*In Proc. IEEE Symposium on Security and Privacy (S&P)*, May 2006

# Cognitive Authentication Schemes Safe Against Spyware (Short Paper)

Daphna Weinshall
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Jerusalem Israel 91904, daphna@cs.huji.ac.il

## Abstract

*Can we secure user authentication against eavesdropping adversaries, relying on human cognitive functions alone, unassisted by any external computational device? To accomplish this goal, we propose challenge response protocols that rely on a shared secret set of pictures. Under the considered brute-force attack the protocols are safe against eavesdropping, in that a modestly powered adversary who fully records a series of successful interactions cannot compute the user's secret. Moreover, the protocols can be tuned to any desired level of security against random guessing, where security can be traded-off with authentication time. The proposed protocols have two drawbacks: First, training is required to familiarize the user with the secret set of pictures. Second, depending on the level of security required, entry time can be significantly longer than with alternative methods. We describe user studies showing that people can use these protocols successfully, and quantify the time it takes for training and for successful authentication. We show evidence that the secret can be maintained for a long time (up to a year) with relatively low loss.*

## 1. Introduction

We address the problem of user authentication over insecure networks and from potentially compromised computers, such as in internet cafes. In such cases there is a high risk that an eavesdropping adversary may record the communication between the user and the main computer, before it is possible to rely on the protection of secure encryption. We assume that this adversary can record all information exchanged during the authentication, including user input (such as keyboard entries and mouse clicks) and screen content. It is therefore necessary to develop secure authentication protocols, where overhearing a sequence of unencrypted successful authentication sessions will not let the adversary pose as the legitimate user at a later time.

Clearly our everyday non-user-friendly password in not secure in the sense we require - by merely recording the input of the user to the intermediate computer, the adversary can discover the user's password after a single successful authentication session. Biometric identification (based on such physiological traits as fingerprints and iris shape) is indeed more secure against theft or forgetting, but it is just as easy for the adversary to obtain this key as it is to obtain a password. There are a number of existing secure solutions which require the user to carry a computational aid, such as an OTP card that generates one time passwords, one-time password sheets, or a laptop armed with secure authentication protocols. But this approach has its drawbacks: users cannot get authenticated without the device, which can be stolen, lost, or made unusable (e.g., when its battery runs out).

Can we develop a user authentication scheme that is secure against eavesdropping adversaries, and yet can be used reliably by most humans without the need for any external computational aid? Not much has been said about this problem. Recent systems have been developed which use easy to remember passwords, including abstract art pictures like in the "Deja vu" system [2], graphical passwords (see survey in [6]), or memorized motor sequences. Most of these schemes use passwords that are indeed easier to remember, but otherwise are not any safer than regular passwords against eavesdropping adversaries. A few recent cryptography papers tried to address this issue, but their proposed protocols are either not secure for any sufficient length of time [5, 4], or impractical in that most humans cannot reliably use them [3]. In our previous work [9], we took advantage of the vast capacity of human memory to design protocols that use each memory item only once, and are therefore as safe against eavesdropping as one time passwords.

Here we propose a challenge response protocol, where authentication is based on the user answering correctly a sequence of challenges posed by the computer (cf. [3]). The challenges (or queries) are based on a shared secret between the computer and the user, which consists of a random division of a fixed set of

pictures into two sub-groups. Authentication is done via a challenge-response protocol: the computer poses a sequence of challenges to the user, which can only be answered correctly by someone who knows the shared secret. Once the probability of random guessing goes below a fixed threshold, the computer authenticates the user.

The proposed protocols have the following two important characteristic: (i) The password is a random (machine generated) division of a fixed set of pictures. As a result, the space of used passwords is as big as the space of all possible passwords, which implies safety against dictionary attacks. This benefit is obtained at the cost of relatively long training time, where the user is familiarized with the secret in a secure location. (ii) The interactive nature of the protocol is intended to make it resistant to attacks by adversary eavesdroppers, including what is sometimes referred to as spyware and shoulder-surfing. This benefit is obtained at the cost of relatively long login time of a few minutes.

The main advantage of our method over [3] is its relative human-friendliness. In Section 3 we report user studies with 11 naive participants, showing that the protocols can be effectively used by these participants, with high reliability and for a long period of time. However, unlike the protocols described in [3, 4], we are not able to provide any formal analysis proving that the protocols are indeed safe. Instead, in Section 2.3 we analyze the best brute-force attacks, identifying the range of parameters which make them impractical.

## 2. Authentication protocol

We define the following authentication scheme. The computer assigns to each user two sets of pictures: (i) A set $\mathcal{B}$ of $N$ common pictures. (ii) A set $\mathcal{F} \subset \mathcal{B}$ of $M < N$ pictures. Set $\mathcal{B}$ is common knowledge and may be fully or partially shared among different users. Set $\mathcal{F}$ is arbitrarily selected for each user, and its composition is the essence of the shared secret between the user and the computer; typically $M < \frac{N}{2}$.

During training in a secure location, the user is trained to distinguish the pictures in the set $\mathcal{F}$ from the remaining pictures in the super-set $\mathcal{B}$.

During authentication, the computer randomly challenges the user with the following query:
(1) A set of $n$ pictures is randomly selected from $\mathcal{B}$. In the example of Fig. 1 $n = N = 80$, and therefore all the pictures from $\mathcal{B}$ are shown.
(2) The user is asked a simple multiple-choice question with $P$ possible answers about the random set, which can be answered correctly only by someone who knows which pictures in the random set belong to $\mathcal{F}$. In the example of Fig. 1 $P = 4$, and the multiple-choice question appears at the top of the panel, letting the user choose one of 4 integers in the range [0..3].
(3) The process is repeated $k$ times; after each iteration, the verifier computes the probability that the sequence of answers has been generated by random guessing. Specifically, we use the following model: if the user has made $e \leq k$ errors, compute the probability to obtain $e$ or fewer errors in a sequence of $k$ Bernoulli trials with $\frac{1}{P}$ chance of success.
(4) The computer stops and authenticates the user when the probability of correctly guessing (as estimated in the previous step) goes below a pre-fixed threshold $T$. If this is not accomplished within a certain number of trials, the user is rejected.

The parameters defined above determine the security and convenience of the scheme. Large values for $M, N$ increase security, but also prolong the training time. Large values of $n$ and small values of $P$ increase security by reducing the information exposed in each observed query, but also prolong the login time. Finally, small values of $T$ increase security against random guessing, but prolong the login time.

Note that the (possibly compromised) authentication machine is given by the server, for each query, the pattern to be displayed on the screen and the required answer. This does not compromise security, even if this information is stolen, as the protocol is designed to be safe against adversaries with such knowledge.

Next we discuss how to construct the multiple-choice queries (Section 2.1), how to conduct effective training (Section 2.2), and how resistant our protocols are to brute-force attacks (Section 2.3).

### 2.1. Query construction

The query should be easy for humans to compute unassisted, and cannot therefore include complex mathematical operations. We define two types of queries which trade-off training time and login time, and which also differ in the characteristic of their security against various attacks.

#### 2.1.1. High complexity query

In this protocol we use a public set of $N = 80$ pictures $\mathcal{B}$, and a secret subset of $M = 30$ pictures $\mathcal{F} \subset \mathcal{B}$. In each query all $n = N = 80$ pictures from $\mathcal{B}$ are shown in random order. The user is asked a relatively complex question about the set of pictures, with $P = 4$ possible choices.

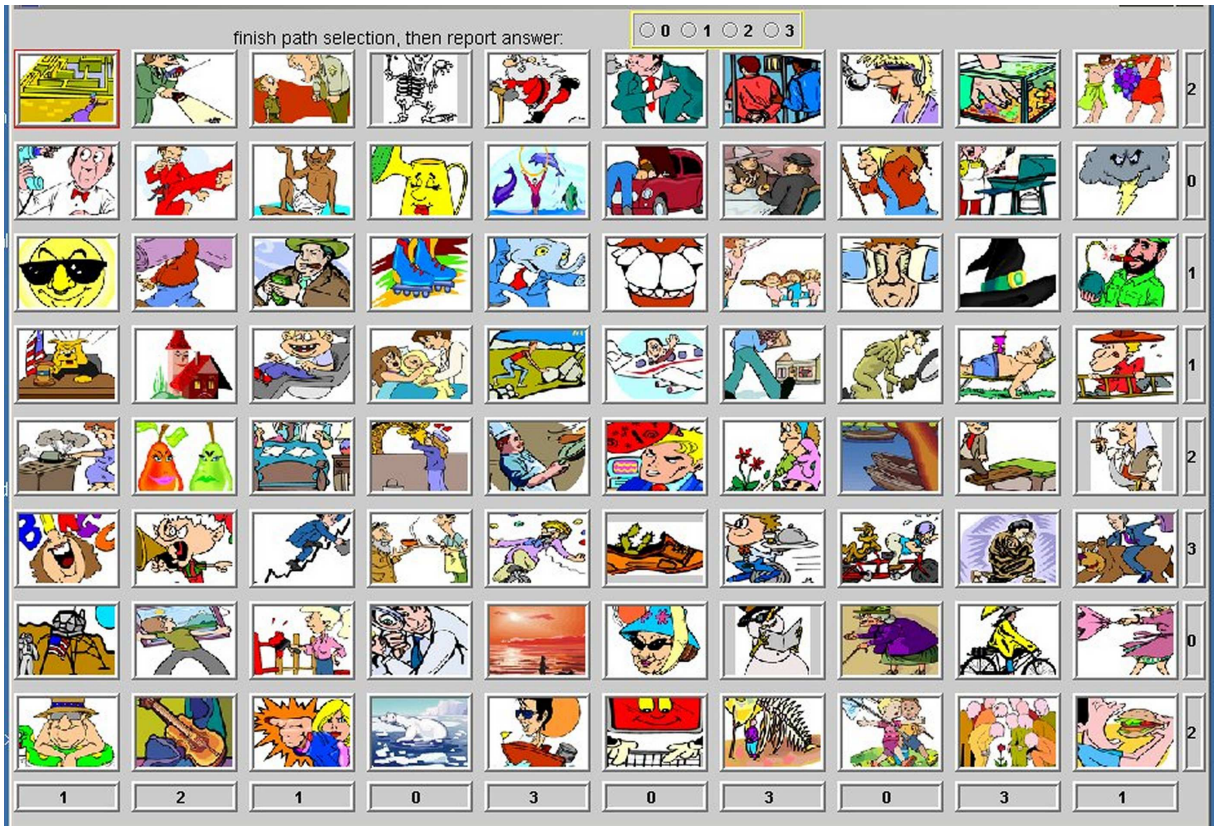


**Figure 1.** A high complexity query panel (best seen in color).

Specifically, the 80 pictures are presented in a panel composed of an $8 \times 10$ regular grid (see Fig. 1). Users are asked to (mentally) compute a path, starting from the top-left picture in the panel (marked by darkened red boundary in Fig. 1). The rules of movement along the path are the following: move down if you stand on a picture which is in $\mathcal{F}$, move right otherwise. Finish when you reach the right-most end or bottom end of the panel (the respective column and row in Fig. 1 composed of numbers and not pictures), and record the final number you have reached.[1]

The terminal row and column include the numbers $[0, 1, 2, 3]$. The numbers are distributed in such a way that the probability to reach each of them is roughly $0.25$, assuming that queries are built from independent random permutations of the original set of pictures.

In Section 2.3 we show that this protocol is secure against selected brute-force attacks by adversaries whose space or time complexity is smaller than $2^{47}$. If we reduce the number of choices to $P = 2$ (a binary query), we get a better bound of $2^{56}$ (result not shown).

### 2.1.2. Low complexity query

In this protocol we use a public set of $N = 240$ pictures $\mathcal{B}$, and a secret subset of $M = 60$ pictures $\mathcal{F} \subset \mathcal{B}$. In each query $n = 20$ random pictures from $\mathcal{B}$ are shown, and the user is asked a relatively simple binary question ($P = 2$) about a few of these pictures.

Specifically, the 20 pictures shown in each query are randomly selected from the set $\mathcal{B}$, and presented in a $4 \times 5$ panel. Each picture is assigned a random bit (0 or 1), which is shown next to it. (The bits are balanced, with 10 random pictures assigned 0 and the rest assigned 1.) The user is instructed to scan the panel in order: from left to right, one row after another, and identify the first, second and last pictures from subset $\mathcal{F}$. She should then compare the 3 associated bits, and answer whether their majority is 0 or 1.

This query is not susceptible to brute-force attacks, since the number of all possible secrets is $\binom{N}{M} \approx 2^{190}$. This is achieved by increasing the size of the secret

[1] Although the verbal explanation of this computation is somewhat lengthy and appears diffi cult, it is actually easy and intuitive for people to perform. The only diffi culty, which makes it "complex", is the need to mentally scan many pictures, which simply takes time.

as measured by $N, M$ (compare to the high complexity query above), which implies longer training time and lower reliability in memory retention. On the other hand, because each answer depends on only 3 pictures in the query, this protocol is susceptible to probabilistic attacks. In simulations of such attacks (omitted, see [8]) we see evidence that the security may be sufficient against most practical adversaries.

## 2.2. Training procedure

Training is composed of multiple sessions, with two phases:

*Phase 1*, in which the user is familiarized with the subset of pictures $\mathcal{F}$ in isolation. In this phase all the pictures from $\mathcal{F}$ are shown once in random order. Each picture is shown for a few seconds, at which time the user is instructed to memorize the picture for a second or so. He then answers a multiple choice question (unrelated to the queries used in the authentication protocol) which depends on the details of the memorized picture, see [8] for details.

*Phase 2*, in which the user is presented with random query panels including pictures from the set $\mathcal{B}$ as defined in the authentication protocol. First, the user is asked to identify and mark the pictures from $\mathcal{F}$. Second, the user is asked to answer the multiple-choice question associated with the query as in the actual authentication protocol. During this phase the user receives full feedback.

All participants underwent two mandatory training sessions on two consecutive days: on the first day each participant performed *phase 1* only; on the second day each participant performed both phases consecutively. Participants were offered a choice of a third training session on the third consecutive day, which included *phase 2* only; they were instructed to choose this option if feeling low confidence in their ability to perform the task without feedback. Three participants trained on the high complexity query, and the two participants trained on the low complexity query, chose this option.

## 2.3. Resistance to various attacks

Let $\tilde{H} = \binom{N}{M}$ denote the complexity of the *brute-force attack* against our protocols. Recall that since passwords in our protocols are randomly generated by a machine, all passwords are equally likely. Therefore there is no *dictionary attack* that can improve over the *brute-force attack*. We also define $\hat{H}$ to denote the complexity of an *enumeration attack* - a "clever" brute-force attack which uses to its advantage the fact that only $n \leq N$ bits are shown at any given query, and that not all of them are used in the query computation. This attack still checks all the possible passwords; however, it keeps a list of all possible passwords in a succinct form, and as a result can get by with less bookkeeping. Our protocols' resistance is measured by $\min(\hat{H}, \tilde{H})$, which should be "large enough".

### 2.3.1. The model

Let $h$ denote an $N$-dimensional trinary vector representing a consistent hypothesis about the user's secret. Each index $i$ in this vector corresponds to an item in the set $\mathcal{B}$; the value of $h_i$ is 1 if the corresponding item is in the secret set $\mathcal{F}$, 0 if it isn't, and -1 if its affiliation is not known (in other words, it can be either 0 or 1). To help the adversary in the construction of consistent hypotheses, we assume that the adversary knows the correct answer for each observed query.

**Brute-force attack:** $\tilde{H} = \binom{N}{M}$ denotes the complexity of the *brute-force attack* against a protocol. It is the number of all vectors $h$ with exactly $M$ 1's and the rest all 0's.

All the protocols described in this paper are not safe against a very powerful adversary, which can successfully mount a brute-force attack. Note that the first observed query and its (correct) answer reduces the number of possible secrets by $\frac{1}{P}$. After $k$ observations, the number of viable secrets is reduced to $\frac{1}{P^k}$ of the original space. Thus if an adversary can maintain a list of all possible $\tilde{H}$ solution vectors $h$, this elimination process will converge in a short time to a single answer, and the secret will be revealed. We must therefore choose $N$ and $M$ large enough, so that $\tilde{H}$ is larger than the resources available to the most powerful adversary, against which we should be protected.

**Enumeration attack:** Let $\mathcal{H}_t$ denote the set of all possible different hypotheses which are consistent with the observed data at time $t$, i.e., after $t$ observations of queries and (correct) answers. Let $\hat{H} = \max_t |\mathcal{H}_t|$. $\hat{H}$ is the complexity of an *enumeration attack* - an attack that keeps a list of all consistent hypotheses remaining at time $t$, which are consistent with all the data observed up to time $t$. Clearly $\hat{H} < 3^N$. However, typically in our protocols $\hat{H} < \tilde{H}$ since the answer to each query depends on only a fraction of the bits in the secret.

To see why this is true, let us inspect the *enumeration attack* more closely. This attack works the opposite way from the *brute-force* attack: instead of starting from the set of all possible secrets (i.e., all vectors in $2^N$ with exactly $M$ 1's) and eliminating those inconsistent with the observations, the *enumeration attack* works by expanding and maintaining the set of all consistent hypotheses.

Initially, before any observation has been made, this set includes a single element, the constant $N$-dimensional vector with all $-1$. Therefore $|\mathcal{H}_0| = 1$. As the number of observations increases, $\mathcal{H}_t$ is modified to include all the vectors consistent with the old and new observations. The size of this set initially increases exponentially (roughly as $q^t$, if $q$ is the number of vectors consistent with one observed query). But at some point contradictions start to appear between vectors consistent with past and present observations, and eventually the size of the sets $\mathcal{H}_t$ starts to shrink back until a single element remains. The remaining vector must be a binary vector representing the user's secret, where each item in the secret is assigned 1, and each of the remaining items is assigned 0. It then follows that, because the *enumeration attack* must store and access all sets $\mathcal{H}_t$, its complexity (time and space) is $\hat{H} = \max_t |\mathcal{H}_t|$.

#### 2.3.2. High complexity query

We use simulations to estimate $\hat{H}$ for the high complexity protocol described above, based on the sampling of $\mathcal{H}_t$ (exact calculation for interesting values of $N, M$ is not computationally feasible). The results are shown in Table 1 for a range of parameters, illustrating the various trade-offs one should be aware of. The set of parameters corresponding to our user study below appears in the first row. In this case our protocol achieves only moderate security in current cryptography standards; however, recall that the attacker must see a number of successful authentication entries before being able to even start searching the space of $2^{47}$ possibilities. The second case (row 2) shows a set of parameters which achieves a higher level of security, consistent with what is currently considered secure for encrypted passwords.

**Table 1.**

| $N$ | $M$ | $P$ | query size | # bits $\hat{H}$ | # bits $\tilde{H}$ |
|---|---|---|---|---|---|
| 80 | 30 | 4 | $8 \times 10$ | 47 | 73 |
| 120 | 50 | 2 | $8 \times 10$ | 84 | 114 |
| 95 | 40 | 8 | $8 \times 10$ | 47 | 89 |
| 145 | 55 | 4 | $4 \times 5$ | 47 | 135 |

The last two cases show the effect of two manipulations that can shorten the authentication process: either increase the number of choices in the multiple choice question (row 3) to decrease the total number of queries required, or simplify the query by decreasing the number of pictures shown in the grid (row 4) to shorten each query. In both cases we show values of $N, M$ which achieve the same level of security against the conjectured attacks as in row 1 - 47 bits.

## 3. User study

We tested the two types of protocols described above in a user study that spanned a long period of time: up to 6 months with the high complexity query, and roughly 1 year with the low complexity query. The goal was to probe password memorability and the protocols' ease of use. Results are described below.

### 3.1. High complexity query

The protocol is described in Section 2.1.1. 9 undergraduate students from the faculty of natural sciences (none of which was a computer science student), all in their mid 20's, participated in the study, and were paid a fixed amount per session. All participants underwent two or three training sessions on three consecutive days, as described above. With a few exceptions of isolated queries, all participants took roughly 15-20 seconds to conclude a query.

After training, participants went through a sequence of experimental sessions including 24 queries each, in the following time schedule: 1 days after training, 2 days, 5 days, 1 week after training, and then roughly once a week for a period of 10 weeks. All participants were committed to this schedule. Some of the participants agreed to participate in additional sessions, which took place on a loose schedule for the next 4 months. Participants received feedback as to whether the final answer they had reported was correct, which resulted in high memory retention and self correction. Some errors, however, did not correspond to failing memory but to lapses in concentration. Results for all participants are summarized in Fig. 2.

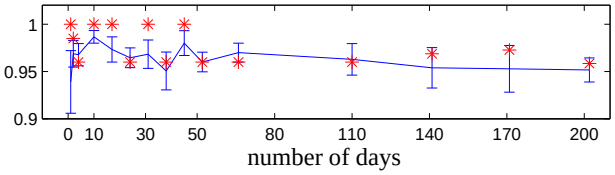


**Figure 2.** Results (success rate) of 9 participants answering high complexity queries, as a function of the time that has passed since the last training session. We plot mean values, and standard errors as error-bars; median values are indicated by stars.

### 3.2. Low complexity query

The protocol is described in Section 2.1.2. Two volunteers, in the age range of 25-35 years, participated in the study. Both participants underwent three training sessions on three consecutive days, as described above.

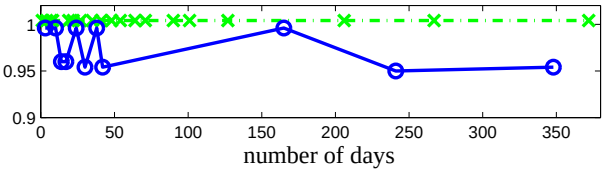


**Figure 3.** Results (success rate) of 2 participants answering low complexity queries.

With a few exceptions of isolated queries, both participants took roughly 5 seconds to conclude a query. After training, participants went through a sequence of experimental sessions including 20-25 queries each. They did not receive feedback. Results are summarized in Fig. 3. Note that after one year, one participant maintained perfect performance. Note also the large gaps (around 3 months) between the later test sessions.

### 3.3. Length of authentication

In our user study, all participants demonstrated success rate of over $95\%$ per query. In fact, many participants had perfect memory retention all the way to the last trial (we can distinguish memory errors from concentration errors in our participants' profiles), which for one participant took place almost one year after her initial training.

The number $k$ of queries needed to conclude an authentication session successfully depends on the security threshold $T$. Suppose we fix a modest security threshold of $T = 10^{-6}$, i.e., a guessing adversary will manage to pose as the legitimate user once in a million trials. Suppose also that our user answers correctly $95\%$ of the time (most participants in our study did better). The average number of queries per successful entry is approximately 11 to achieve the required threshold in the high complexity protocol (where chance of guessing is 1 in 4), and 22 in the low complexity protocol (where chance of guessing is 0.5).

In the settings used in our user study, a typical entry takes just over 3 minutes with the high complexity protocol (where each 4-choice query takes 15-20 seconds), and just over 1.5 minutes with the low complexity protocol (where each binary query takes roughly 5 seconds). Note that the high complexity protocol would take only 1.5 minutes when using the set of parameters corresponding to the 3rd row of Table 1. In fact, if lower security is acceptable, both protocols can be further shortened by allowing more bits to be revealed by the user in each query, e.g., by increasing the number of choices in the multiple choice query.

## 4. Conclusions and Discussion

We have described challenge response authentication protocols that can be used by humans, relying only on the user's natural cognitive abilities, unassisted by any external computation device. A modestly powered observer who records any feasible number of successful authentication sessions cannot recover the user's secret by the conjectured brute-force or enumeration method. Thus the protocols appear to be safe against eavesdropping, without relying on any encryption. The main drawbacks of the proposed protocols are two: users need training in a secure location, and authentication takes time as it involves a series of challenges. An interesting property of these protocols is the ability to trade-off authentication time with security, asking many questions only when high security is needed or when an attack is going on.

Our user study has an interesting implication regarding the usability of graphical passwords in general. It has been shown that user choice can reduce the entropy of chosen graphical passwords below what is considered safe [1, 7]. Our user study demonstrates that, given training, the participants can learn an arbitrary machine-generated set of pictures, and retain is for a very long period of time. This set can be used as a regular graphical password, with a sufficiently high entropy.

## References

[1] D. Davis, F. Monrose, and M. K. Reiter. On user choice in graphical password schemes. In *Proc. 13th USENIX Security Symposium*, 2004.

[2] R. Dhamija and A. Perrig. Deja vu: A user study using images for authentication. In *Proc. 9th USENIX Security Symposium*, 2000.

[3] N. J. Hopper and M. Blum. Secure human identification protocols. In *Proc. Advances in Cryptology*, 2001.

[4] S. Li and H.-Y. Shum. Secure human-computer identification (interface) systems against peeping attacks: Sechci. Cryptology ePrint Archive, Report 2005/268, 2005.

[5] T. Matsumoto. Human-computer cryptography: an attempt. In *Proc. Conf. on Computer and communications security*, pages 68 – 75, 1996.

[6] X. Suo, Y. Zhu, and G. S. Owen. Graphical passwords: A survey. In *Proc. 21st Annual Computer Security Applications Conference*, 2005.

[7] J. Thorpe and P. van Oorschot. Graphical dictionaries and the memorable space of graphical passwords. In *Proc. 13th USENIX Security Symposium*, 2004.

[8] D. Weinshall. Cognitive authentication schemes for unassisted humans, safe against spyware, HUJI TR 2006-5.

[9] D. Weinshall and S. Kirkpatrick. Passwords you'll never forget, but can't recall. In *Proc. Conf. on Computer Human Interfaces*, 2004.