On the Security of a Two-Factor Authentication Scheme[⋆]

Luigi Catuogno[1] and Clemente Galdi[2]

[1] Dipartimento di Informatica ed Applicazioni, Università di Salerno
Via Ponte Don Melillo, I-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, I-80126 Napoli (NA), Italy
clemente.galdi@unina.it

Abstract. In this paper we evaluate the security of a two-factor Graphical Password scheme proposed in [1]. As in the original paper, we model the attack of a passive adversary as a boolean formula whose truth assignment corresponds to the user secret. We show that there exist a small number of secrets that a passive adversary cannot extract, independently from the amount information she manages to eavesdrop. We then experimentally evaluate the security of the scheme. Our tests show that the number of sessions the adversary needs to gather in order to be able to extract the users secret is relatively small. However, the amount of time needed to actually extract the user secret from the collected information grows exponentially in the system parameters, making the secret extraction unfeasible. Finally we observe that the graphical password scheme can be easily restated in as a device-device authentication mechanism.

1 Introduction

In a Graphical Password scheme, a remote system authenticates a user by means of a challenge/response scheme in which the system poses the challenge as "something" depicted on the terminal screen. This challenge can be answered correctly by the legitimate user who knows a certain secret. The user "computes" the response as the output of a *cognitive function* that takes as inputs the secret and what she sees on the terminal. Existing schemes vary according to (a) the way in which secrets and challenges are represented, (b) the cognitive function definition and (c) how the user sends back the response (e.g., by typing some numbers on a keypad, by clicking on some areas of an image etc.)

The main threat to Graphical Password Schemes is known as *shoulder surfing attack*. In this attack an adversary observes "over the user's shoulders" whatever appears on the screen and everything she does during the authentication session, and collects any information useful to extract the user secret. Since early

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

schemes' main objective was ergonomy, most of the existing Graphical Password schemes do not implement any effective countermeasures against a malicious observer, who, in some case, could learn the user secret just observing a single or few authentication sessions. We refer the reader to [2] for a survey on the topic. Notice that "shoulder surfing" in human authentication protocols corresponds to the classical "passive eavesdropping" attack.

The authors in [1] presented a Graphical Password scheme that, under some assumptions, can be used as a two factor authentication scheme. In the same paper the authors present an attack based on Formula Satisfiability (SAT, for short) as earlier proposed in [3]. This attack expresses the information the adversary can obtain, by observing the communication between the user and the system, as a boolean formula whose truth assignment corresponds to the user secret.

As in the original paper, we assume the adversary to be a passive eavesdropper. 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. Because of this limitation, we say that an attack is successful only if the adversary can extract the user secret.

Our Contribution. In this paper we restate the protocol presented in [1] as a two-factor authentication scheme and we analyze the SAT-based attack described therein. We first show that there exists a small number of secrets, logarithmic in the size of the secrets' space, that cannot be uniquely extracted even when the adversary is provided with an unbounded number of authentication transcripts.

We then experimentally evaluate the security of the two-factor authentication scheme using the SAT-based attack. We show that the amount of time needed to extract the secret using such attack grows exponentially in the scheme parameters, making the authentication scheme extremely interesting. Indeed, even if the adversary is able to collect a sufficient number of transcripts, she may not be able to extract the user's secret, simply because such operation is computationally infeasible.

We stress that our experimental evaluation only considers sets of "small" parameters for which the authentication scheme is *not* secure. As it will be clear in Section 4, it was *not feasible* to obtain results for bigger values of the parameters, i.e., for the parameters that make the scheme secure. We finally show that such an authentication scheme can be easily deployed for the authentication of small devices, e.g. RFID tag-to-reader or reader-to-tag authentication. Due to space limitations, proofs and figures are omitted from this version of the paper.

2 The GRAPE scheme

Let O be a set consisting of $q = pa$ distinct objects, for some positive integers a and p. In GRAPE [1], a *challenge* is a random permutation $\alpha = (o_1, \ldots, o_q)$ of all the objects in O. The challenge α in shown to the user as a matrix consisting of a rows and $p = q/a$ columns.

The answer to the challenge, the "password", corresponds to the position of a sequence of secret objects $\sigma = (\sigma_1, \ldots, \sigma_m)$ in the challenge matrix. It is clear that the password typed in by the user changes in each session as the challenge changes. To be more precise the secret is a sequence of m questions, called *queries* like: *"On which row of the screen do you see the secret object σ_i?"*. Queries are chosen independently and, hence, the set of possible queries has size $|O|^m$.

The user response to the challenge is an array $\beta = (\beta_1, \ldots, \beta_m)$, where each β_i is a number drawn from the set $A = \{0, 1, \ldots a-1\}$ and corresponds to the answer to the i-th query. A session transcript is a pair $\tau = (\alpha, \beta)$, where α is a challenge and β is the user response to α.

The original scheme was analyzed under three different authentication strategies. We only focus on the "correct-wrong" one in which the user is required to correctly answer exactly k out of n queries while giving *wrong* answers to the remaining ones.

A two-factor authentication protocol. The "correct-wrong" strategy just described enjoys particular features when the system arranges the objects on two rows, i.e., $a = 2$, and the user is required to answer correctly to exactly $m/a = m/2$ queries. Indeed, in case the user, for every challenge, has to answer correctly to *a specific set of $m/2$ queries* (unknown to the adversary), it is possible to show that the probability with which the adversary succeeds in authenticating giving random answers to the challenge drops to $1/2^m$, where m is the number of objects in the secret. Clearly, the required set of correct answers needs to change for each challenge, otherwise the adversary will be able to use a counting argument and identify the user secret.

The authentication scheme we will analyze is the following: A challenge α is a random arrangement of the objects in O on two rows. The user is required to answer correctly to *a specific set of $m/2$ out of m queries* and to give *wrong* answers to the remaining ones.

It is not reasonable to assume that a human being is able to compute or remember a list of different sets of answers (to be used in consecutive authentication sessions) to which she has to answer correctly. Here comes into play an authentication token whose only role is to generate the *specific set of $m/2$ answers* in place of the user. We notice that such tokens are already used for providing one-time PINs. However, 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 the presented authentication scheme, the mere possession of the device still does not allow the adversary to succeed in the authentication without the knowledge of the user secret. Thus the user secret still plays a central role in the multi-modal authentication scheme. We assume that the adversary is not allowed to read the token.

3 Analysis

In this section we describe the SAT based attack to the two-factor authentication scheme just described, presented in [1]. Roughly speaking, given a number of transcripts corresponding to successful authentication sessions, we model the information that an adversary may obtain by means of a boolean formula in a way that a truth assignment to the boolean formula corresponds to a possible user secret. Clearly, as the number of transcripts used to construct the formula grows, the number of possible truth assignment decreases. However, for the formula we construct, there always exists at least one truth assignment, i.e., the one corresponding to the user secret.

We then show that, if the adversary is provided with a sufficient number of transcripts, such approach correctly extracts the user secret when its "plurality" is at least 3, i.e., if the user secret contains at least 3 different objects. At the same time, we show that if the secret plurality is at most 2, this approach cannot distinguish among strongly related secrets. However, since the latter case only occurs in a small number of cases, the proposed strategy can still be used to extract the user secret.

Preliminaries. In the following we will denote by α^k the challenge for the the k-th transcript. Since $a = 2$, α^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 indices of the objects on the first (resp. the second) row of α^k. Since the challenge α^k is a random *permutation* of the q distinct objects in the set O, it holds that $\{i_1^k, \ldots, i_p^k\} \cap \{i_{p+1}^k, \ldots, i_q^k\} = \emptyset$ and $\{i_1^k, \ldots, i_p^k\} \cup \{i_{p+1}^k, \ldots, i_q^k\} = \{1, \ldots, q\}$.

Let $\sigma = (\sigma_1, \ldots, \sigma_m) \in O^m$ denote the user secret, and let $\beta^k \in \{0,1\}^m$ be the response response of the user to challange α^k w.r.t. σ.

We say that an array $c = (c_1, \ldots, c_m) \in \{0,1\}^m$ is the *correct answer*, w.r.t. secret $\sigma = (\sigma_1, \ldots, \sigma_m)$, to the challenge $\alpha^k = ((i_1^k, \ldots, i_p^k), (i_{p+1}^k, \ldots, i_q^k))$ if the i-th component of the secret belongs to row c_i of the challenge. More formally, for $i = 1, \ldots, m$, $c_i = 0$ iff $\sigma_i \in \{o_{i_1^k}, \ldots, o_{i_p^k}\}$, $c_i = 1$ otherwise.

Recall that, since the user correctly answers to exactly $m/2$ queries while gives wrong answers to the remaining $m/2$, for a given transcript (α^k, β^k) if c is the correct answer to the challenge α^k w.r.t. a secret σ, the arrays β^k and c agree on exactly $m/2$ components. In other words, the adversary (1) is able to obtain the transcript (α^k, β^k); (2) knows that exactly $m/2$ components of β^k are correct and (3) she does not c. Let us define $A_m = \{a = (a_1, \ldots, a_m) \in \{0,1\}^m \mid w(a) = m/2\}$, where $w(\cdot)$ denotes the Hamming weight of a.

Finally, let ψ be a boolean formula defined on a set X of variables and let x be a truth assignment for the variables in X. If x satisfies ψ we will write $x \vDash \psi$, otherwise we will write $x \nvDash \psi$.

Constructing the Formula. Given the above definitions, we show how to construct a boolean formula given a set of transcripts. 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.

Let $\sigma = (\sigma_1, \ldots, \sigma_m)$ be a secret, where $\sigma_i \in O$ for $i = 1, \ldots, m$, . The *Truth Assignment* $T_\sigma = (x_{1,1}, \ldots, x_{q,m})$ induced by σ is defined as: $x_{i,j} = 1$ if $\sigma_j = o_i$, or $x_{i,j} = 0$ otherwise.

Since each o_j appears in α^k exactly once, for every i, the i-th component of the user secret belongs either to row zero or row one of α^k. For every $j = 1, \ldots, m$, i.e., for every component of the user secret, we define $\phi^k_{0,j} = x_{i^k_1,j} \vee \ldots \vee x_{i^k_p,j}$ and $\phi^k_{1,j} = x_{i^k_{p+1},j} \vee \ldots \vee x_{i^k_q,j}$. It holds that, if j-th component of the user secret, σ_j, belong to row zero in challange α^k, then $T_\sigma \models \phi^k_{0,j}$. At the same time, σ_j does not belong to row one of α^k and thus $T_\sigma \models \overline{\phi^k_{1,j}}$. This means that $T_\sigma \models \phi^k_{0,j} \wedge \overline{\phi^k_{1,j}}$. Similarly, if σ_j belongs to row one of α^k then $T_\sigma \models \overline{\phi^k_{0,j}} \wedge \phi^k_{1,j}$. Notice that the same holds for every $j = 1, \ldots, m$. Given a transcript and an array $a = (a_1, \ldots, a_m) \in A_m$ we will use the following notation:

$$\psi^k(a) = \bigwedge_{j=1}^{m} \left(\phi^k_{\beta_j \oplus a_j, j} \wedge \overline{\phi^k_{(1-\beta_j) \oplus a_j, j}} \right) \qquad \text{and} \qquad \psi^k = \bigvee_{a \in A_m} \psi^k(a) \qquad (1)$$

From the above discussion, $T_\sigma \models \psi^k(a)$ if the correct answer c for α^k can be written as $c = \beta^k \oplus a$ and, thus, $T_\sigma \models \psi^k$. It is not hard to show that, for every truth assignment x, if $x \models \psi^k(a)$ then $x \not\models \psi^k(b)$, for every $b \in A_m \setminus a$. Intuitively, the satisfiability of the above formulas follows from the observation that, for a generic transcript (α, β), there exists exactly one boolean array $(a_1, \ldots, a_m)$ that identifies the correct and wrong answers in β. 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, thus for $\psi = \bigwedge_{k=1}^{t} \psi^k$ it holds that $T_\sigma \models \psi$. Notice that the number of variables $x_{i,j}$ does *not* depend on the number of transcripts, i.e, for every k, the formulas ψ^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: $\epsilon_{m,q} = \bigwedge_{j=1}^{m} \bigvee_{i=1}^{q} (\overline{x_{1,j}} \wedge \ldots \wedge \overline{x_{i-1,j}} \wedge x_{i,j} \wedge \overline{x_{i+1,j}} \wedge \ldots \wedge \overline{x_{q,j}})$.

For every possible secret σ and for every possible sequence of successful transcripts $((\alpha^1, \beta^1), \ldots, (\alpha^t, \beta^t))$, if ψ and $\epsilon_{m,q}$ are defined as above and if $\mu_\psi = \psi \wedge \epsilon_{m,q}$ it holds that $T_\sigma \models \mu_\psi$. Notice that a truth assignment for μ_ψ 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_ψ is satisfiable also in this case but there might exists multiple truth assignments.

Impossibility result. A passive attack to such a scheme has an inherent impossibility result. We say that two secrets $\sigma, \chi \in O^m$ are *indistinguishable* if, for every transcript (α^k, β^k) it holds that $T_\sigma \models \psi^k \wedge \epsilon_{m,q}$ if and only if $T_\chi \models \psi^k \wedge \epsilon_{m,q}$. In other words, there exists no transcript that can be used to discriminate one secret from the other. Furthermore, we define the plurality of a secret σ, denoted by $p(\sigma)$, as the number of *different* objects composing the secrets. For example, the plurality of $\sigma = (1, 1, \ldots, 1)$ is equal to 1, the plurality of $\sigma = (1, \ldots, 1, 3, \ldots, 3)$ is equal to 2. Finally, if σ is a secret with plurality equal to two, then it is composed by two different objects, say $\sigma_1, \sigma_2 \in O$. The complement $\overline{\sigma}$ of σ is obtained from σ substituting each occurrence of σ_1 with σ_2 and viceversa. We can prove the following:

Theorem 1. *Let $\sigma \in O^m$ be a secret. It holds that:*

- *If $p(\sigma) = 1$, then σ is indistinguishable from any other secret with plurality 1 and it is distinguishable from all the secrets with plurality greater than 1;*
- *If $p(\sigma) = 2$, then σ is indistinguishable from $\overline{\sigma}$ and it is distinguishable from all the other secrets;*
- *if $p(\sigma) > 2$, then σ is distinguishable from all the other secrets;*

Theorem 1 states that, given a sufficient number of transcripts, the formula can be used to extract every secret with plurality greater than 2. At the same time, even if the adversary is given access to an infinite sequence of transcripts and *independently* from the specific attack, there exist a number of secrets, logarithmic in the size of the secrets' space, that cannot be uniquely identified.

4 Experimental evaluation

In this section we describe the experiments we have run in order to evaluate the performance of the system under analysis. The experiments have been run on a cluster composed by 3 nodes, each equipped with two quad-core Xeon processors with 8 Gb of RAM running Scientific Linux and the Mosix Cluster Management system. The results we report in this section have been obtained using the SAT solvers NoClause [4] and SatMate [5], with exactly the same behavior. The reason of using two different solvers was to self-validate the results by avoiding the possibility that one solver was performing particularly well/bad given the specific formula structure. Such solvers take as input boolean formulae in different formats. We have used the ISCAS[3] format that, essentially, describes the circuit associated to the formula by means of INPUT, OUTPUT, AND, NOT, OR, XOR gates.

One of the problems in automating consecutive runs of the experiments, is the fact that the formula μ_ψ we need to evaluate is always satisfiable. On one hand, NoClause simply outputs a statement "Satisfiable"/"Unsatisfiable", and thus, in our case, it always outputs "Satisfiable". On the other hand, SatMate (if the formula is satisfiable) provides a truth assignment. Unfortunately, in case the number of transcripts is not sufficient to extract the user secret, the formula μ_ψ has multiple truth assignments. Thus, the output of SatMate should have been checked against the actual user secret.

In order to unify (and simplify) the testing using both solvers, we have preferred to work around the above differences as follows: we have added a new clause δ to the formula μ_ψ that excludes the user secret. More precisely, if $\sigma = (\sigma_1 = o_{i_1}, \ldots, \sigma_m = o_{i_m})$ is the user secret, we define $\delta = \overline{(x_{i_1,1} \wedge \ldots \wedge x_{i_m,m})}$.

[3] See http://logic.pdmi.ras.ru/∼basolver/rtl.html for some details.

Intuitively, $(x_{i_1,1} = 1, \ldots, x_{i_m,m} = 1)$ is always a truth assignment for the formula μ_ψ. Thus if μ_ψ has at least two truth assignments, then $\mu_\psi \wedge \delta$ is still satisfiable. On the other hand, if $(x_{i_1,1} = 1, \ldots, x_{i_m,m} = 1)$ is the *only* truth assignment for μ_ψ, then $\mu_\psi \wedge \delta$ is not satisfiable. Thus, from our point of view, if $\mu_\psi \wedge \delta$ is satisfiable, the number of transcripts used to construct μ_ψ is not sufficient to extract the user secret. On the other hand if $\mu_\psi \wedge \delta$ is not satisfiable, then the number of transcripts used to construct μ_ψ is sufficient for extracting the user secret. Every run of an experiment is identified by three parameters, the secret length m, the number of objects in the challenge q and the number of transcripts t used to construct the formula μ_ψ.

We have run several experiments with secret lengths of $4, 6$ or 8. We stress a secret of length 8 is *not* secure in a real-life deployment since an adversary has probability $1/2^8 = 1/256$ of guessing the user secret. On the other hand, as it will be clear soon, it was simply unfeasible to obtain results for values of m greater than 8. Our experiments can be useful, however, to determine the behavior of the system in case it is instantiated with *secure real-life* parameters.

The first thing we have done is tried to evaluate the minimum number of transcript that are needed in order to extract the secret as a function of the secret length and the number of objects in the challenge. As it was expected, as the secret length increases, the percentage of successfully extracted secret slightly decreases. However, in all cases, 30 transcripts are sufficient to extract the secret with probability close to 1. We have then evaluated the time growth depending on three variables. If we fix the secret length and the number of objects in the challenge, the average solution time increases *exponentially* until the number of transcripts reaches 20 while, after this value, it stabilizes or even slightly decreases. This can be explained since with "few" transcripts, the number of truth assignments for the formula is high and, thus, the solver can easily find one truth assignment. On the other hand, when the number of transcripts increases, the number of truth assignments for $\mu_\psi \wedge \delta$ decreases quickly to zero, and thus the solver needs more or less the same time for proving the formula to be unsatisfiable or to find a truth assignment.

If we do consider the time growth as function of the secret length m or the number of objects q in the challenge, it can be seen that the average solution time increases *exponentially* in both variables. As for the secret length, the exponential growth was expected since the size of the input formula grows exponentially in this parameter. This unexpected growth in time made simply unfeasible running tests with a bigger values for the parameters.

5 Applications to RFID

Following the lead of [6], we have considered possible applications of our scheme to the authentication of small devices like RFIDs. In particular, we have focused our attention to the tag-to-reader authentication. The two factor authentication scheme can be easily deployed as a "one-factor" authentication scheme on low-cost devices in which the tag plays the role of the user while the reader impersonated the terminal. Since the scheme is independent from the specific et of objects, in a such context, the challenge can be any arrangement of all binary strings in the set $\{0,1\}^{\ell}$. Clearly, there exists no "cognitive function", but the tag is only required to search its secret among a set of binary strings. Finally, the tag itself could either store the sequence of sets of queries to which it has to give the "correct answer". Alternatively, the device might generate such set using a pseudo-random generator, in which case the seed of the PRNG will be part of the secret shared between the tag and the terminal. Finally, periodical proactive secret updates could be provided more frequently with RFIDs w.r.t. the human scenario.

6 Conclusions.

In this paper we have experimentally evaluated a two-factor authentication scheme initially proposed in [1]. We have first shown that there exists a small number of secrets that cannot be uniquely extracted even if the adversary is provided with an infinite record of transcripts. We have then experimentally evaluated the attack performance using two different SAT solvers. Our results show that, although the number of transcripts needed to extract the user secret is small, even with small values of the parameter it becomes sometime infeasible to extract such information from the given set of transcripts. An interesting result could be to prove the hardness of the secret extraction process.

Moreover, we point out that our scheme does not use any cryptographic primitive and, for a basic implementation, it requires limited resources, making it suitable for applications to computationally constrained devices.

References

1. Catuogno, L., Galdi, C.: A graphical pin authentication mechanism for smart cards and low-cost devices. In: 2nd Workshop on Information Security Theory and Practices (WISTP 08). Volume 5019 of Lecture Notes in Computer Science., Springer-Verlag (2008)
2. 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, Tucson AZ (US). (2005) 463–472
3. Golle, P., Wagner, D.: Cryptanalysis of a cognitive authentication scheme (extended abstract). In: IEEE Symposium on Security and Privacy, IEEE Computer Society (2007) 66–70
4. Thiffault, C., Bacchus, F., Walsh, T.: Solving non-clausal formulas with dpll search. In Wallace, M., ed.: CP. Volume 3258 of Lecture Notes in Computer Science., Springer (2004) 663–678
5. Jain, H., Bartzis, C., Clarke, E.M.: Satisfiability checking of non-clausal formulas using general matings. In Biere, A., Gomes, C.P., eds.: SAT. Volume 4121 of Lecture Notes in Computer Science., Springer (2006) 75–89
6. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Proceedings of 25th International Cryptology Conference (CRYPTO 2005). Volume 3621 of Lecture Notes in Computer Science., Springer (2005) 293–308