Hebrew University
Leibniz Center for Research in Computer Science
TR 2006-5, 2006

Cognitive Authentication Schemes for Unassisted Humans, Safe Against Spyware

Daphna Weinshall
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Jerusalem, Israel 91904
{Email: 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 brute-force attack the protocols are safe against eavesdropping, in that an observer who fully records any feasible series of successful interactions cannot practically 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 effortlessly 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, a laptop armed with secure authentication protocols, or a simple transparency (as in "visual cryptography" [11]). 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, such as a small number of faces like in the commercial $Passface^{TM}$ System [3], abstract art pictures like in the "Deja vu" system [5], graphical passwords [20], or memorized motor sequences. 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 [10, 9], or impractical in that most humans cannot reliably use them [6]. Safety against shoulder surfing is discussed in [17], but this approach cannot protect users against eavesdropping adversaries with access to computational resources. Human factors in security systems have been discussed in a number of recent papers and workshops, including [19, 1].

In our previous work [18], we have proposed an approach which is motivated by insights from perception and cognitive psychology. Basically, we studied a variety of memory modalities including pictures [14, 15] and pseudo-words [13], taking advantage of the vast capacity of human memory to design protocols that use each memory item only once, and are thus perfectly safe against eavesdropping. These protocols have the added value (or drawback) that they cannot be "loaned" by legitimate users to other people. The main drawback of these protocols is that with extensive use they require frequent retraining of users.

This raises the problem addressed in the current paper: can we design an authentication protocol that is relatively easy for most people to use without any computational aid, that is safe against eavesdropping adversaries even after the completion of a large number of successful authentication sessions, and that does not require retraining?

In answer we propose a challenge response protocol, where authentication is based on the user answering correctly a sequence of challenges posed by the computer (cf. [6]). The challenges (or queries) are based on a shared secret between the computer and the user. Specifically, we use picture recognition as our basic cognitive primitive, since it proved most reliable in our previous study. Thus the human and the computer share a secret - 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:

- 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[1], which implies safety against dictionary attacks. This benefit is obtained at the cost of a relatively long training time, where the user is familiarized with the secret in a secure location.

[1]This is not true for user chosen alphanumeric or graphical passwords [4, 16].

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

- The interactive nature of the protocol makes 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, to conclude a series of challenge-response interactions.

The main advantage of our method over [6] 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[2]. However, unlike the protocols described in [6, 9], we are not able to provide any formal analysis proving that the protocols are indeed safe. Instead, in Section 2.4 we analyze the best brute-force and related attacks, identifying the range of parameters which make them impractical. We also simulate probabilistic attacks (see Appendix B), in order to gain some insight into their potential effectiveness against our protocols.

2 Authentication protocol

Define the following authentication scheme:

[2]Most users also seem to perceive these protocols as "fun".

- The computer assigns to each user two sets of pictures:
  - A set $\mathcal{B}$ of N common pictures. In the example of Fig. 1 the panel shows all the pictures in the set, with $N = 80$.
  - 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 was 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 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:

- M and N, the respective sizes of the secret picture subset and the common picture set, determine the security of the protocol. The larger the sets the more secure the protocol is against eavesdropping adversaries, but the longer it takes to familiarize the user with the secret.

- $n \leq N$, the number of items shown in each query, affects the security of the protocols against eavesdropping adversaries. Specifically, the mixing in the set of all secrets consistent with the observed challenge-response interaction gets larger as n increases (if the query depends on $O(n)$ presented items). We therefore expect the protocol to be more secure with larger n, but it would take the user longer to answer each query.

- $2 < P < 2^n$, the number of choices in the multiple-choice query, also affects security: the smaller P is, the larger is the size of the set of all secrets consistent with the observed challenge-response interaction. As a result the protocol is more secure for smaller P, but it requires more queries to achieve the same level of security against guessing adversaries.

- The threshold T determines the security of the protocol against random guessing. The protocol is more secure for smaller T, at the cost of longer authentication time (or larger k) because more queries must be answered correctly prior to authentication.

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 numerical 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.

In our user studies described in Section 3 we experimented with two parameter sets, trading off security and convenience:

1. In the first variant (defined in Section 2.2.1) we use $N = 80$, $M = 30$, $n = N = 80$, and $P = 4$ (see Fig. 1). The complexity of the brute-force attack is bounded by the total number of different secrets - $\binom{80}{30} \approx 2^{73}$. The query is complex, involving many of the $n = 80$ presented pictures. This implies that the set of all valid secrets consistent with a given challenge-response interaction is large. As a result, the complexity of the attack which keeps track of all the valid secrets *is not* much smaller than the complexity of the brute-force attack, and the security of this protocol is high. On the down side, the high complexity of the query implies longer authentication time.

2. In the second variant (defined in Section 2.2.2) we use $N = 240$, $M = 60$, $n = 20$, and $P = 2$. The complexity of the brute-force attack is bounded by the total number of different secrets - $\binom{240}{60} \approx 2^{190}$. The query is simple, involving only a few of the $n = 20$ presented pictures (note also that $n << N$). This implies that the set of all valid secrets consistent with a given challenge-response interaction is small. As a result, the complexity of the attack which keeps track of all the valid secrets *is* much smaller than the complexity of the brute-force attack, and the final security of this protocol is moderate. On the positive side, the low complexity of the query implies shorter authentication time.

We shall now address a few questions concerning the details of the protocols:

- How to construct a secret which is relatively easy for people to remember? Based on our previous work [18], we chose picture recognition as the basic cognitive modality, see Section 2.1.

- How to construct multiple-choice queries so that both conditions are met: (i) users find the query manageable, and can answer correctly within a short time; (ii) an eavesdropping adversary cannot practically learn the secret simply by recording successful authentication sessions. This is discussed in Section 2.2.

- How to conduct effective training? This question is discussed in Section 2.3.

- How powerful should an eavesdropping adversary be in order to break the system and discover a user's secret? in other words, can a powerful adversary be able to successfully pose as the legitimate user after observing a number of successful authentication sessions? This issue is discussed in Section 2.4, while probabilistic attacks against the system are discussed in Appendix B.

2.1 Shared secret: human factor

The shared secret between the human and the computer should be easy for people to remember, and not easy for them to give away to other people. In addition, the secret should be relatively quick to memorize

for most people, its memory trace should persist for a long time, and recognition should be relatively quick and precise. For comparison, note that the common password, which relies on the unassisted retrieval of memory items such as characters, is not very easy to memorize, rather easy to give away to unauthorized users, and the persistence of the memory trace without frequent repetition is moderate.

The first requirement would best be served by automatic (implicit) memory phenomena which require low awareness, such as procedural memory (a skill), perceptual learning or priming. However, most of the candidate phenomena from perceptual learning and priming do not satisfy the remaining requirements: for some phenomena the length of training is "unreasonably" long (e.g., [8]), for some memory persistence is too short-term, while for others the measurement of familiarity with the secret is "unacceptably" tedious [13].

We therefore chose the explicit recognition of memory items. Pictures emerged as the most promising cognitive modality, for two main reasons: large capacity - it appears like people can remember a vast amount of pictures [15], and retention - pictures can be remembered unrehearsed for a long time (years) after only a short exposure for a few seconds [2]. In [18] we have also demonstrated the feasibility of other memory modalities, such as pseudo-words, which can be used when pictures are not appropriate.

2.2 Query construction

The query is constructed with two goals in mind: First and foremost, in order to appeal to users, the query should be easy for humans to compute unassisted, and cannot therefore include complex mathematical operations. At the same time, the correct answer to the query should not reveal to an uninformed observer "too much" about the shared secret. Strictly speaking, this means that the solution to the inverse problem, i.e., given L observations of queries and answers find the mapping from query to answer, must be a hard problem with proven lower bounds on its complexity. However, very few problems have known lower bounds, and restricting ourselves to such problems is likely to miss our first goal, which is user-friendliness. We therefore require only that the protocol is secure against the brute-force attack by a computationally limited passive eavesdropping adversary (i.e., adversaries whose code breaking capability is limited to spaces with some bounded entropy), and any obvious refinement of this attack.

Specifically, 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*. In Section 2.4 we 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 (see Section 2.4). Usually in our protocols $\hat{H} < \tilde{H}$. We require that $\min(\hat{H}, \tilde{H})$ be "large enough". The threshold which defines "large enough" depends on the assumed power of the eavesdropping adversary, whom we expect to be able to defeat.

As always, there is a tradeoff between the usability of the system and its safety as measured by $\min(\hat{H}, \tilde{H})$. To quantify this tradeoff, we define two types of queries which will be used in our user study:

2.2.1 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.

Specifically, the 80 pictures are presented in a panel composed of an 8×10 regular grid (see Fig. 1). Users are asked to (mentally) compute a path, starting from the top-left picture in the panel (marked by 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.[3]

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[4] 0.25, assuming that queries are built from independent random permutations of the original set of pictures.

In this type of query, $\tilde{H} \approx 2^{73}$. In Section 2.4 we discuss simulations which provide an estimate for $\hat{H} \approx 2^{47}$, see Table 1. Thus this protocol is secure against 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 that $\hat{H} \approx 2^{56}$ (result not shown).

2.2.2 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 4×5 panel (see Fig. 2). Each picture is assigned a random bit (0 or 1), which is shown next to it, as illustrated in Fig. 2. (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. We studied two variants of possible binary queries:

1. The user should identify the first and last pictures from subset $\mathcal{F}$. He should then compare the associated bits, and answer whether they are "the same" or "different".

2. The user should 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.[5]

The second variant is a bit more difficult for users to compute, but corresponds to higher security level $\hat{H}$. This is because the computation in each query involves more pictures - 3 instead of 2.

In this type of query, with the high value chosen for N, M, $\tilde{H} \approx 2^{190}$. This provides a very high level of security against the brute-force attack, which becomes impractical for any contemporary conceivable adversary. This protocol is also not susceptible to effective enumeration attacks: the number of possible answers is exponential in the number of bits (pictures) presented n, because the answer depends on only a small number of pictures (2 or 3). However, because of this last fact this protocol is susceptible to probabilistic attacks. In the simulations discussed in Appendix B we demonstrate moderate security when 2 pictures are

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

[4] The distribution of $[0, 1, 2, 3]$ in the arrangement shown in Fig. 1 is exactly $[0.252, 0.247, 0.248, 0.253]$.

[5] The user is instructed what to do when there are less than 2 familiar pictures in the first variant, or less than 3 (but not 1) in the second variant.

Figure 2: A low complexity query panel.

used to compute the answer. Extrapolating from these simulations, we expect security with 3 pictures to be sufficient against most practical adversaries.

Note, however, that this moderate security is achieved by increasing the size of the secret as measured by N, M (compare to the high complexity query above). This, in turn, implies longer training time and lower reliability in memory retention. Thus we see another example of the trade-off between ease of use and reliability.

2.3 Training and fortifying user's memory

Training is composed of multiple sessions, with two phases:

1. *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, and then answer a multiple choice question (unrelated to the queries used in the authentication protocol) which depends on the details of the memorized picture.

2. *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

Figure 3: Change blindness example: the original picture (on the left) is shown for 400 ms, followed by a mask shown for 100 ms, followed by a modified image shown for 400 ms. The first and last pictures differ in a localized detail: one ball has disappeared in the transition between the pictures. For most people, the presence of the mask makes it very hard to notice the difference between the pictures; without the mask, the difference between the pictures pops out immediately and effortlessly.

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.

During *phase 1*, in order to solidify the memorization process and force users to pay attention to the details of the memorized picture, we took advantage of an interesting phenomenon in human attention, the *change blindness* paradigm [12]. The effect is described pictorially in Fig. 3.

For our purpose, during the construction of the picture database, each picture was assigned a matching pair, another picture which is similar to it in almost everything but a well localized (and possibly meaningful) detail as in Fig. 3. In the initial presentation of each picture, it is shown flickering with its pair, with a short blank mask in between (which should make the change hard to detect). Next, the user is shown 3 flickering pairs side by side: the original picture flickering with itself, the modified picture flickering with itself, and the original picture flickering with its modified pair. Only in the third condition is there a change between the two flickering pictures. The 3 flickering pairs are randomly arranged, and the user is asked to click on the pair where there is a change between the flickering images. If the user chooses wrongly, he is shown the original image flickering with its pair again, but **without** the interleaving mask. This makes the detection of the change immediate. The multiple choice selection is then repeated until the user gets it right.

2.4 Security against search-based attacks

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 make the unrealistic assumption that all the user's answers are correct and there is no noise in the system.

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 by roughly $\frac{1}{P^k}$. 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 which 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 data, the *enumeration attack* works by expanding 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. Specifically, $\mathcal{H}_t$ is defined by a merge operation between two sets of vectors: $\mathcal{H}_{t-1}$, and the set of all vectors that are consistent with query t and the reported answer; each vector from the first set is merged with each vector in the second set. The merge of two vectors h^1, h^2 is defined as the vector h with the following values: $h_i = h_i^1 = h_2^i$ when the elements are equal, $h_i = h_i^1$ if $h_i^2 = -1$ and vice versa. If there exists i such that $h_i^1 = 0$ and $h_i^2 = 1$ or vice versa, a contradiction is found and the merge operation fails.

From the above construction it follows that the size of the sets $\mathcal{H}_t$ initially increases quite fast with t (roughly as q^t, if q is the number of vectors consistent with one observed query). As the number of contradictions increases, eventually the size of the sets $\mathcal{H}_t$ will start 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 $\hat{H} = \max_t |\mathcal{H}_t|$.

High-complexity queries: We use simulations to compute $\hat{H}$ for the high-complexity protocol, described in Section 2.2.1. However, our computational resources do not allow us to simulate the full *enumeration attack* with sufficiently interesting values of N, M. Instead, at each time t we merge only a small sample of vectors from $\mathcal{H}_{t-1}$, and use the result of the merge operation to create a sample of $\mathcal{H}_t$ and to estimate the size of $\mathcal{H}_t$. (See further discussion in Appendix A.)

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.

The next 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 as in row 1 - 47 bits.

N	M	range	row×column	# bits $\hat{H}$	# bits $\tilde{H}$
80	30	[0-3]	8×10	47	73
120	50	[0-1]	8×10	84	114
95	40	[0-7]	8×10	47	89
145	55	[0-3]	4×5	47	135

Table 1: Mean value of $log_2(\hat{H})$ (the complexity of the *enumeration attack*) and $log_2(\tilde{H})$ (the complexity of the brute-force attack) taken over 100 runs. Simulation parameters are shown: N - the size of the common set $\mathcal{B}$, M - the size of the secret set $\mathcal{F}$, and the grid size of the query (controlled by n). The range of values in each multiple-choice query, which affects the value of $\hat{H}$ but not $\tilde{H}$, is also shown.

2.5 Discussion

The protocols designed in this paper are intended to provide secure user authentication, without using any external device, and without relying on encryption (which does not protect a user who uses a compromised computer). In addition, security should be provided for a long period of time, against limited but powerful adversaries who may intercept a very large number of successful authentication interactions.

These requirements are reflected in some of the choices made in the design of the protocols described above. In particular, the high-complexity query is characterized by a computation that relies on many bits in the user's secret, while retrieving a very small number of bits (1 or 2) in the final multiple-choice answer. This seemingly wasteful procedure is characteristic of challenge-response protocols [6, 9], and is intended to limit what an eavesdropper can learn from multiple eavesdropping sessions. This is reflected in the complexity measure $\hat{H}$ of this protocol, which is almost as high as the complexity of the brute-force attack. We chose the path finding task because it is easy and fun for people to do. Security is achieved at the expense of authentication time: each query takes roughly 15-20 seconds to conclude (for practically everyone), since many pictures must be mentally scanned.

The low-complexity query was designed to explore an alternative approach, where the computation relies on a few bits only in the user's secret. However, to obtain any security against eavesdroppers in this protocol we must increase the number of all possible secrets by increasing the size of the data-bank (the total number of pictures). Recall that users have to be familiarized with all the pictures in the data-bank, and be able to distinguish between two sub-sets of pictures in it. The larger the databank is, the harder it is for users to learn the secret, and the less reliable the secret's retention is. With a comfortable number of pictures for most people as we used above, the low-complexity protocol provided lower security. On the positive size, each query took roughly 5 seconds to conclude, since only a few pictures were scanned.

In our earlier work [18] we explored another alternative, a protocol which used a very large databank, and which did not filter the query's answer to a multiple-choice question. This protocol proved to be effective and fast to use, but security against eavesdropping was seriously limited as each secret picture could be used only once.

3 User study

We tested the two types of protocols described above in a user study that spanned long periods of time: up to 6 months with the high complexity query, and roughly 1 year with the low complexity query. Results are

described below.

3.1 High complexity query

The protocol is described in Section 2.2.1. 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. 4.

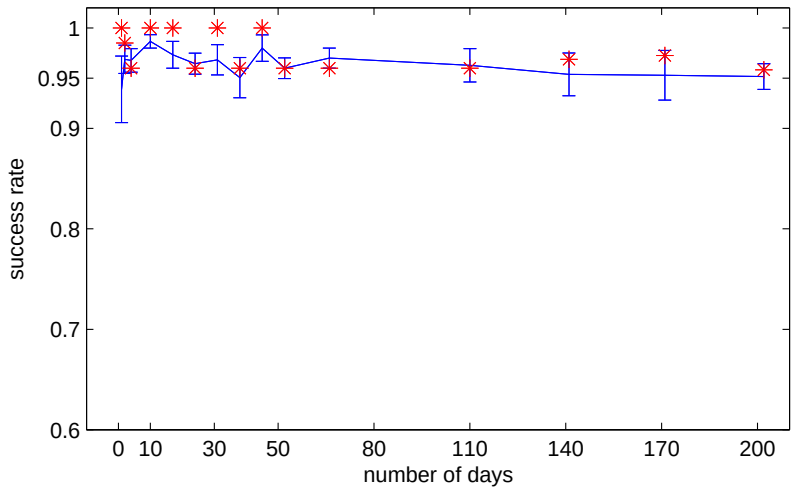

Figure 4: Results (success rate) of 9 participants answering high complexity queries, as a function of the time that has passed since the last training session. We plot mean values, and standard errors as error-bars; median values are indicated by red stars.

A question may arise whether the users, when computing the answer to the query, reveal anything about the secret? this would happen, for example, if they use their fingers to track the path, or point to the screen. When briefing the participants, we had asked them to calculate the path mentally, and not point to the screen. All participants complied with the request. When (and if) interviewed after concluding a session, participants said that the path computation was easy and intuitive, and that they did not feel the need to

track it with their fingers or point to the screen. One young user (7 years old), who did not participate in the official study but got to play around with the protocol, also expressed similar feelings about the task and its intuitive appeal.

3.2 Low complexity query

The protocol is described in Section 2.2.2. 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. The participants were asked to answer both variants of binary questions as described in Section 2.2.2. But since their performance on both questions was comparable, we report below results only for the second question, involving majority computation. 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. Participants did not receive feedback. Results are shown in Fig. 5, for a period spanning almost a full year, depending on the participants' availability. Results are summarized in Fig. 5.

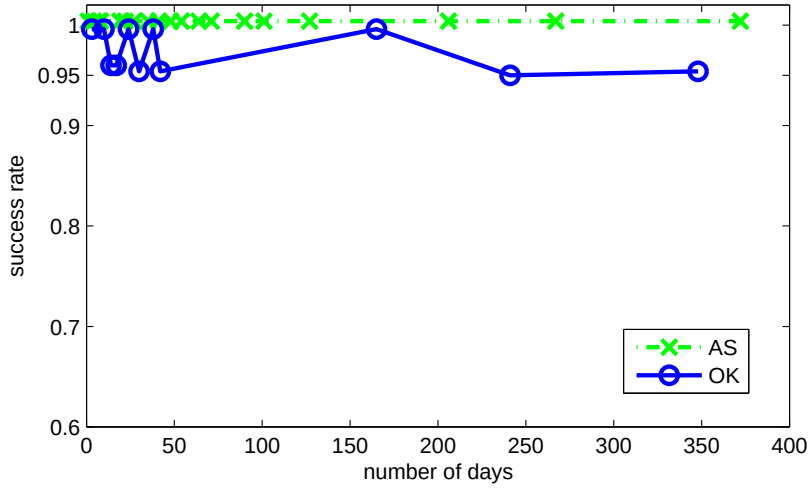

Figure 5: Results of user study with two participants, showing the average success rate in answering a set of 20-25 low complexity queries, as a function of the time that has passed since training.

3.3 The length of an authentication session

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 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. Moreover, 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.

3.4 Discussion

In our user study we did not compare our method to other methods, because very few such alternative methods exist. There is some very interesting work on visual authentication and graphical passwords in the literature (e.g. [7, 3, 5]), but those protocols are intended to improve memorability; they are *not* intended to be safe against eavesdropping, and are therefore not directly relevant to our main focus. Other protocols use external devices, such as an OTP card or transparency (e.g., [11]), but this again is beyond the scope of the present focus.

Two experimental protocols [6, 9] have been suggested with the intent of solving our problem, i.e., safety against eavesdropping with a very large number of compromised interactions, without using any external device. These protocols are hard to compare directly to our method. The method of [9] is based on the recognition of familiar pictures, and has an on-line demo in http://www.hooklee.com/SecHCI/SecHCI.asp. The method of [6] is based on an NPC problem with some knowledge of lower bounds. It is therefore provably hard to break. However, it is rather difficult to use for most people, and the reliability is moderate (as reported in one of their web-published user studies).

Note that 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 [4, 16]. 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 very high (and seemingly secure) entropy.

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. An observer who records any feasible number of successful authentication sessions cannot recover the user's secret by brute-force or any simple 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.

The proposed protocols have a few interesting properties. First, it is not easy for most users to pass their secret on to someone else, since picture recognition is only semi-implicit, and describing a large set of pictures to someone else is not straightforward. For similar reasons it is hard to steal the secret, or coerce users to give it away.

Another interesting property is the ability to trade-off authentication time with security. The number of successful answers that are needed to authenticate a user can be adapted to suit the current needs, asking many questions when high security is needed or when an attack is going on, or only a few questions when low security is sufficient.

We have chosen picture recognition in the user study described above. In future work we intend to study the usability of other modalities. In particular, we would like to investigate cognitive or procedural skills that are completely implicit, and are therefore even harder to pass on to unauthorized users. Such skills clearly exist, and the question is how to incorporate them into usable authentication protocols. Another desirable property is training time; there are cognitive modalities that can be taught in a short time (such as priming phenomena), and it remains to be seen whether they can be fit a useable authentication protocol.

Appendix

A Note on simulations

The simulations of the high-complexity protocol in Section 2.4 effectively produce an estimate for a quantity which bounds $\hat{H}$ from above. The reason is that the sample procedure fails to take into account redundancy in $\mathcal{H}_t$, where elements of the merge operation overlap, or when they overlap a path that has been shown to be impossible in some previous time. We therefore compared our estimate of $\hat{H}$ to its real value for small values of N, M, where we can compute $\hat{H}$ exactly. In this range we found that the estimation procedure gave a very good predictor of the actual value (missing by only $1 - 2\%$), as shown in Table 2.

N	M	# bits (est.)	# bits (full)
8	4	3.43	3.46
16	8	9.46	9.18
20	10	12.26	12.14
24	12	14.9	15.1

Table 2: Mean value of $log_2(\hat{H})$ taken over 100 runs, using the estimating procedure (3rd column) and the full simulation (4th column).

B Probabilistic attacks

One effective approach which may be used by an eavesdropping adversary is a probabilistic attack. We tested this approach against the low complexity protocol with same/different choice, which is most susceptible to probabilistic attacks (see discussion in Section 2.2.2). Even in this case, we show that a successful attack requires more observed sessions than seems feasible to expect, and the threat therefore appears to be minor.

Specifically, in the low complexity query with same/different choice, the answer of the user ultimately depends on two pictures only in each panel. We therefore consider all the pairs in the set $\mathcal{B}$ which totals $28,680$ for $N = 240$. Given a query and the user's answer, we can implement a voting scheme that should give on average higher scores to pairs in $\mathcal{F}$, as compared to the remaining pairs in $\mathcal{B}$. After sufficient observations (large t), we may be able to estimate reliably which pairs consistently get higher votes, from which we may be able to derive a credible guess for $\mathcal{F}$.

We have simulated this probabilistic attack as follows:

- Build a table of size $N \times N$ (initialized to 0), to represent all pairs of images.
- For each query:
  - Check if the first and last pictures in the panel are consistent with the user's answer. If not, eliminate this pair completely since it cannot be in $\mathcal{F}$ (this is an absolute final veto vote).[6]
  - For each pair in the panel (total of $\binom{n}{2}$ pairs), add a vote whose value is inversely proportional to the distance between the pictures in the panel (where distance varies between 1 and $n-1$). The sign of the vote reflects whether the pair matches the correct answer (positive sign) or not (negative sign).
- After t queries, estimate $\mathcal{F}$ using the following greedy procedure:
  - select the picture with the largest (remaining) marginal vote;[7]
  - eliminate all points which are linked to it as an impossible pair, and strengthen the total value of all the pairs which include the selected point;
  - repeat M times.
- Simulate a user who uses the estimated $\mathcal{F}$ to enter the system, and measure her success rate by the average number of correct answers.

The results of the simulated attack are shown in Table 3. Note in particular the third column p, which shows the success rate of an imposter who uses the probabilistic attack discussed above. We see that the low complexity protocol, with $|\mathcal{B}| = 240$, is only moderately secure against probabilistic attacks. This is because after observing 3000 authentication queries (the third row in the table), which correspond to at most 300 successful entries into the system (the number of queries per entry depends on the security threshold T), the adversary can pose as the legitimate user with success rate of $74 \pm 20\%$. This means that a persistent adversary can reach 94% success rate in 3 or 4 trials, which may be good enough for the system to accept him as the legitimate user (that he isn't).

The low complexity query with majority vote would require many more observations to reach the same level of performance, which may make it sufficiently safe against probabilistic attacks, given that the number of authentication sessions conducted by human users cannot exceed tens of thousands. Finally, the high complexity queries appear to be safe against such attacks.

[6] The queries were constructed such that no other pair can be eliminated with absolute certainty.

[7] Marginal vote for a point is the sum of all the votes over all the pairs which include that point, where only votes which exceed the median are considered.

$\|\mathcal{B}\|$	L	p	std
240	1000	0.52	0.03
240	2000	0.63	0.16
240	3000	0.72	0.2
300	2000	0.54	0.06

Table 3: Results of the simulated probabilistic attack. The columns are organized as follows: $|\mathcal{B}|$ - size of shared set $\mathcal{B}$, L - number or simulation runs, mean (p) and standard deviation (std) for the success rate of an imposter over many simulated attacks.

References

[1] C. Long A. Patrick and S. Flinn (Organizers). Workshop on human-computer interaction and security systems, April, 2003.

[2] B. C. Cave. Very long-lasting priming in picture naming. *Psychol. Sci.*, 8:322–325, 1997.

[3] Real User Corporation. The Science Behind Passfaces. In *http://www.realuser.com/published/ScienceBehindPassfaces.pdf*, June, 2004.

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

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

[6] N. J. Hopper and M. Blum. Secure human identification protocols. In *Proc. Advances in Cryptology*, pages 52–66, 2001.

[7] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin. The design and analysis of graphical passowrds. In *Proc. 8th USENIX Security Symposium*, 1999.

[8] A. Karni and D. Sagi. The time course of learning a visual skill. *Nature*, 365:250–252, 1993.

[9] S. Li and H.-Y. Shum. SecHCI: Secure human-computer identification (interface) systems against peeping attacks, 2003.

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

[11] M. Naor and B. Pinkas. Visual authentication and identification. In *Proc. Advances in Cryptology*, pages 322–336, 1997.

[12] R.A. Rensink, J.K. O'Regan, and J.J. Clark. On the failure to detect changes in scenes across brief interruptions. *Visual Cognition*, 7:127–145, 2000.

[13] A. Salaso, R. M. Shiffrin, and T. C. Feustel. Building permanent memory codes: Codification and repetition effects in word identification. *J Exp Psyc: General*, 114(1):50–77, 1985.

[14] R. N. Shepard. Recognition memory for words, sentences, and pictures. *J Verb Learn Verb Behav*, 6:156–163, 1967.

[15] L. Standing, J. Conezio, and R. N. Haber. Perception and memory for pictures: single trial learning of 2500 visual stimuli. *Psychol. Sci.*, 19:73–74, 1970.

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

[17] K. Richter V. Roth and R. Freidinger. A pin-entry method resilient against shoulder surfing. In *Proc. 11th ACM Conf. on Computer and cOmmunications Security*, 2004.

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

[19] A. Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proc. 9th USENIX Security Symposium*, 2000.

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