A Graphical PIN Authentication Mechanism with Applications to Smart Cards and Low-Cost Devices*

Luigi Catuogno[1] and Clemente Galdi[2]

[1] Dipartimento di Informatica ed Applicazioni, Università di Salerno
Via Ponte Don Melillo, 84084 Fisciano (SA), Italy
luicat@dia.unisa.it
[2] Dipartimento di Scienze Fisiche, Università di Napoli "Federico II"
Compl. Univ. Monte S.Angelo, Via Cinthia, 80126 Napoli (NA), Italy
clemente.galdi@unina.it

Abstract. Passwords and PINs are still the most deployed authentication mechanisms and their protection is a classical branch of research in computer security. Several password schemes, as well as more sophisticated tokens, algorithms, and protocols, have been proposed during the last years. Some proposals require dedicated devices, such as biometric sensors, whereas, others of them have high computational requirements. Graphical passwords are a promising research branch, but implementation of many proposed schemes often requires considerable resources (*e.g.,* data storage, high quality displays) making difficult their usage on small devices, like old fashioned ATM terminals, smart cards and many low-price cellular phones.

In this paper we present a graphical mechanism that handles authentication by means of a numerical PIN, that users have to type on the basis of a secret sequence of objects and a graphical challenge. The proposed scheme can be instantiated in a way to require low computation capabilities, making it also suitable for small devices with limited resources. We prove that our scheme is effective against "shoulder surfing" attacks.

1 Introduction

Passwords and PINs are still the most deployed authentication mechanism, although they suffer of relevant and well known weakness [3]. The protection of passwords is a classical branch of research in computer security. Several important improvements to the old-fashioned alphanumeric passwords, according to the context of different applications, have been proposed in the last years. Indeed, literature on authentication and passwords is huge, here we just cite Kerberos [4], S/Key [5] and OPIE [6].

* This work was partially supported by the European Union under IST FET Small/medium-scale focused research project FRONTS (Contract n. 215270).

J.A. Onieva et al. (Eds.): WISTP 2008, LNCS 5019, pp. 16–35, 2008.

Two important aspects in dealing with passwords are the following:

1. Passwords should be easy enough to be remembered but strong enough in order to avoid guessing attacks;
2. The authentication mechanism should be resilient against classical threats, like shoulder surfing attacks, i.e., the capability of recording the interaction of the user and the terminal; moreover, it should be light enough to be used also on small computers.

In the following, we describe what we call the *ATM Scenario* where the need for an authentication mechanism satisfying the above requirements becomes critical.

In the ATM Scenario the user uses a magnetic strip card to access ATM terminals. In order to be authenticated, the user pushes her card (that carries only her identification data) in the ATM reader and types her four digit PIN; afterwards, the ATM sends the user's credentials to the remote authentication server through a PSTN network. This approach is daily used by thousand of users, nevertheless it suffers from some well-known vulnerabilities. Magnetic strip cards can be easily cloned and, PIN numbers can be collected in many ways. For example, an adversary could have placed a hidden micro-camera pointing to the ATM panel somewhere in the neighborhood. A recent tampering technique is accomplished by means of a *skimmer*, i.e., a reader equipped with an EPROM memory that is glued upon the ATM reader, so that strips of passing cards can be dumped to the EPROM. A forged spotlight is also placed upon the keyboard in order to record the insertion of the PIN. The skimmer allows adversaries to collect an undefined number of user sessions obtaining all information needed to clone user cards.

Graphical passwords are a promising authentication mechanism that faces many drawbacks of old-style password/PIN based scheme. The basic idea is to ask the user to click on some predefined parts of an image displayed on the screen by the system, according to a certain sequence. Such a method has been improved during the last years, in order to obtain schemes offering enhanced security. However, the majority of proposed schemes require costly hardware (*e.g.*, medium/high resolution displays and graphic adapters, touch screen, data storage, high computational resources etc.). This makes some of the proposed schemes not suitable to be implemented on low cost equipments (*e.g.*, current ATM terminals that are still the overwhelming majority).

In this paper we propose a graphical PIN scheme based on the challenge-response paradigm that is effective to prevent "shoulder surfing" attacks. Our scheme could replace the classical PIN authentication mechanism in the two scenarios described above. The design of the scheme follows three important guidelines:

– The scheme should be *independent* from the specific set of objects that are used for the graphical challenge. In particular, our scheme can be deployed also on terminals that are equipped with small sized or cheap displays like the ones of the cellular phones, or through the classical (color and monochrome) 10 inch

CRT monitor that still equips thousands of ATM terminals. Moreover, user responses should be composed as well by any sophisticated pointing device as by simple keypad.
- The generation of challenges and the verification of user's responses should be *affordable* also by computer with limited computational resources (*e.g.* smart cards, security tokens).
- The user is simply required to *recognize* the position of some objects on the screen. She is *not required to compute* any function.

Our Results. In this paper we assume that the terminal used by the user cannot be tampered. In other words, an adversary is allowed to record the challenges displayed by the terminal and the activity of the user but she is not allowed to alter in any way the behavior of the parties. Furthermore, we assume that a sequence of three unsuccessful authentications leads to the block of the user account. This assumption is extremely common in many application scenaria, e.g., ATM. Furthermore, the adversary does not know when a legitimate user will successfully authenticate (and reset the "failure counter"). We say that an attack is successful if the adversary can "extract the user secret".

We present a strategy that can withstand shoulder surfing attacks. More precisely, in our scheme the challenge issued by the system is a random arrangement of the objects into a matrix displayed on the screen. During her authentication session, the user is required to type as PIN the position of a sequence of secret objects in the challenge matrix. Clearly, the PIN typed in by the user changes in each session as the challenge changes. To be more precise, the queries the user is required to answer are questions like *"On which row of the screen do you see object o?"*. Hence, in order to compute the correct response, the user has to watch the screen and answer some/all the questions corresponding to her secret objects, according to a given protocol.

We have experimentally evaluated the robustness of the proposed schemes against "shoulder surfing" attacks. We first analyze a naive protocol, where the user has to answer correctly to all queries of the challenge, *i.e.*, she has to compose the PIN with the digits representing the correct row number of all objects in her secret sequence.

Then, we describe two different protocols, called *user-randomized protocols*, where the user is allowed to reply the challenge issued by the server with a certain number of random or wrong answers. We show that, these randomized variations increase, w.r.t. to the naive scheme, the number of sessions the adversary needs to collect before being able to successfully extract the user secret. Following the approach presented in [2], it is possible to show a SAT-based attack.

We stress that the set of objects used to construct the challenges has an impact on the usability of the scheme. The objects used to construct the challenge should depend on the application scenario or, even better, on the users' preferences. For example, painters might be more comfortable with paintings than mathematicians that, in turn, might easily identify a sequence of numbers with specific properties. Notice that it might be even possible to use "letters"

as objects to be displayed. In this case the graphical password the user needs to remember reduces to a "classical" password.

On the other hand, complex objects cannot be displayed/managed on low-cost devices. Furthermore, the more complex are the images, the harder is the task of automatic classification that, in turn, could help the adversary in attacking the scheme.

Our scheme is independent from the specific set of objects. This makes it is suitable for deployment both on complex and simple devices and tunable on the specific application scenario.

Since our scheme requires a limited computational ability both to the user and the authenticator, following the lead of [7], our scheme could be easily deployed in those contexts where small sized devices with poor computational resources (*e.g.,* pervasive devices) are involved. In particular, our scheme could fit a RFID infrastructure as tag-to-reader and/or reader-to-tag authentication protocol within the Minimalist model defined in [8]. Moreover, our scheme could be used to enforce multi-factor authentication schemes via smartcard as card-to-reader authentication protocol. Note that even on cheapest devices, randomized protocols we present in this work could be implemented chosing set up paramenters beyond the ones affordable by human users.

2 Related Work

Identification of users through insecure channels is a classical problem in the area of computer security. One of the earliest researches on this topic is due to Lamport [9], who proposed a *one-time* password scheme, i.e., an authentication method in which the user has to prove the knowledge of the password instead of providing it. This scheme belongs to the family of *challenge and response* protocols, where the system issues a challenge to the user, who has to compute a given function of the challenge and of the secret password. The system successfully authenticates the user if the provided result is correct. The term *one-time* means that the same password can be used for several authentication rounds, but the response computed by the user is different for each round. Some implementations of the above scheme were proposed in [5,6]. The main drawback of this approach is that the user needs the help of a cryptographic device in order to compute her answer correctly. Several research has been done on defining human computable challenges [10,11,12] and evaluating the security of the resulting protocols [13,14,15].

Graphical passwords constitute a solution in this direction, since, as shown in [16], it is easier for the user to consider images instead of letters and numbers. On the other hand, since password-based identification schemes are very common, user might accept easily a letter-based password scheme in stead of a graphical one. An authentication mechanism using graphical passwords was first proposed by Blonder [17]. In his scheme an image is displayed on the screen and the user is required to click on some previously chosen regions of the image, according to a certain sequence. Images, regions and sequences of clicks are selected at user's

registration time. In the *Dèjá vu* scheme [18,19], the user, during the registration phase, is allowed to choose some images from a set of random pictures generated by the system. Later on, in order to be authenticated, the user has to recognize her pre-selected images in the set of images shown by the system. Jansen *et al.* proposed an analogous paradigm in [20,21], whereas, the "Pass-Faces" project by Real User Corp. [22] uses images of human faces instead of generic pictures. In the *Draw a Secret* scheme [23] the user is required to paint a pre-defined two-dimensional picture in the same way she did during the registration phase (that is, drawing lines and points in the same order and in the same coordinates).

Sobrado and Birget[24] proposed a scheme where, during the registration phase, the user chooses a set of small pictures (pass-icons). When the user logs in, the system shows her a screenshot populated by many different icons. In order to be authenticated, the user has to click any icon belonging to the convex-hull whose vertices are the pre-selected pass-icons. This scheme has been improved in [25].

Roth *et al.* [26] focused their attention on handling PINs of magnetic strip cards, where each PIN digit is inserted by the user in several rounds. In each round, the system shows the possible digits randomly partitioned into two sets, whose elements are depicted with a different color (*e.g.* black and white) and the user has to select the color related to the set the current digit belongs to. The intersection of sets selected at every round gives the PIN digit for the user. The security of the scheme against attacks performed by adversaries either with human memorization capabilities or with camera recording capabilities was also discussed in [26].

In the scheme presented in [1], the user and the system share a secret subset $\mathcal{F}$ of a set of public pictures $\mathcal{B}$. The authentication process is composed of several rounds. In each round the system shows to the user a table containing a picture of $\mathcal{B}$ in each cell, in a random order. The user is asked to find, across the table, a path between the image located to the top-left corner of the table and the last column or the last row of the table. The setup of this scheme is quite complicated. Users need to pass a training phase that spans over two days, and the login time can require up to some minutes. In [2] the authors present a simple attacks that breaks the scheme presented in [1]. They used information collected by observing a limited number of queries in building a system of boolean expression. Using a PC running a SAT solver [27], they are able to find the secret under the default parameters reported by [1] in 102 seconds, after collecting just 60 round transcripts.

Recently, in [28] the authors present a system that allows users to enter passwords by using the orientation of their pupils. The users input their password using gaze-based typing. Computer vision techniques are used to track the orientation of the user's pupils and to extract the password. The authors show the time needed to enter and the error rates obtained by using their system is comparable with the ones obtained by using a keyboard. Furthermore, the users tend to prefer the use of gaze-based systems in place of classical password/PIN-entry methods. On the other hand, such scheme requires costly hardware since image

analysis has to be executed in an on-line fashion, i.e., while the user is "typing in" the password.

For a wider overview about research on graphical passwords, we suggest the reader to take a look at the survey by Suo *et al.* [29] and visit the web site of the "Graphical Passwords Project"[30] at Rutgers.

3 Preliminaries

In this section we introduce the notation and conventions used in the rest of the paper.

Objects and Challenges. The protocols described in this paper belong to the family of *challenge and response* authentication schemes, where the system issues a random challenge to the user, who is required to compute a response, according to the challenge and to a secret shared between the user and the system. In particular, the challenges consist of random pictures containing several objects. We denote by O the set of all distinct objects and by $q = ak$, for some positive integers a and k, its cardinality. A *challenge* is a sequence $\alpha = (o_1, \ldots, o_q)$, where o_i is an object drawn from O. The objects in α are arranged in a matrix with a rows and $k = q/a$ columns.

Secrets Description. In our protocols the secret is a sequence of m objects $\sigma = (\sigma_1, \ldots, \sigma_m)$. The authentication protocol consists of m questions, called *queries*. The i-th query is a question of the following type: *"On which row of the screen do you see the object σ_i?"*. Since questions are chosen independently, the set of possible queries has size $|O|^m$. Since the m objects in the secret are chosen independently, the set of possible secrets has size $|O|^m$.

Responses and Session Transcripts. Upon reception of a challenge, the user is required to compute a response, according to the secret queries shared with the system. A response is a vector $\beta = (\rho_1, \ldots, \rho_m)$, where each ρ_i is a number drawn from a set $A = \{0, 1, \ldots a-1\}$, representing the answer to the i-th query, according to the challenge. A Session Transcript is a pair $\tau = (\alpha, \beta)$, where α is a challenge and β is the user response to α.

4 A Naive Protocol

In this section we describe a first protocol allowing a user U to authenticate herself to a terminal T. We assume U and T share a sequence $\sigma = (\sigma_1, \ldots, \sigma_m)$ of m queries. Furthermore, the user U is provided with a token (*e.g.*, a smart-card), carrying all the information needed to identify U, (*e.g.,* U's account number).

Upon insertion of the token into the terminal, the terminal constructs a challenge α by partitioning the set O of possible objects into a (disjoint) sets $Q_1, \ldots, Q_a$, corresponding to distinct rows displayed on the screen, such that each row contains exactly q/a objects, i.e., $|Q_i| = q/a$, for $i = 1, \ldots, a$. Notice that a denotes the number of possible answers to each query, i.e., the cardinality of the set A.

The introduction of the set A as the set of possible answers is due to the practical idea that users' answers should not be complex to be computed. In other words, in order for the system to be usable, the user should not be forced to search an object in a set with many elements before computing the correct answer.

The use of the set A allows us to restrict the set of possible answers from $\{0,\ldots,9\}^m$, to $\{1,\ldots,a\}^m$, where $a < 10$. Under this assumption, in order to avoid the possibility for the adversary to randomly guess the array β, the number of queries should be sufficiently large. For example, in order to have an answer space containing at least 10000 elements, the number of digits the user should type, when $a = 4$, is at least 7, since $a^7 = 16834$.

The user U, on input the challenge α is required to compute her response $\beta = (\rho_1,\ldots,\rho_m)$, where ρ_i is the answer to the i-th query "On which row is the object σ_i displayed?". The user passes the authentication test if *all* answers in β are correct.

We note that the authentication consists of a *single* round. The terminal T displays a single challenge and the user replies with m integers drawn from A.

Blind Attack. The first attack we consider to the above protocol is the *blind attack*, where the adversary simply tries to guess the correct answer to a random challenge, without any knowledge of previous authentication transcripts. Clearly, the success probability of such an adversary is $1/a^m$, since there are a^m possible answers, i.e., a answers for each one of the m queries.

The Recording Attack. We now consider the case in which the attacker has the chance to control the terminal T, by recording a certain (finite) number of authentication transcripts from successful sessions carried out by the user U. We also assume that the adversary cannot tamper T. In other words, the adversary can (a) read the information contained on the token; (b) read the challenge issued by the terminal T and (c) read the User reply to the challenge. The adversary cannot actively interfere with the authentication process and, in particular, she can neither (a) modify the challenge presented to the user nor (b) modify the user's answer.

In order to evaluate the robustness of the proposed scheme, we assume that the goal of the adversary is to extract the user secret given a certain number of transcripts[1]. Recall that a sequence of three unsuccessful authentications leads to the block of the user account. For this reason, since the naive protocol authenticates the user only if she correctly replies to *all* the queries in the challenge, we consider the extraction of the secret a necessary condition for the adversary to impersonate the user with probability 1.

We have experimentally evaluated the robustness of the proposed protocol. The simulations we have carried out aim at identifying the *average number* of transcripts that the adversary needs in order to correctly extract the user secret

[1] As we will see in the next section, secret extraction is not the only possible goal for the adversary.

as a function of (a) the number of objects q used to construct the queries; (b) the number of rows a used to partition the objects and (c) the number of objects m in the user's secret.

Let $\beta = (\rho_1, \ldots, \rho_m)$. For each object o_j in the set of objects we keep m counters, denoted by $w_{j,1}, \ldots, w_{j,m}$, one for each component in the user secret. An object o_j is said to be a *candidate* for the i-th component of the secret if $w_{j,i} = \max_{o_l \in O} w_{l,i}$. In other words, an object is a candidate for the i-th component if its i-th counter has the maximum value among the counters for the the specific component.

Since challenges are randomly constructed, the average is computed over 10000 executions of the following experiment:

- A user secret is uniformly selected among the $|O|^m$ possible secrets.
- The adversary requires as many transcripts (α, β), where $\alpha = (Q_1, \ldots, Q_a)$ and $\beta = (\rho_1, \ldots, \rho_m))$, she needs to extract the user secret. The i-th counter associated to object o_j, $w_{j,i}$, is incremented if o_j belongs to the row identified by the answer to the i-th query, i.e., $o_j \in Q_{\rho_i}$.
- The process terminates when *for the first time* there exists, for each component in the secret, exactly one candidate object.

Intuitively, the above process identifies the user secret because each answer to the challenge is always correct. For this reason, after the analysis of k transcripts, the counters associated to each object in the user secret will have value k, i.e., each such object will be a candidate for its component. On the other hand, because of the randomized nature of the challenge creation process, as k grows all the objects that do not belong to the user secret will, eventually, have a counter whose value is strictly less than k.

In Figure 1 we report the average number of transcripts needed to extract the secret when the number of objects in the user secret is $m = 15$, the total number of objects $q \in \{20, 60, 80, 100\}$ and the number of rows used to partition the objects a belongs to $\{2, 4, 5, 10\}$.

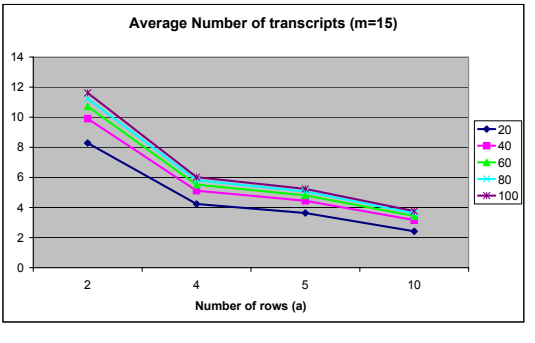

Fig. 1. Strategy Naive: Average number of transcripts needed to extract the secret with $m = 15$

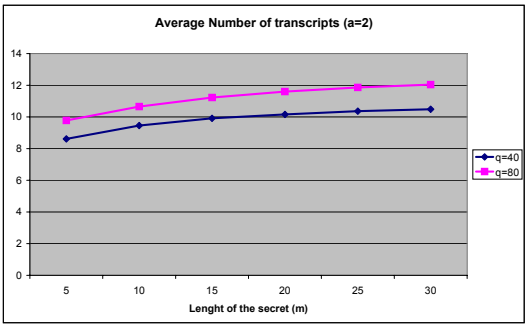

Fig. 2. Strategy Naive: Average number of transcripts needed to extract the secret with variable length secrets

It can be seen that whenever the value of a increases, the average number of required transcript drops quickly. Intuitively, this is due to the fact that the bigger is the value of a, the smaller is the number of objects on each row and, thus, the higher is the information gained by the adversary for each transcript.

In Figure 2 we report the dependence of the average number of required transcripts and the length of the user secret when $a = 2$. As expected, the longer is the secret, the bigger is the number transcripts needed to extract the user secret. However, the number of required transcripts grows slowly w.r.t. the length of the secret. If we consider $q = 40$ and $a = 2$, the average number of transcripts needed to extract a secret containing $m = 10$ objects is slightly less than 10, while the corresponding value for $m = 30$ is slightly higher than 10.

The above discussion shows that, on one hand, the values of m can be low enough to guarantee usability. On the other hand, m cannot be too small in order to prevent blind attacks.

The above experimental evaluation allow to define the following scenaria:

- Small-sized displays, lower security: $q = 40, a = 2, m = 10$. In this case, the probability of a blind attack is $9.7 \cdot 10^{-4}$. The average (resp., minimum) number of transcripts (over 10,000 experiments) the adversary needs to collect before being able to extract is 9.45 (resp., 6).
- Bigger displays, higher security: $q = 80, a = 2, m = 15$. In this case, the probability of a blind attack is $3 \cdot 10^{-5}$. The average (resp., minimum) number of transcripts (over 10,000 experiments) the adversary needs to collect before being able to extract 11.23 (resp., 7).

5 User Randomized Protocols

In this section we explore the possibility that the user herself randomizes the protocol. In other words, the user is allowed to give either random or wrong answers to some randomly chosen queries. It is immediate that the efficiency of these two

strategies is different since, intuitively, a random answer does not reveal any information about user's secret while a wrong one does. Furthermore, an adversary always "guesses" a random answer, but it may fail in guessing a "wrong" answer. Thus one basic difference between these two strategies: If the user is allowed to give *wrong* answers (as opposed to *random* ones), we can require as an acceptance criterion that *exactly* c out of m answers are correct *and* that *exactly* $r = m - c$ out of m are wrong (as opposed to *at least* c correct answers out of m.) Clearly, the "correct-random" strategy should be easier to attack.

Notice that user randomization slightly modifies the goal of the adversary. Indeed, in this case, the adversary it is not required anymore to *completely* extract the user secret. An attack is successful if it manages to extract a sequence of objects that can be used for the authentication. The extracted sequence will certainly contain some components of the user secret but it may also contain some objects that do not belong to it.

It is immediate that the success probability of a blind attack for randomized protocols is greater that the corresponding probability for the naive deterministic protocol with the same parameters. For this reason, we have to carefully consider such success probability in order to avoid situations in which it is difficult for the adversary to extract a sequence of objects that allow the authentication but, at the same time, it is easy to be successful using a blind attack.

Although it is well known that, for various reasons, humans are not good random generators, we will still assume that a user can randomly select objects for the following reasons: (a) If users are well-trained and informed about the consequences of their misbehavior, they will actually try to select objects randomly instead of deterministically; (b) our scheme is also applicable for device authentication, i.e., in a non-human context.

5.1 Correct and Random Answers

Let $1 \leq c \leq m$ be an integer. The user randomly selects c out of the m queries and gives correct answers only to these queries while returns random answers to the remaining $r = m - c$ ones. Clearly, if $c = m$ the protocol is the one presented in the previous section.

Blind Attack. Let us consider the success probability of a blind attack. First of all we notice that the maximum number of random answer depends on the value of a. Indeed, let σ be a secret and let $(\alpha, \overline{\beta})$ be a transcript in which all the answers in $\overline{\beta}$ are correct. If the adversary constructs β by randomly picking values in the range $\{1, \ldots, a\}$, the expected number of components in β that will be equal to the corresponding component in $\overline{\beta}$ is m/a. Thus the adversary will be able to correctly guess m/a components of the reply. Since the authentication criterium is "β contains at least c correct answers", if we let $r > m/a$, the adversary will be able to successfully authenticate w.h.p. For this reason, we will only consider values for r that are strictly less than m/a.

Algorithm Authenticate(O, a, c)

1. T constructs a challenge α by randomly partitioning O into a sets $Q_1, \ldots, Q_a$ such that $|Q_i| = q/a$, for $i = 1, \ldots, a$, and displays it on the screen.
2. U computes her response $\beta = (\beta_1, \ldots, \beta_m)$ by correctly answering to c queries and giving random answers to the remaining ones. U sends β back to T.
3. T authenticates U if *at least* c answers are correct.

Fig. 3. An improved authentication protocol

Since the user is required to correctly answer c queries, while she is allowed to give *random* answers to the remaining $r = m - c$ queries, the success probability of a blind attack in this case is $\sum_{h=c}^{m} \binom{m}{h} 1/a^h (1 - 1/a)^{m-h}$.

The Recording Attack. Recall that the goal of the adversary is to obtain a sequence of objects that can be used for successfully authenticate to the terminal. Thus, if the authentication protocol allows the user to reply using $r = m - c$ out of m random answers, it is enough that the adversary manages to correctly extract *at least* c components of the secret. Such set of objects is enough to fulfill her goal.

Notice that the strategy used to extract the user secret presented in the previous section does not work with the randomized authentication strategy. Indeed, in the randomized case, the extraction process cannot stop "the first time there exists, for each component in the secret, exactly one candidate". Intuitively, since user answers are randomized, if at a certain time there exists a single candidate for a given component, such candidate might be different from the actual component in the secret.

For this reason, we have slightly modified the attack strategy. Instead of allowing the adversary to obtain as many transcripts she needs, we provide her t transcripts $(\alpha_1, \beta_1), \ldots, (\alpha_t, \beta_t)$. As in the previous case, the adversary counts the number of times each object belongs to the row identified by the user answers. After t transcripts it may be the case that for some components the adversary has identified more than one candidate, i.e., there exist at least 2 objects whose counter for the specific component has the maximum value. In this case we randomly pick one of these objects as actual candidate. If, instead, for each component there exists exactly one candidate, the following cases may arise:

- All candidates are correct. The adversary has correctly extracted the whole user secret. We call such sequences of objects *good*.
- The number of correct candidates belongs to $\{c, \ldots, m - 1\}$. The user secret has *not* been correctly extracted but the sequence of objects is a valid authentication secret. We call such sequences of objects *valid*.
- The number of correct candidates is strictly less than c. We call such sequences of objects *wrong*.

We assume that the adversary is successful if she manages to extract either a good or a valid secret.

We have first analyzed the dependence of the *sum* of the number of good and valid sequences w.r.t. the number of random answers allowed by the scheme when (a) q belongs to the set $\{20, 40, 60, 80, 100\}$, (b) $a = 2$, i.e., the q objects are partitioned into two sets, (c) $m = 15$, i.e., user secret consists of 15 objects and (d) the adversary is provided with $t = 15$ transcripts. Similar results can be obtained using different parameters. Since, as stated above, the maximum number r of random answer has to be strictly less than $m/2$, we consider the case in which r belongs to the set $\{0, \ldots, m/3 = 5\}$.

From our experiments we can derive that, even if the number transcripts provided to the adversary is "high"[2], as the number of random answers increases, the number of good (resp., good and valid) secrets decreases quickly. Furthermore, in some cases, the adversary is not even able to extract a valid secret out of the given transcripts.

At this point we have considered the case in which q is fixed to 80 and we let the value of a to belong to the set $\{2, 4, 8, 10\}$. Notice that, since a is not constant, also the maximum number of random answers varies depending on a. Also in this case the length of the secret consists of $m = 15$ objects and we have provided the adversary with $t = 15$ transcripts.

From the results of the experiments we can derive that the adversary's probability of extracting good secret increases very quickly as the value of a increases. On the other hand, if we consider both good and valid secrets, the probability of success of the adversary is extremely high. For the specific set of parameters, the adversary may fail in extracting a good or a valid secret only if $a = 2$.

Finally we have evaluated the success probability of the adversary when the number of transcripts t provided increases from 10 to 30. As expected, as the number of transcripts given to the adversary increases, the probability of extracting a good or a valid secret increases. Notice that if the number of random answers allowed by the scheme increases, the success probability of the adversary decreases. Unfortunately if we set $r = m/3$, the success probability p of a blind attack becomes high, ($p = 0.15$). If we set $r < m/3$, e.g., $r = 4$ in our example, the probability of success of a blind attack decreases to 0.059 while the authentication scheme is still resilient to an adversary that can collect up to 15 transcripts without being able to extract neither a good nor a valid sequence.

5.2 Correct and Wrong Answers

In the previous section we have analyzed the case in which the user has to give correct and random answers. We now consider the case in which the user can alternate correct and *wrong* answer. As stated above, we assume that the user is required to answer to each query with *exactly* c out of m correct answers and *exactly* $r = m - c$ out of m wrong ones.

[2] Recall that the average number of transcripts needed to correctly extract the user secret with $q = 80$, $a = 2$ and $m = 15$ using the Naive strategy is slightly higher that 10.

Blind Attack. In this case, if the user is required to answer c correct queries and give wrong answers to the remaining $r = m - c$, the success probability of a blind attack is $\binom{m}{c}(1/a)^c(1-1/a)^{m-c}$.

Recall that in the Correct-random strategy, the number of random answers cannot be to high. Indeed, if for example $r = m$, the adversary has probability 1 of being successful in a blind attack.

For this strategy, such limitation does not apply. Indeed the adversary needs to guess *exactly c* correct answers out of m as opposed to *at least c* for the correct-random case. For this reason the value of c can range from 0 to m. Clearly the success probability of a blind attack is maximized when $c = m/2$. However, in this case, such probability is never equal to one.

The Recording Attack. We have experimentally verified this strategy using the same approach we have used the same approach described in Section 5.1.

We have first analyzed the case in which q belongs to the set $\{20, 40, 60, 80, 100\}$, $a = 2$, i.e., the q objects are partitioned into two sets, $m = 15$, i.e., user secret consists of 15 objects. Similar results are obtained with different sets of parameters. For such experiments, and the adversary is provided with $t = 40$ transcripts. The number r of wrong answers required to the user ranges in $\{0, 1, \ldots, 7\}$. Surprisingly, the behavior of the success probability does not strongly depend on the number of objects.

Figure 4 show the percentage of success of the adversary in extracting good or valid secrets from the transcripts. The different curves, each describing a different value of q, are very close to each other.

We have then analyzed the case in which q is fixed to 80 and the value of a belongs to the set $\{2, 4, 8, 10\}$ while keeping the values and ranges of the remaining parameters as in the previous set of experiments. Figure 5 reports the percentage of success when the value of q is fixed to 80 and $m = 15$. In this case, as expected, the lower is the value of a, the lower is the percentage of success of the adversary.

We have also analyzed the dependence of the success probability of the adversary w.r.t. the number of transcripts provided. As expected, the higher is the number of transcript, the higher is the success probability of the adversary. Furthermore, as the number of wrong answers required by the scheme grows from 0 to $m/2$, the the adversary's probability of success decreases.

The case $a = 2$ has an interesting property. Assume that the number of errors required by the scheme is $m/a = m/2$. We can restate the previous statement as "for each i, the user answers correctly i-th query with probability $1/2$". In our setting, this implies that the counter associated to the i-th object of the user secret is incremented, at each transcript, with probability $1/2$. Now notice that this is (approximately) the same probability with which the the i-th counter of any other object is incremented.

This means that the frequencies with which the objects are selected by the user are more or less the same and, thus, the user secret cannot be identified by using the counters approach. The impossibility of using the attack technique described so far is due to the fact that counters associated to each object only

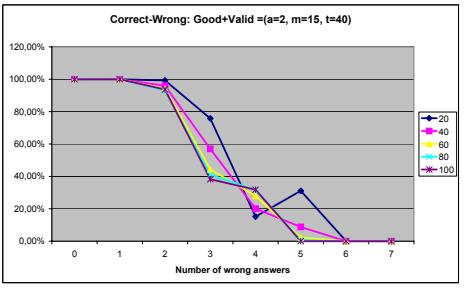

Fig. 4. Strategy Correct-Wrong. The adversary is provided with 40 transcripts. Percentage of good and valid secrets extracted as function of the number of wrong answers with $a = 2, m = 15$ and different values of q.

consider the occurrence of each object independently for each component of the secret. In other words, the attack strategy does not consider the fact that in each transcript there are *exactly* c correct answers and $m - c$ wrong ones. As the number of wrong answers approaches to $m/2$, the number of transcripts needed to extract either a good or a valid secret increases. If such number is approximately $m/2$, an attack that uses a counting argument cannot extract neither a good nor a valid secret, even if the adversary is provided with an extremely high number of transcripts. Such arguments are supported by the results of the experiments. Indeed, Figure 6 (resp., Figure 7) shows that when the number of wrong answers is approximately $m/2$, even if the adversary is provided with 100 transcripts, she cannot even extract a good (resp., a valid) sequence of objects.

Unfortunately the value $r = m/2$ cannot be used in practice since the success probability of a blind attack in this case is high. For example if $m = 15, a = 2$ and $r = 8$, the probability of a blind attack is equal to 0.19. If we reduce the value of r to 5, the probability of success of a blind attack decreases to 0.09. Since we assume that three unsuccessful authentication trials lead to the block of the user account, such set of parameters may be satisfactory in some application scenaria. On the other hand, the latter set of parameters is resilient to an adversary that allows the adversary to collet up to 36 transcripts.

5.3 Possible Extensions and a SAT-Based Attack

The authentication strategies presented in this paper guarantee that the adversary cannot extract a good or a valid sequence of object given a certain number of transcripts.

In the "Correct-Wrong" strategy the number of required transcript can be as high as 36. We have argued and experimentally verified that if the number of answers the user is allowed to give to each challenge may increase to $m/2$, the adversary might not be able to extract a valid transcript using the attack

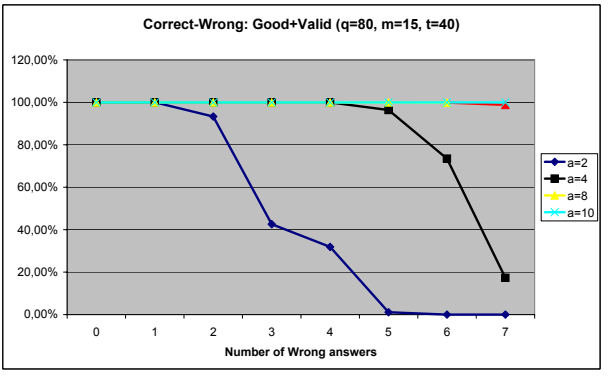

Fig. 5. Strategy Correct-Wrong. The adversary is provided with 40 transcripts. Percentage of good and valid secrets extracted as function of the number of wrong answers with $q = 80, m = 15$ for different value of a.

strategy presented so far. Unfortunately such parameter setting cannot be used because of the high probability of success for a blind attack.

On the other hand we may require the user to answer correctly to a *specific* set of answers (instead of *any set containing exactly c correct answers*). Clearly, the required set of correct answer need to change for each challenge, otherwise the adversary will immediately identify the components of the user secret that always correspond to the correct answers. In this case we have the following side effects:

- The success probability of a blind attack decreases to $(1/a)^c(1-1/a)^{m-c} = 1/2^m$;
- The length of the user secret decreases; In this case, the length of the secret can be safely decreased to 10.
- The user needs to remember the specific set of objects to which she has to answer correctly. Clearly this makes the user secret longer. We may circumvent this problem by providing the user with a specific hardware device that provides, at each authentication, a different set of answers to which the user has to answer correctly. We notice that such tokens are already used for providing one-time PINs. However, we notice that if the token is used to provide the one-time PIN "in clear", an adversary that steals the token can easily impersonate the legitimate user. In our case, the mere possession of the device still does not allow the adversary to authenticate without the knowledge of the user secret. Thus the user secret still plays a central role in the multi-modal authentication scheme. We stress that in a recording attack, the adversary is not allowed to read the token.

Under the above assumptions, it is possible to consider the setting in which $a = 2$, the user secret consists of m objects and the number of correct answers is $m/2$. As argued in the previous section, in this case the adversary cannot use

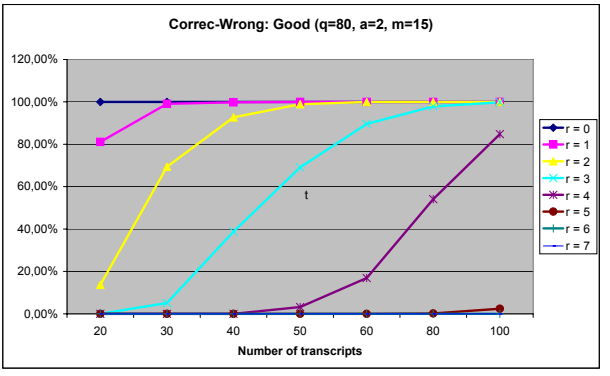

Fig. 6. Strategy Correct-Wrong (m=15, q=80, a=2). The adversary is provided with a number of transcripts in the range $\{20, 30, 40, 50, 60, 70, 80, 90, 100\}$. Percentage of good secrets extracted.

the attack technique described so far to extract good or valid transcripts. On the other hand, we can use the same technique presented in [2] to extract the user secret.

Although we focus on the Correct-wrong strategy, we show that the attack can be used also for the other authentication strategies presented in the paper.

Let us denote by $\alpha(k)$ the challenge for the the k-th transcript and let $\beta^{(k)}$ be the corresponding response. Since $a = 2$, $\alpha^{(k)}$ is a matrix consisting of 2 rows and $p = q/2$ columns. Let $(i_1^{(k)}, \ldots, i_p^{(k)})$ (resp., $(i_{p+1}^{(k)}, \ldots, i_q^{(k)})$) be the first (resp. the second) row of $\alpha^{(k)}$. In order to simplify the notation, we will omit the transcript number k when it is clear from the context.

We assign m different boolean variable $x_{i,1}, \ldots, x_{i,m}$ to each object o_i, with $i = 1, \ldots, q$. Intuitively, $x_{i,j} = 1$ implies that the j-th component of the user secret is o_i. Since each o_j appears in α exactly once, for every i, the i-th component of the user secret belongs either to the first or to the second row of α. For every $t = 1, \ldots, m$, i.e., for every component of the user secret, we define $\phi_{0,t} = x_{i_1,t} \vee \ldots \vee x_{i_p,t}$ and $\phi_{1,t} = x_{i_{p+1},t} \vee \ldots \vee x_{i_q,t}$

The adversary does know the user response $\beta = (\beta_1, \ldots, \beta_m)$, but she does not know which component of the response is correct and which one is wrong. On the other hand, the adversary knows that exactly $m/2$ out of m answers are correct. Let us define $A_m = \{\overline{a} = (a_1, \ldots, a_m) \in \{0,1\}^m | w(\overline{a}) = m/2\}$, where $w(\cdot)$ denotes the Hamming weight of $\overline{a}$. Intuitively, if $a_i = 0$, the i-th answer contained in β is correct, otherwise is wrong.

Given the above notation, we can state that the following formula is satisfiable:

$$\psi = \bigvee_{(a_1,\ldots,a_m)\in A_m} \bigwedge_{j=1}^{m} (\phi_{\beta_j \oplus a_j, j} \wedge \neg \phi_{(1-\beta_j)\oplus a_j, j}). \tag{1}$$

Intuitively, the satisfiability of the above formula follows from the observation that: For a generic transcript (α, β) there exists a boolean array $(a_1, \ldots, a_m)$ that

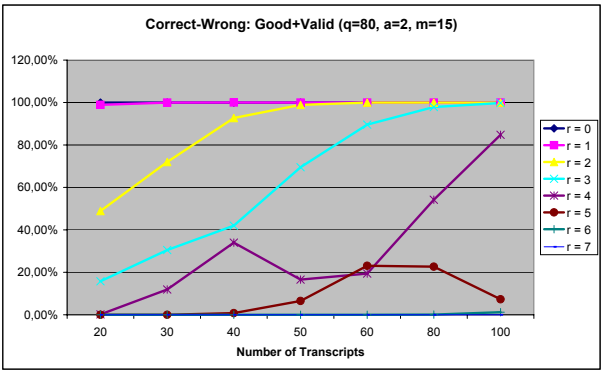

Fig. 7. Strategy Correct-Wrong (m=15, q=80, a=2). The adversary is provided with a number of transcripts in the range $\{20, 30, 40, 50, 60, 70, 80, 90, 100\}$. Percentage of good and valid secrets extracted.

identifies the correct and wrong answers. If the j-th answer in β is correct, i.e., $a_j = 0$, then the j-th component in the user secret belongs to the row identified by β_j (and, obviously, does not belong the the row identified by $1 - \beta_j$). Similar arguments apply for $a_j = 1$.

If the adversary is provided with t transcripts, the above formula has to be satisfied for each transcript. For this reason, if we denote by $\psi^{(k)}$ the Formula (1), properly rewritten for the k-th transcript, the following formula is satisfiable: $\psi' = \bigwedge_{k=1}^{t} \psi^{(k)}$.

Notice that the number of variables $x_{i,j}$ does *not* depend on the number of transcripts, i.e, all the formulas $\psi^{(k)}$ are written using the same variables.

The last constraint we need to consider is the fact that, each component of the secret consists of exactly one object. The above statement can be expressed by the following:

$$\epsilon = \bigwedge_{j=1}^{m} \bigvee_{i=1}^{q} (\neg x_{1,j} \wedge \ldots \wedge \neg x_{i-1,j} \wedge x_{i,j} \wedge \neg x_{i+1,j} \wedge \ldots \wedge \neg x_{q,j})$$

For any possible sequence of successful transcripts $((\alpha_1, \beta_1), \ldots, (\alpha_t, \beta_t))$ and for any possible secret σ, the formula $\mu = \epsilon \wedge \psi'$ is satisfiable. Notice that a truth assignment for μ might not represent the actual user secret. As an example, consider the case in which the adversary only holds a single transcript. Clearly the formula μ is satisfiable also in this case but there might exists multiple truth assignments. Clearly as the number of transcripts held by the adversary increases, the number of possible truth assignment for μ converges to 1, i.e., the actual user secret.

The attack just described can be easily modified for the Naive and the Correct-Random authentication strategies. In the former case, since all answers are correct, it is enough to consider the set $A_m = \{(0, 0, \ldots, 0)\}$. In the latter case, if r

is the number of random answers allowed by the scheme, the set A_m should be defined as: $A_m = \{\overline{a} = (a_1, \ldots, a_m) \in \{0,1\}^m | w(\overline{a}) \leq r\}$.

Currently we are implementing a test environment to experimentally evaluate the resiliency of the scheme presented w.r.t. this attack.

6 Conclusion and Future Work

In this paper we have presented a simple graphical PIN authentication mechanism that is resilient against shoulder surfing attacks. Our scheme is independent on the specific set of objects used to construct the challenges. The scheme may be implemented on low cost devices, does not require any special training for the users and requires a single round of interaction between the user and the terminal. We have argued that a secret consisting of 15 objects, e.g. letters, is enough to prevent the adversary to successfully authenticate even if the manages to obtain 36 transcripts.

The presented scheme can be also used for low-cost device authentication, e.g., RFID tag-to-reader or reader-to-tag authentication.

A number of extensions are possible for our scheme. An interesting variations is to authenticate the user if she answers correctly to a *specific* set of answers. Furthermore, is it possible to design a scheme in which the adversary manages to extract the user's secret *only if* she obtains a sequence of *consecutive* authentications? In the presented scheme, the adversary simply needs to obtain *any* sufficiently long sequence of authentications. If it should be possible to bind the secret extraction to the consecutiveness of the collected transcripts, in the real world the adversary may have very few chances of being successful.

Finally, we are currently experimentally evaluating the resilience of our scheme w.r.t. the SAT-based attack.

Acknowledgements

The authors thank Gene Itkis and Pino Persiano for their useful comments and suggestions.

References

1. Weinshall, D.: Cognitive authentication schemes safe against spyware (short paper). In: IEEE Symposium on Security and Privacy, pp. 295–300. IEEE Computer Society, Los Alamitos (2006)
2. Golle, P., Wagner, D.: Cryptanalysis of a cognitive authentication scheme (extended abstract). In: IEEE Symposium on Security and Privacy, pp. 66–70. IEEE Computer Society, Los Alamitos (2007)
3. Anderson, R.J.: Why cryptosystems fail. Commun. ACM 37, 32–40 (1994)
4. Steiner, J.G., Neuman, B.C., Schiller, J.I.: Kerberos: An authentication service for open network systems. In: USENIX Winter, pp. 191–202 (1988)

5. Haller, N.M.: The S/KEY one-time password system. In: Proceedings of the Symposium on Network and Distributed System Security, pp. 151–157 (1994)
6. McDonald, D.L., Atkinson, R.J., Metz, C.: One time passwords in everything (OPIE): Experiences with building and using stronger authentication. In: Fifth USENIX UNIX Security Symposium, Salt Lake City, Utah(USA) (1995)
7. Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
8. Juels, A.: Minimalist cryptography for low-cost rfid tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 149–164. Springer, Heidelberg (2005)
9. Lamport, L.: Password authentification with insecure communication. Commun. ACM 24, 770–772 (1981)
10. Matsumoto, T., Imai, H.: Human Identification through Insecure Channel. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 409–421. Springer, Heidelberg (1991)
11. Wang, C.H., Hwang, T., Tsai, J.J.: On the Matsumoto and Imai's Human Identification Scheme. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 382–392. Springer, Heidelberg (1995)
12. Matsumoto, T.: Human-computer cryptography: An attempt. In: ACM Conference on Computer and Communications Security, pp. 68–75 (1996)
13. Hopper, N.J., Blum, M.: A Secure Human-Computer Authentication Scheme. In: Carnagie Mellon University Technical Report. Vol. CMU-CS-00-139 (2000)
14. Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)
15. Katz, J., Shin, J.S.: Parallel and Concurrent Security of the HB and HB$^+$ Protocols. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 73–87. Springer, Heidelberg (2006)
16. Grady, C.L., Mcintosh, A.R., Rajah, M.N., Craik, F.I.M.: Neural correlates of the episodic encoding of pictures and words. Proc. Natl. Acad. Sci. USA 95, 2703–2708 (1998)
17. Blonder, G.E.: Graphical passwords. Lucent Technologies Inc, Murray Hill, NJ (US), US Patent no. 5559961 (1996)
18. Perrig, A., Song, D.: Hash visualization: A new technique to improve real-world security. In: Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (1999)
19. Dhamija, R., Perring, A.: Déjà vu: A user study using images for authentication. In: IX USENIX UNIX Security Symposium, Denver, Colorado (2000)
20. Jensen, W., Gavrila, S., Korolev, V., Ayers, R., Swanstrom, R.: Picture password: a visual login technique for mobile devices. In: National Institute of Standards and Technologies Interagency Report, vol. NISTIR 7030 (2003)
21. Jensen, W.: Authenticating users on handheld devices. In: Proceedings of Canadian Information Technology Security Symposium (2003)
22. Real User Coorp.: Pass faces (1998), `http://www.realuser.com`
23. Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., Rubin, A.D.: The design and analysis of graphical passwords. In: Proceedings of the 8th USENIX security Symposium, Washington DC (1999)
24. Sobrado, L., Birget, J.C.: Graphical password. The Rutgers Scholar, an electronic Bulletin for undergraduate research 4 (2002)
25. Wiedenbeck, S., Waters, J., Sobrado, L., Birget, J.C.: Design and evaluation of a shoulder-surfing resistant graphical password scheme. In: Proceedings of Advanced Visual Interfaces AVI 2006, Venice, ACM Press, New York, NY (2006)

26. Roth, V., Richter, K., Freidinger, R.: A pin-entry method resilient against shoulder surfing. In: CCS 2004: Proceedings of the 11th ACM conference on Computer and communications security, pp. 236–245. ACM Press, New York (2004)
27. University of British Columbia (Ubcsat, the stochastic local search sat solver), http://www.satlib.org/ubcsat
28. Kumar, M., Garfinkel, T., Boneh, D., Winograd, T.: Reducing shoulder-surfing by using gaze-based password entry. In: Symposium On Usable Privacy and Security (SOUPS) (2007)
29. Suo, X., Zhu, Y., Owen, G.S.: Graphical passwords: a survey. In: Proceedings of 21st Annual Computer Security Application Conference (ACSAC 2005), December 5-9, 2005, Tucson AZ (US), pp. 463–472 (2005)
30. Graphical Password Project: Fa1ces (1998), http://www.realuser.com