Cryptanalysis of a Cognitive Authentication Scheme

Philippe Golle* David Wagner†

Abstract

We present attacks against two cognitive authentication schemes [3] recently proposed at the 2006 IEEE Symposium on Security and Privacy. These authentication schemes are designed to be secure against eavesdropping attacks while relying only on human cognitive skills. They achieve authentication via challenge response protocols based on a shared secret set of pictures. Our attacks use a SAT solver to recover a user's key in a few seconds, after observing only a small number of successful logins. These attacks demonstrate that the authentication schemes of [3] are not secure against an eavesdropping adversary.

1 Introduction

Consider a user who wants to establish a secure authenticated connection to a server from an untrusted client. For example, the user may be logging into her bank account from an Internet café. The password schemes traditionally used for authentication are insecure in this case, because the untrusted client can record the user's password, and use it to later impersonate the user. This threat must be taken seriously, given the spread of key-loggers and other spyware on users' machines.

To defend against this threat, [3] proposes new authentication schemes that are designed to be secure against eavesdropping attacks (these authentication schemes are described in more detail in a technical report [4]). The eavesdropping adversary is assumed capable of observing all user input and all communication between the client and the server. A further distinctive advantage of the schemes of [3] is that they rely only on human cognitive abilities. Users can engage unaided in the authentication protocol. Authentication is achieved via challenge response protocols based on a shared secret set of pictures.

In this paper, we show that the new cognitive authentication schemes of [3] are insecure against eavesdropping attacks. Assuming a passive eavesdropping adversary, we propose attacks that recover a user's key in a few seconds, after observing only a small number of successful logins. The crux of our attacks is the observation that every user's response to an authentication challenge allows the adversary to learn a boolean relationship between the bits of the user's secret key. These boolean relationships can easily be expressed in disjunctive normal form. With enough relationships, a SAT solver quickly recovers the user's key.

We demonstrate our attacks against the schemes exactly as defined in [3]. We also show that our attacks still work against variants of the schemes with larger parameters. This suggests that these schemes and their variants are fundamentally vulnerable to attacks based on SAT solvers. We are not hopeful that a secure variant of these schemes can be designed.

*Palo Alto Research Center – pgolle@parc.com

†University of California, Berkeley – daw@cs.berkeley.edu

Organization. We start with an overview of the cognitive authentication protocol of [3] (section 2). We then propose attacks against two versions of this protocol: the high complexity version (section 3) and the low complexity version (section 4).

2 Cognitive Authentication Protocol

The cognitive authentication scheme of [3] uses a set $\mathcal{B}$ of public pictures. We let $N = |\mathcal{B}|$ denote the size of this set. The secret authentication key of a user is a secret subset $\mathcal{F} \subset \mathcal{B}$ of size $M < N$. The authentication protocol consists of a number of challenge response rounds. The number of rounds in the protocol is adjustable depending on the level of security desired (more rounds lower the success probability of a guessing adversary). In each round, the user must answer a query about the subset $\mathcal{F}$. The exact nature of the query depends on the version of the protocol considered. In [3], two versions are proposed: a high complexity version, and a low complexity version (we present attacks against both versions of the protocol in sections 3 and 4.) Authentication succeeds if the user supplies correct replies to all challenge queries.

3 High Complexity Protocol

In this version of the protocol, the user is presented in each round with $n < N$ pictures selected randomly from $\mathcal{B}$. These n pictures are displayed in random order in a rectangular panel of R rows and C columns. Furthermore, an integer in the range $[0; P-1]$ is associated with each column and each row of the panel. We denote v_r the integer associated with row r and w_c the integer associated with column c. An example of such a panel is given in Figure 1.

P_{44}	P_{31}	P_{1}	P_{26}	P_{66}	2
P_{46}	P_{24}	P_{21}	P_{77}	P_{43}	3
P_{13}	P_{16}	P_{79}	P_{38}	P_{59}	0
P_{36}	P_{58}	P_{76}	P_{15}	P_{53}	2
0	1	0	3	1	

Figure 1: An example of a high complexity panel query. This 4-by-5 panel consists of $n = 20$ pictures drawn at random for a set of $N = 80$ pictures. The numbers shown along the bottom and right edges of the panel are the values w_c and v_r associated with the columns and rows of the panel. In this example, the numbers are chosen in the range $[0; 3]$, i.e. $P = 4$.

Given such a panel as a challenge, the user is asked to follow a mental path through the panel as follows. Starting from the top left corner, the user moves down whenever the current cell contains a picture that belongs to her secret set $\mathcal{F}$, and moves right whenever the current cell contains a picture in $\mathcal{B} - \mathcal{F}$. This path eventually exits the panel either through the bottom or the right edge of the panel. If the path exits the panel through the bottom edge, the user reports the value w_c associated with the exit column. If the path exits the panel through the right edge, the user reports the value v_r associated with the exit row.

Numerical parameters. The implementation described in [3] proposes the following parameters. The set $\mathcal{B}$ of public images is of size $N = 80$. A user's key consists of a secret subset of $M = 30$ images. In each challenge query, all pictures are displayed ($n = N = 80$) in a panel of $R = 8$ rows and $C = 10$ columns. The values v_r and w_c are chosen in the range $[0; 3]$, in such a way that

users' replies to queries are approximately uniformly distributed in the range $[0; 3]$. Each run of the authentication protocol consists of 11 rounds of challenge-response.

We demonstrate our attack first with this exact set of parameters. In particular, we use the values v_r and w_c defined in Figure 1 on page 3 of [3]. Our attacks are however not sensitive to this particular choice of parameters, and we show that they work against variants of the scheme with much larger parameters.

3.1 Attack

We propose an attack that recovers the secret key of a user after observing the user's replies to a few authentication challenges. Let $\mathcal{F}$ denote the secret set of pictures chosen by the user under attack. We define N boolean variables $A_1, \ldots, A_N$ associated with the N pictures in the public set $\mathcal{B}$. Let $A_i = 1$ if the i-th picture belongs to the set $\mathcal{F}$ and otherwise $A_i = 0$. We denote $\overline{A}_i$ the negation of variable A_i. Note that recovering the user's key is equivalent to determining the values of $A_1, \ldots, A_N$.

Boolean variables. Each challenge-response round of the authentication protocol reveals information about the user's key, and thus about the variables $A_1, \ldots, A_N$. To capture this information fully, we need to define the following additional boolean variables in every round:

- For $1 \leq r \leq R$ and $1 \leq c \leq C$, we introduce a boolean variable $B^k_{(r,c)}$ associated with the cell in row r and column c of the panel submitted to the user in round k. We define $B^k_{(r,c)} = 1$ if the path computed by the user in round k passes through cell (r, c), and $B^k_{(r,c)} = 0$ otherwise.
- For $1 \leq r \leq R$, we introduce a boolean variable $B^k_{(r,C+1)}$, and define $B^k_{(r,C+1)} = 1$ if the path computed by the user exits the panel on the right in row r, and $B^k_{(r,C+1)} = 0$ otherwise.
- For $1 \leq c \leq C$, we introduce a boolean variable $B^k_{(R+1,c)}$, and define $B^k_{(R+1,c)} = 1$ if the path computed by the user exits the panel at the bottom in column c, and $B^k_{(R+1,c)} = 0$ otherwise.

Boolean formulas. Given the user's replies to the panel queries, we learn the following boolean formulas between the variables $A_1, \ldots, A_N$ and $B^k_{(r,c)}$. First, we know that in every round the path computed by the user starts in the top left corner of the panel:

$$B^k_{(1,1)} = 1 \quad \forall\, k$$

Let $f(k, r, c)$ denote the index of the picture in the cell at row r, column c, of the panel submitted to the user in round k. The following formulas express the rules that the user follows to compute a path through the panel:

$$\begin{aligned} \left(A_{f(k,r,c)} \wedge B^k_{(r,c)}\right) &\Rightarrow B^k_{(r+1,c)} \quad \forall\, k,\ \forall\, r \in \{1, \ldots, R\},\ \forall\, c \in \{1, \ldots, C\} \\ \left(\overline{A}_{f(k,r,c)} \wedge B^k_{(r,c)}\right) &\Rightarrow B^k_{(r,c+1)} \quad \forall\, k,\ \forall\, r \in \{1, \ldots, R\},\ \forall\, c \in \{1, \ldots, C\} \end{aligned}$$

Finally, let $p^k \in \{0, \ldots, P-1\}$ denote the reply submitted by the user in round k. The following rules express the constraints on the path imposed by the user's reply. First, the path cannot end in row r if $v_r \neq p^k$, nor in column c if $w_c \neq p^k$:

$$\begin{aligned} \overline{B^k}_{(r,C+1)} &\quad \forall\, r \text{ such that } v_r \neq p^k \\ \overline{B^k}_{(R+1,c)} &\quad \forall\, c \text{ such that } w_c \neq p^k \end{aligned}$$

Secondly, the path must end in a row r for which $v_r = p^k$ or a column c for which $w_c = p^k$:

$$\Big(\bigvee_{r \mid v_r = p^k} B^k_{(r,C+1)} \Big) \vee \Big(\bigvee_{c \mid w_c = p^k} B^k_{(R+1,c)} \Big)$$

All the boolean formulas above can be converted into equivalent boolean formulas in disjunctive normal form. For example, $P \Rightarrow Q$ is equivalent to $\overline{P} \vee Q$. These are given as input to a SAT solver. Given enough formulas, a SAT solver quickly outputs a unique assignment of values to the variables $A_1, \ldots, A_N$. This assignment corresponds to the user's secret key.

Implementation of the attack. We tested our attacks with the UBCSAT implementation [2] of the SAPS [1] SAT solver (this SAT solver was the most efficient of those we tested). Table 1 summarizes the results of our attacks against the high complexity protocol. We tested the attacks with various of the parameters suggested in [3]. In all cases, our attack correctly recovers a unique candidate for the the user's secret key. The complexity of the attack is measured according to the number of challenge-response rounds that a passive adversary must observe, and the running time (in seconds) of the SAT solver. The running time of the attack was measured on a desktop PC running Windows XP with dual 3.40 GHZ CPUs and 1.00 GB of RAM.

Authentication protocol parameters				Attack complexity	
N	M	P	Panel size	# Rounds	Time (seconds)
80	30	4	8 by 10	60	102
80	30	4	8 by 10	100	7
120	45	4	8 by 10	500	45
120	45	2	8 by 10	1000	≈ 960

Table 1: Cost of our attack against the high complexity authentication protocol, for various choices of protocol parameters.

Discussion. The first row of Table 1 shows that, for parameters suggested in [3], a passive eavesdropping adversary recovers a user's secret key in seconds after observing only 60 rounds of challenge-response. Bearing in mind that successful authentication requires multiple rounds of challenge-response (11 rounds are suggested in [3]), the attacker learns a user's secret key after observing as few as 6 successful user logins.

This attack is close to optimal in the number of rounds it requires. Indeed, the secret keys of users are chosen from a space with $\log_2 \binom{80}{30} \approx 72.9$ bits of entropy. With $P = 4$, each user's reply to a challenge query reveals $\log_2(4) = 2$ bits of entropy of the secret key. Thus, $37 \approx 72.9/2$ challenge-response rounds are needed at the very minimum to uniquely recover a user's key.

With fewer than 60 rounds, our attack tends to recover multiple candidates for a user's key. More rounds on the other hand decrease the running time of the attack, as shown in the second row of Table 1.

The last two rows of Table 1 show that the attack also works for a larger choice of parameters (when users' keys consist of $M = 45$ pictures chosen from a set of size $N = 120$). Given the fundamental limitations of SAT solvers, there is no doubt that our attack would not work with sufficiently large parameters. However, large parameters present users with the challenge of memorizing a large number of images. The results of Table 1 suggest that user's memory might fail long before large enough parameters place the authentication scheme beyond the reach of our attack.

4 Low Complexity Protocol

The setup for this version of the protocol is identical to the high complexity protocol. Recall that we denote $\mathcal{B}$ a public set of pictures of size N. The authentication key of a user is a secret subset $\mathcal{F} \subset \mathcal{B}$ of size $M < N$. The difference lies in the questions asked of the user in each authentication challenge.

In the low complexity version of the authentication protocol, the user is presented in each challenge-response round with an ordered list of n pictures selected randomly from $\mathcal{B}$. Each picture in the list is assigned a random bit (either 0 or 1) which is shown next to it. These random bits are chosen such that the number of pictures assigned 0 equals the number of pictures assigned 1. The user is asked one of the following binary questions:

- **First case.** Identify the first picture in the ordered list that belongs to $\mathcal{F}$, and the last picture in the list that belongs to $\mathcal{F}$. Let b_0 and b_1 be the bits associated with these two images. Output $b_0 \oplus b_1$.
- **Second case.** Identify the first, second and last pictures in the ordered list that belong to $\mathcal{F}$. Let b_0, b_1 and b_2 be the bits associated with these pictures. Output the majority of b_0, b_1, b_2.

The description of the protocol given in [3] does not specify what reply the user should give when the ordered list contains fewer than 2 images from $\mathcal{F}$ (in the first case) or fewer than 3 images from $\mathcal{F}$ (in the second case). The parameters of the protocol are chosen such that this situation happens very infrequently. In our analysis, we generate ordered lists such that this situation never happens.

Numerical parameters. The implementation described in [3] proposes the following parameters. The set $\mathcal{B}$ of public images is of size $N = 240$. A user's key consists of a secret subset of $M = 60$ images. In each challenge query, $n = 20$ random pictures are displayed. Each run of the low complexity authentication protocol consists of 22 rounds of challenge-response.

4.1 Attack

We propose an attack that recovers the key of a user after observing the user's replies to a few authentication rounds. Let $\mathcal{F}$ denote the secret set of pictures chosen by the user under attack. As in section 3.1, we define N boolean variables $A_1, \ldots, A_N$ associated with the N pictures in the set $\mathcal{B}$. We define $A_i = 1$ if the i-th picture belongs to the set $\mathcal{F}$ and otherwise $A_i = 0$. We denote $\overline{A}_i$ the negation of variable A_i. Note that recovering the user's key is equivalent to determining the values of $A_1, \ldots, A_N$.

In every round, observing the user's reply to an authentication query reveals boolean relationships between the variables $A_1, \ldots, A_N$. Specifically, let $i_1, \ldots, i_n \in \{1, \ldots, N\}$ denote the indices of the n pictures presented to the user in the ordered list, and let $b_1, \ldots, b_n$ denote the bits associated with these pictures. Finally, let b denote the bit returned by the user. We learn first that at least one of the images in the ordered list is known to the user. In other words, the following formula is true:

$$(A_{i_1} \vee A_{i_2} \vee \ldots \vee A_{i_n}).$$

First case. For all $1 \leq j < k \leq n$, we know that $b_j \oplus b_k \neq b$ implies that pictures i_j and i_k cannot be the first and last pictures that belong to $\mathcal{F}$ in the ordered list. Thus:

- If $b_1 \oplus b_n \neq b$, we learn that pictures i_1 and i_n cannot both belong to $\mathcal{F}$. In other words, if $b_1 \oplus b_n \neq b$ the following formula is true: $(\overline{A}_{i_1} \vee \overline{A}_{i_n})$.

- For $1 \leq j < k \leq n$ such that $(j,k) \neq (1,n)$ and $b_j \oplus b_k \neq b$, we learn that if pictures i_j and i_k both belong to $\mathcal{F}$, then there must exist at least one other picture that belongs to $\mathcal{F}$ in the range $[1, j-1] \cup [k+1, n]$. In other words, for all $1 \leq j < k \leq n$ such that $(j,k) \neq (1,n)$ and $b_j \oplus b_k \neq b$ the following formula is true:

$$(A_{i_j} \wedge A_{i_k}) \Rightarrow \Big((A_{i_1} \vee \ldots \vee A_{i_{j-1}}) \vee (A_{i_{k+1}} \vee \ldots \vee A_{i_n}) \Big).$$

Second case. For all $1 \leq j < k < l \leq n$, we know that if the majority of the three bits (b_j, b_k, b_l) is not equal to b, then this implies that pictures i_j, i_k and i_l cannot be the first, second and last pictures that belong to $\mathcal{F}$ in the ordered list. Thus:

- If the majority of (b_1, b_2, b_n) is not equal to b, the following formula is true: $(\overline{A}_{i_1} \vee \overline{A}_{i_2} \vee \overline{A}_{i_n})$.
- For all $1 \leq j < k < l \leq n$ such that $(j,k,l) \neq (1,2,n)$ and such that the majority of (b_j, b_k, b_l) is not equal to b, we learn that if pictures i_j, i_k and i_l all belong to $\mathcal{F}$, then there must exist at least one other picture that belongs to $\mathcal{F}$ in the range $[1, j-1] \cup [j+1, k-1] \cup [l+1, n]$. In other words, for all $1 \leq j < k < l \leq n$ such that $(j,k,l) \neq (1,2,n)$ and the majority of (b_j, b_k, b_l) is not equal to b, the following formula is true:

$$(A_{i_j} \wedge A_{i_k} \wedge A_{i_l}) \Rightarrow \Big((A_{i_1} \vee \ldots \vee A_{i_{j-1}}) \vee (A_{i_{j+1}} \vee \ldots \vee A_{i_{k-1}}) \vee (A_{i_{l+1}} \vee \ldots \vee A_{i_n}) \Big).$$

As in section 3.1, these boolean formulas are converted into equivalent boolean formulas in disjunctive normal form and given as input to a SAT solver. Given enough formulas, a SAT solver quickly outputs a unique assignment of values to the variables $A_1, \ldots, A_N$. This assignment corresponds to the user's secret key.

Implementation of the attack. We tested our attacks with the UBCSAT implementation of the SAPS SAT solver on a desktop PC running Windows XP with dual 3.40 GHZ CPUs and 1.00 GB of RAM. Table 2 summarizes the results of our attacks. With these parameters suggested in [3], our attack recovers the user's authentication key in less than 1 second, given 250 challenge-response rounds (first case) or 400 rounds (second case).

Authentication protocol parameters				Attack complexity	
N	M	n	Query type	# Rounds	Time (seconds)
240	60	20	Case 1	250	< 1
600	150	20	Case 1	800	2.6
240	60	20	Case 2	400	< 1

Table 2: Cost of our attack against the low complexity authentication protocol, for various choices of protocol parameters.

Discussion. As noted earlier, successful authentication requires multiple rounds of challenge-response (22 rounds are suggested in [3] for the low complexity protocol). Thus, the first row of Table 2 shows that a passive eavesdropping adversary can recover a user's secret key after observing as few as 12 successful user logins. With fewer rounds, our attack tends to recover multiple candidates for a user's key. Our attack also works for a larger choice of parameters. When users' keys consist of $M = 150$ pictures chosen from a set of size $N = 600$, the attack recovers the user's secret key in 2.6 seconds. No reasonable choice of parameters (from the viewpoint of the memory effort required of users) can place the low-complexity authentication scheme beyond the reach of our attack.

5 Conclusion

We have shown that the cognitive authentication schemes proposed in [3] are insecure against eavesdropping attacks. Assuming a passive eavesdropping adversary, our attacks recover a user's secret key in a few seconds, after observing only a small number of successful logins. Designing secure authentication schemes that resist eavesdropping attacks *and* rely only on the cognitive abilities of humans remains a challenging open problem.

References

[1] F. Hutter, D. Tompkins and H. Hoos. Scaling and Probabilistic Smoothing: Efficient Dynamic Local Search for SAT. In LNCS 2470: Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming, pages 233–248. Springer Verlag, 2002.

[2] UBCSAT. The Stochastic Local Search SAT Solver from The University of British Columbia. `http://www.satlib.org/ubcsat/`

[3] D. Weinshall. Cognitive authentication schemes safe against spyware (short paper). In *Proc. of the 2006 Symposium on Security and Privacy*, pp. 295–300.

[4] D. Weinshall. Cognitive authentication schemes safe against spyware. Hebrew University. Leibniz Center for Research in Computer Science. TR 2006-5, 2006.