Dèjá Vu have lower guessing probabilities than preferences-based security questions, and the actual password space should be close to the theoretical password space (though correlations and different probabilities are still possible with abstract images or images from the same topic). Another advantage over preferences-based security questions is the ability of the site to continuously learn guessing probabilities and correlations, and to remove images with high/low probabilities and avoid displaying correlating images together in the setup phase of a specific user. Therefore, since

Dèjá Vu has a similar theoretical password space as preferences-based security questions, but reduced guessing probabilities (with a good choice of the images portfolio), its security is stronger. In addition, since the guessing probabilities of our method and Dèjá Vu's are similar, our larger theoretical password space gets the lead.

Another advantage of our method in comparison to Dèjá Vu and preferences-based security questions is the ability to increase the authentication difficulty to the m-stages scheme. Suppose the attacker knows c of the user's custom images; with the k-stages scheme his guessing probability is $(\frac{1}{n})^{k-c}$. With the m-stages scheme, he has to guess the correct k-c stages where the other custom images are displayed, and then guess the correct images on those stages. This reduces the guessing probability to $\frac{1}{\binom{m-c}{k-c}} \cdot (\frac{1}{n})^{k-c}$.

4 Applications

4.1 Authentication to Highly-Sensitive Sites

On [3] we describe an extensive long-term user study we conducted, which examined login mechanisms that force and train users to login safely and resist phishing attacks. We tested two *forcing functions* mechanisms:

A login bookmark which forces users to reach the site's login page via the bookmark by not allowing them to login when reaching it otherwise. The login page looks for a secret token contained in the bookmark link and presents an error message if a familiar token is not provided. This mechanism also provides *two-factor authentication* as both the secret token and the password are necessary for authentication.

An interactive user-custom image which forces the user to look for and click her custom image on the login page, by hiding the password text field until the custom image is clicked.

The study's results showed that a combined method, which uses a login bookmark with an interactive custom image, displays several images in the login page, and training the user not to follow links by intentionally providing 'non-working' links, received outstanding prevention and detection rates, and resisted 93% of the attacks.

We exploited the idea of the login bookmark and interactive custom images, and used them in conjunction with our learn-by-use authentication mechanism. The only difference of the primary login in comparison to the combined method in [3] is that the user now has k custom images instead of just one, and one of them is displayed on the login page along with L-1 other images (e.g. L=4). In addition to the site identification that the image click provides to the user, it also provides her with the necessary memorization of the custom images.

Users can choose to go to the fallback authentication page when they have forgotten their passwords or when using a mobile application/browser on a mobile device, where it might be easier to click images than typing the password.

Access to a user's fallback authentication page requires the secret token, just like the primary login. This prevents anyone who doesn't have the secret token from accessing the fallback authentication page, and from performing guessing attacks or spoofing the fallback authentication page.

Entrance to the fallback authentication page can be done via the primary login page; After the bookmark click, the site initially identifies the user by her secret token and sends back the login page with the user's custom image. After the user clicks her custom image, the site reveals, in addition to the password text field, a button which leads the user to the fallback authentication page. This promises that the user effectively identifies the site before trying to authenticate. It is important to note that the login bookmark itself isn't necessary to achieve two-factor and two-sided authentication, only the secret token is. An alternative can be a cookie containing the secret token, which should be previously installed via a secondary channel on each computer the user uses (for example with a registration email). Yet, we recommend using the login bookmark for several security and usability advantages discussed on [3].

We designed and implemented WAPP (Web Application Phishing Protection), a server side solution which implements the above mentioned mechanisms for primary login, as well as our fallback authentication mechanism (see `http://submit2.cs.biu.ac.il/WAPP/` for a live demo).

4.2 Authentication to Less-Sensitive Sites

The mechanism we described can be altered for less-sensitive sites which do not aim to provide effective phishing protection or two-factor authentication. Many of today's sites that require authentication (e.g. gmail, facebook) provide the user with an option to create an authentication cookie containing a secret token when the user provides her password for the first time on a certain browser (usually with a "keep me signed in" check box). On the next logins from the same browser, those sites don't require a password at all if the cookie is installed. Our mechanism can also be used on such sites:

- When an authentication cookie is already installed, the site can display one of the user's custom images, together with several non-custom images (e.g., L=9) which provides the necessary memorization and further security (since a password is not required). The mechanism can be set to allow no mistakes and double the amount of images immediately after a single mistaken click (and increasing the difficulty rapidly on further mistakes). If a password is also required, the mechanism also provides phishing protection.
- When an authentication cookie is not installed, no other initial identifier (such as the username) should be used instead. If the username is used to initially recognize the user and display her image, then the site is susceptible to MITM attacks: the attacker can spoof the initial page where the user provides her username, forward the username to the original site, get the image and display it to the user, and finally get the password from the user. This even provides a false-sense-of-security to the users.

Therefore, the site should first ask for the password and only then display the images. This does not provide phishing protection, but it provides the necessary memorization and further weak (and increasing) security.
- If the user has forgotten her password or uses a mobile device, the site can allow her to authenticate using the k-stages authentication by providing her username only.

5 User Study

5.1 Study's Framework System

For our long-term phishing study in [3], we used an online exercise submission system called 'submit' (`submit.cs.biu.ac.il`), which is used by most courses at the computer science department of Bar-Ilan University. With the submit system, students submit their exercises and receive emails announcing new grades. Due to its wide usage, most users logged-in to the system dozens to hundreds of times throughout the study. We have used the system for three semesters, among a population of ∼400 students. In addition to its primary usage as an exercise submission system, we simulated several phishing attacks and collected the results.

We extended the phishing study to test our fallback authentication mechanism in the fourth semester. Users were asked to choose 3 or 4 new custom images (in addition to their existing custom image). One of their custom images was randomly selected and displayed on the login page on each login (along with three other non-custom images). Users were tested with the k-stages authentication scheme twice – immediately after they have chosen their additional custom images and once again 2-3 months later, while using the system regularly (and clicking one of their images on each login). In both tests, users had only one authentication attempt, and they were not forced to succeed in order to proceed with their exercise submission.

We logged the success rates and authentication times, and also examined if there was a decrease in the detection rates of spoofed login pages in comparison to the previous semesters (where users had only one custom image), especially for spoofed login pages displaying only fake images.

5.2 Results

The memorability of the custom images and the authentication times are of course affected by the amount of entrances to the system between the setup and the authentication attempt. The more times a user enters the system, the better she remembers her images and authenticates faster. Table 1 summarizes our results and compares them to Dhamija and Perrig's study [2], where all users authenticated once after one week from the setup phase. The table displays the results of users from the top 25% of entrances (21+), top 50% of entrances (8+) and a base line of minimum entrances we chose (5+). The results show reasonable authentication times, similar to Dèjá Vu.

Table 1. Memorability and authentication times of our method in comparison to Dèjá Vu. We expect that allowing several login attempts will improve our method's results.

method	entrances	images used	weeks from setup	no mistakes	up to one mistake	auth. attempts	auth. time (seconds)
Ours	5+	photos+abstract	8-12	65.3%±11.2	87.7%±7.7	1	38
Ours	8+	photos+abstract	8-12	67.5%±12.2	90%±7.8	1	38
Ours	21+	photos+abstract	8-12	84.2%±13.8	94.7%±8.4	1	31
Dèjá Vu	0	photos	1	95%±8	100%	no limit	27
Dèjá Vu	0	abstract	1	90%±11	100%	no limit	32

The success rates of users who entered the system eight times or more are quite good, especially if the system allows a single mistake. For users who entered the system more than 20 times, the results are very good even when no mistake is allowed. The results are expected to improve even more if several attempts are allowed and if users will *have to* succeed to use the system. Yet, it can be assumed that users remembered their custom image that preceded the experiment, so the results can be somewhat reduced. For a better picture, a longer-term experiment, which compares our method with other techniques, is still needed.

Surprisingly, we found better success rates for users with five custom images than users with four, suggesting that using five images and allowing a single mistake is the best option. Even with four images (out of five) needed for authentication, the password space of our method is larger than the password space of Dèjá Vu with five images[1].

Finally, there was *no degradation* in the detection rates of spoofed pages for users with eight or more entrances, and no increased false negative rates (falsely reporting a non-spoofed page).

6 Conclusions and Future Work

We have described an images-based authentication mechanism which reminds the user of her custom images on each primary login, so they will be highly memorable when a fallback authentication is needed. Our fallback authentication mechanism was shown to be quick and memorable after sufficient training on our user study. Our mechanism also increases its difficulty adaptively upon detecting impersonation attacks. The mechanism is well suited for different web authentication scenarios, providing different levels of security and quick authentication times. Our fallback authentication scheme provides a large password space and high resistance to guessing attacks, and for sensitive sites it also provides effective phishing protection.

Another important authentication scenario that our mechanism suits for, is the authentication to mobile devices. Since authentication to mobile devices should be quick (as it is done very frequently) and the mere possession of

[1] $25^4/\binom{5}{4}$=78,125 and $\binom{25}{5}$=53,130 respectively.

the device can be assumed to provide some identification, than the authentication can start with our single-stage authentication and rapidly increase its difficulty upon each mistaken click. It can also be combined with other identifying measures.

Acknowledgments. This research was supported by grant 1354/11 from the Israeli Science Foundation (ISF).

References

[1] Davis, D., Monrose, F.: On user choice in graphical password schemes. In: Proceedings of the 13th USENIX Security Symposium. USENIX (August 2004)
[2] Dhamija, R., Perrig, A.: Dèjá vu: a user study using images for authentication. In: Proceedings of the 9th conference on USENIX Security Symposium, vol. 9. USENIX Association, Berkeley (2000)
[3] Herzberg, A., Margulies, R.: Forcing Johnny to Login Safely. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 452–471. Springer, Heidelberg (2011)
[4] Jakobsson, M., Yang, L., Wetzel, S.: Quantifying the security of preference-based authentication. In: DIM 2008: Proceedings of the 4th ACM Workshop on Digital Identity Management, pp. 61–70. ACM, New York (2008)
[5] Karlof, C., Tygar, J.D., Wagner, D.: Conditioned-safe ceremonies and a user study of an application to web authentication. In: SOUPS 2009: Proceedings of the 5th Symposium on Usable Privacy and Security (2009)
[6] Margulies, R.: Usable and phishing-resistant authentication mechanisms. Master's thesis, Bar-Ilan University (July 2011), `http://submit2.cs.biu.ac.il/WAPP/thesis.pdf`
[7] Rabkin, A.: Personal knowledge questions for fallback authentication: security questions in the era of facebook. In: SOUPS 2008: Proceedings of the 4th Symposium on Usable Privacy and Security, pp. 13–23. ACM, New York (2008)
[8] Schechter, S., Brush, A.J.B., Egelman, S.: It's no secret. measuring the security and reliability of authentication via 'secret' questions. In: SP 2009: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, pp. 375–390. IEEE Computer Society, Washington, DC (2009)
[9] Shepard, R.N.: Recognition memory for words, sentences, and pictures. Journal of Verbal Learning and Verbal Behavior 6(1), 156–163 (1967)
[10] Suo, X., Zhu, Y., Owen, G.S.: Graphical passwords: A survey. In: Proceedings of the 21st Annual Computer Security Applications Conference, pp. 463–472. IEEE Computer Society, Washington, DC (2005)
[11] Vu, K.-P.L., Proctor, R.W., Bhargav-Spantzel, A., Tai, B.-L.B., Cook, J., Eugene Schultz, E.: Improving password security and memorability to protect personal and organizational information. Int. J. Hum.-Comput. Stud. 65(8), 744–757 (2007)
[12] Zviran, M., Haga, W.J.: User authentication by cognitive passwords: an empirical assessment. In: JCIT: Proceedings of the Fifth Jerusalem Conference on Information Technology, pp. 137–144. IEEE Computer Society Press, Los Alamitos (1990)
[13] Passfaces™: Graphical password technology, `http://realuser.com/` (last visited December 2011)