SSSL: Shoulder Surfing Safe Login

Toni Perković, Mario Čagalj, and Nikola Rakić

Original scientific paper

***Abstract*—Classical PIN-entry methods are vulnerable to a broad class of *observation attacks* (shoulder surfing, key-logging). A number of alternative PIN-entry methods that are based on human cognitive skills have been proposed. These methods can be classified into two classes regarding information available to a passive adversary: (i) the adversary *fully observes* the entire input and output of a PIN-entry procedure, and (ii) the adversary can only *partially observe* the input and/or output.**

In this paper we propose a novel PIN-entry scheme - *Shoulder Surfing Safe Login* (*SSSL*). SSSL is a challenge response protocol that allows a user to login securely in the presence of the adversary who can observe (via key-loggers, cameras) user input. This is accomplished by restricting the access to SSSL challenge values. Compared to existing solutions, SSSL is both user-friendly (not mentally demanding) and cost efficient. Our usability study reveals that the average login time with SSSL is around 8 sec in a 5-digit PIN scenario. We also show the importance of considering side-channel timing attacks in the context of authentication schemes based on human cognitive skills.

I. Introduction

Personal Identification Numbers (PINs) are widely used in modern information systems to authenticate users. Unfortunately, classical PIN-entry methods (via keyboards, keypads and alike) are all vulnerable to *observation attacks*. By simply observing the input of a user who enters his/her PIN, an adversary can potentially compromise the user's login credentials. There are numerous ways to perpetrate an observation attack. Thus, it is possible to obtain login information by shoulder surfing, through the use of concealed micro cameras and telescopes [2]. By installing keyloggers and fake keypads an adversary can easily collect the entire input of a targeted user, from which the adversary can recover the login credentials. Another interesting attack directed at the keypad would be to spray suitable chemicals onto the keypad, from which an adversary may recover user input [3]. The problem of observation attacks is further amplified by widespread of computing devices and services requiring login credentials, such as laptops, PDAs, mobile phones, ATMs, e-banking and e-commerce applications, etc.

This threat has long been recognized and prior work contains a variety of proposals for attempting to deal with it. Many proposals require the user to perform some form of a cognitive task - so called *cognitive authentication schemes*. The problem of designing a cognitive PIN-entry method secure against eavesdroppers is truly challenging. Indeed, it was recently shown in [5] that the cognitive scheme proposed in [23] and all its variants are fundamentally vulnerable to attacks based on SAT solver. It is an open question if there exist a PIN-entry scheme resistant against active attacks [9].

We can divide existing PIN-entry methods roughly in two classes regarding information available to an adversary: (i) the adversary *fully observes* the entire input and output of a PIN-entry procedure, and (ii) the adversary can only *partially observe* the input and/or output. For example, the PIN-entry method [9] belongs to the first class (*fully observable*). In this method all information exchanged between the user and the interrogator is available to the adversary. Unfortunately, this fact significantly increases the amount of cognitive effort for the user. On the other hand, the PIN-entry method [26] falls in the second class (*partially observable*). In this method, the user receives a challenge via a *protected channel*, and enters the response via a public keypad. Method *Shoulder Surfing Safe Login* (*SSSL*) that we propose in this paper is inspired by the work in [26]. Like the method in [26], SSSL also involves a protected channel through which a user receives challenges. However, unlike [26] SSSL does not require users to perform any complicated mathematical or mentally demanding operation (Section III).

The design choice to include the protected channel in our scheme (SSSL) was motivated by the following observations about the methods from the first class (the "fully observable" model). Firstly, designing secure cognitive PIN-entry schemes in the "fully observable" model is challenging as the SAT solver attack show. Secondly, secure PIN-entry schemes from this model involve multiple rounds of a basic challenge-response protocol and they require users to perform complicated mathematical calculations [9], which is a major deterrent to the acceptance of such technology (an average time of about 166 seconds in [9]).

The SSSL PIN-entry scheme implements the *one-time pad* paradigm. Thus, to enter a single digit of a secret PIN, a user first receives a challenge (a random number between 1 and 9) from an interrogator (computer, ATM) over a channel that ensures secrecy and integrity (e.g., via earphones). Next, the user visually locates the received challenge in a special table of digits displayed on his/her computer's screen. Finally, the user responds by clicking on the appropriate button (shown on the computer's screen) that uniquely links the secret challenge with the secret digit of the PIN.

Our solution, *Shoulder Surfing Safe Login*, is simple, user-friendly (not mentally demanding), cheap to implement and allows fast authentication (i.e., the average login time is 8 sec in the 5-digit PIN scenario - Section IV).

The rest of the paper is organized as follows. In Section II,

Manuscript received June 23, 2009; revised June 14, 2010.

The material in this paper was presented in part at the 17^{th} International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2009), Split-Hvar-Korčula, Croatia, Sept. 2009.

Authors are with the Department of electronics, FESB, University of Split, Croatia (e-mail: {toperkov, mcagaljg}@fesb.hr, n1kola.pax@gmail.com).

we discuss related work and give the attacker model along with usability and design requirements. We describe our SSSL method in Section III and provide the usability analysis in Section IV. In Section V, we perform the security analysis of SSSL. In Section VI we introduce to some possible applications of SSSL Finally, we conclude in Section VII.

II. Related Work, Objectives and Threat Model

In this section we summarize related research that motivates our approach to the design of a secure login method. We describe the pros and cons of other existing login methods. Finally, we describe our objectives and the threat model used in this work.

A. Existing PIN-entry Methods

There is a body of research focused on designing secure PIN-entry methods in face of the threat posed by observation attacks. As noted, we can divide existing PIN-entry methods in two classes regarding information available to an adversary. In one class the adversary fully observers the entire input and output of the PIN-entry procedure - the **fully observable** adversarial model. In the second class, the adversary can only partially observe the input and/or output - the **partially observable** adversarial model.

The basic difference between the two classes is that methods developed for the partially observable model usually involve some nonstandard, expensive or hard-to-use hardware [18], [20], [26]. At the same time, designing secure PIN-entry methods appears to be much easier in this model. Indeed, disregarding the threat of *side-channel attacks* that is common to both classes (Section V), we are not aware of any passive attack targeted at methods designed in the partially observable model. On the other hand, methods from the first class are purely software-based solutions [3], [4], [17], [27], [19], [23], [25], [24]. However, designing a method secure even against a simple passive attacks in the fully observable model appears to be challenging [19], [23], [5].

1) PIN-entry in the Fully Observable Model: **Passive Attack on the PIN-entry scheme**. To defeat observation attacks Roth et. al. propose a novel PIN-entry method [19]. Let us assume that a user wants to enter ith digit of his/her secret PIN. The main idea is to present the user two disjoint sets $S_{i,\mathrm{b}}$ and $S_{i,\mathrm{w}}$ of PIN digits layed out on a regular PIN pad, by randomly coloring half of the keys black (b) and the other half white (w). To enter ith digit of his/her secret PIN, the user has to enter in which set ($S_{i,\mathrm{b}}$ or $S_{i,\mathrm{w}}$) the PIN digit is. Then, the sets are shuffled and another round is played. After couple of rounds (e.g., 4) the ATM can determine the entered PIN digit unambiguously by intersecting the sets chosen by the user in each round, e.g.,

$$i\text{th digit} \in S_{i,\mathrm{b}}(1) \cap S_{i,\mathrm{w}}(2) \cap S_{i,\mathrm{w}}(3) \cap S_{i,\mathrm{b}}(4) \ .$$

The same game is played for all PIN digits. This variant of the method is secure against an adversary (characterized by a short term memory) [19]. However, an observer who records an entire session with a camera will be able to determine the entered digit (PIN) by the same algorithm as the ATM.

For this reason, Roth et. al. propose a variant of the basic method by reducing the number of rounds and thereby introducing uncertainty about the entered PIN digits. Speaking mathematically, in a single game with 3 rounds per digit we have $|\cap_{t=1}^{3} S_{i,x(t)}| > 1$, where $x(t) \in \{\mathrm{b}, \mathrm{w}\}$. We call the set $\cap_{t=1}^{3} S_{i,x(t)}(k)$ the uncertainty set of the ith digit in the kth game and denote it with $U_i(k)$.

To recover the ith digit of the user's PIN, the adversary will observe k successful login sessions, in each session record the sets $S_{i,x(t)}(k)$ ($x(t) \in \{\mathrm{b}, \mathrm{w}\}$, $t = 1, 2, 3$) and finally calculate the uncertainty sets $U_i(k)$, $k = 1, 2, \ldots, n$. Finally, by simply intersecting the sets $U_i(k)$, $k = 1, 2, \ldots n$, the ith digit is trivially recovered. To estimate the number of games (i.e., n) before the PIN digit can be recovered, we adopt security parameters from [19] and assume that for each game $k = 1, 2, \ldots, n$ we have $|U_i(k)| = 3$. Note that the uncertainty sets $|U_i(k)|$ ($k = 1, 2$) must not be identical, otherwise all digits from these sets will be accepted as valid ith digit of the PIN. On the other hand, any difference in the uncertainty sets reduces their size by one element. Therefore, we can expect to recover the valid ith digit after observing only 2-3 games (successful logins).

Other solutions include user's biometric information in the process of the authentication. Thus, Malek et. al. [13] propose to use a pressure as a binary input with graphical passwords. Kumar et. al. [12] use a gaze based password entry where an user enters a password or PIN by selecting from an on-screen keyboard using only the orientation of their pupils. In Thorpe et. al. [21] the user authenticates using a pass-thought system where the user thinks of the password and by the electrodes the system records and processes user's brain signals. All of those methods are based on an expensive hardware and we are not aware of a simple attack model which targets them.

SAT-solver Based Attacks on Cognitive Schemes [5]. In Weinshall [23], an user mentally computes a path formed by their portfolio images, and give an answer based on that mentally computed path. Every response to challenge allows to an adversary to learn a boolean relationship between the bits of the users secret key. After observing only several successful authentications, the adversary can reveal the key.

Matsumoto and Imai [15], Wang [22], and Matsumoto [14] provides schemes which are generally vulnerable to an active adversary or require the user to remember a large secret or perform many calculations to achieve a large number of secure authentications.

Next we briefly describe several purely-software based solutions not specifically designed to be secure against observation attacks.

Human cognition has been investigated with the goal to enhance recall of passwords like images [4], or Passfaces [17]. In Blonder [3] and Wiedenbeck et. al. [24] the user clicks on any place on the image to create a password, where [24] allows using arbitrary images. The user authenticates successfully clicking inside the tolerance around each chosen pixel in the correct sequence.

In Passfaces [17] a user's password is represented by a set of pictures showing different human faces. To authenticate, the user first receives a challenge in a form of a 3×3 grid with 9 randomly placed faces. The user selects the face that belongs to his/her set of password faces. This challenge-response protocol repeats several times and the user is successfully authenticated if he/she correctly answers on all challenges. In a related solution Deja Vu [4] pictures can show arbitrary content.

Finally, in the HB [9] scheme the human and the computer share the secret binary vectors $\mathbf{x}$ and $\mathbf{y}$ of length n, where $|\mathbf{x}| = |\mathbf{y}| = k$, that is $\mathbf{x}$ has k positions set to 1. The HB authentication schemes involves multiple rounds (m) of a basic challenge-response protocol. The authors themselves concluded the method was not usable in practice since it took 166 seconds the average time per successful authentication.

2) PIN-entry in the Partially Observable Model: Kuber and Yu [18] and Sasamoto et. al. [20] use a tactile channel as a secure hidden challenge channel. In the first solution the user is given a sequence of tactons to remember. To authenticate, the user rolls with a mouse over nine blank squares on the display causing an unique pattern appear under fingerprints. In the second solution the user simultaneously receives a visual challenge and a hidden tactile challenge via a protected channel. To authenticate the user has to answer correctly to several challenges. The common characteristics of the above solutions is that they require non-standard (potentially hard to use) hardware.

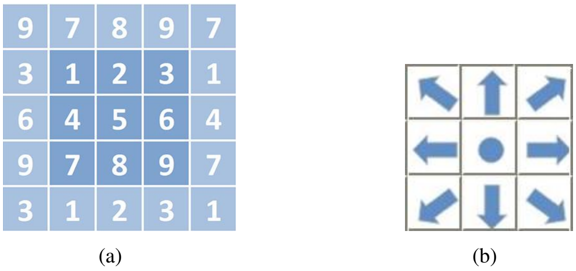
On the contrary, the solution described in the US patent [26] is based on a standard hardware, in which the user receives a challenge (a random number from $0, 1, \ldots, 9$) via a protected audio channel, adds modulo 10 each digit of his genuine PIN to the digits of the random number, and enters back the outcome via public keypad. Unlike our SSSL, this solution requires users to perform mathematical operations.

B. Security and Usability Objectives

1) Threat Model: We consider attacks by three types of adversaries:

- A *passive adversary* who eavesdrops on all public communication between the user and the end system. This adversary will try to learn SSSL protected login credentials by passively recoding the SSSL login procedure. The passive adversary does not interact in any way with the end system.
- An attacker who performs the *side-channel timing attack*. This attacker extends the passive adversary with the capability of recording the user's reaction time during the course of SSSL procedure by, for example, using key-logging malware.
- An *active adversary* who is able to compromise the system (an ATM or a computer) on which the user enters his/her login credentials. In this scenario, the compromised system (an ATM or a login computer) essentially provides connectivity between the user and some trusted end server (e.g., a banking server).

In Section V, we analyze the security of SSSL in the above security model. We show that the SSSL can successfully mitigate all the threats described above.



9	7	8	9	7
3	1	2	3	1
6	4	5	6	4
9	7	8	9	7
3	1	2	3	1

(a)

↖	↑	↗
←	●	→
↙	↓	↘

(b)

Fig. 1. SSSL interface: (a) In the SSSL table each digit i is an immediate neighbor to the other 8 digits from the set $\{1, 2, \ldots, i-1, i+1, \ldots, 9\}$; (b) A user enters his/her response via 8 arrow buttons and one center button.

2) Usability and Design Requirements: We require a login method to meet two additional objectives:

- *User friendliness* - a login method should be easy to learn and use, not mentally demanding (it must not require complex mathematical operations). This directly translates to low login error rates and fast authentication times (see Section IV).
- *Cost-efficiency* - a login method should be cheap to implement. It should not involve expensive and/or non-standardized hardware. Finally, it should be possible to integrate it seamlessly with existing login systems and solutions.

Our SSSL scheme meets all the above requirements. It is secure in a realistic adversarial model - *security*, it does not require mathematical computation on the part of the human user - *user-friendliness*, and finally, it is cheap to implement (not requiring the use of expensive and non-standardized hardware) - *cost-efficient*.

III. SSSL Description

In this section, we first describe the basic SSSL scheme in a PIN entry scenario, that is, in a scenario where the human user enters a purely numerical secret.

A. Secure PIN-entry with SSSL

As mentioned in Section I, the SSSL scheme falls in the second class of PIN-entry methods, i.e., it is designed to work in the partially observable model where the adversary can only partially observe the PIN-entry procedure. SSSL implements the challenge-response paradigm and comprises three major components: (i) *a protected channel* ensuring secrecy and integrity of challenge values, (ii) *an SSSL table* - a table of digits from 1 to 9 organized in such a way that each digit i is an immediate neighbor to the other 8 digits from the set $\{1, 2, \ldots, 9\}$ (Figure 1(a)), and (iii) a set of *response buttons* (Figure 1(b)).

The SSSL method proceeds as follows. Let us assume that a user wants to authenticate to a computer using the following PIN: 46548. Let us denote the PIN digits as $d_0 = 4$, $d_1 = 6$, $d_2 = 5$, $d_3 = 4$ and $d_4 = 8$. The computer will display the SSSL table and the response buttons on its screen as shown in Figure 1. These two components of SSSL are public (observable by an adversary). At time instant t_0, the user will receive a random challenge (one digit long) c_0 selected from

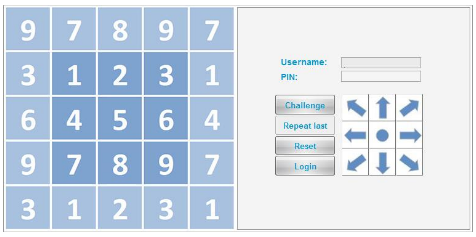
$\{1,\ldots,9\}$. Let us assume $c_0 = 9$ in our example. The user will receive the challenge over a protected channel (e.g., over *earphones* plugged into the computer). We assume that the adversary cannot learn this challenge (the partially observable model). In Sections V we discuss different ways to implement the protected channel (earphones) and we asses the security of SSSL in different adversarial models. We show that it is possible to implement such a protected channel in a way completely transparent to the user, such that SSSL remains secure even in the strong adversarial model (active attacks, fake computers and ATMs, etc.). Getting back to our example. The user looks up in the darker area of the SSSL table (Figure 1(a)) and locates (visually) the first digit of his/her PIN, $d_0 = 4$. The user then locates (visually) the challenge $c_0 = 9$ in the immediate (one-hop) neighborhood of previously located digit $d_0 = 4$. Note that this is always possible, as the digits in the SSSL table are arranged in such a way that each digit i (located in the darker area) is an immediate neighbor to all the other digits from the set $\{1, 2, \ldots, i-1, i+1, \ldots, 9\}$. Finally, the user answers the challenge by clicking a response button (Figure 1(b)) that shows the relative position of the challenge c_0 with respect to the corresponding PIN digit p_0. In our example, the user clicks the "south-west" arrow, that is, he/she responds with $r_0 =$ ⇙. It is easily seen that the response r_0 unambiguously links the challenge with the corresponding PIN digit. Note that adversary can observe the user's response r_0.

At this stage the first PIN digit has been entered and the whole procedure repeats for the remaining PIN digits. For example, at time instant $t_1 > t_0$ the user receives the challenge $c_1 = 6$ to enter the PIN digit $p_1 = 6$. Therefore, he/she responds by clicking the center button (Figure 1(a)), that is, r_1 = ○. The whole PIN-entry procedure is summarized in the following table:

time	t_0	t_1	t_2	t_3	t_4
PIN	4	6	5	4	8
challenge values	9	6	2	1	6
user's response	⇙	○	⇑	⇑	⇗

Note that SSSL does not require any numerical computation on the part of the human user. Moreover, the number of challenge-response rounds equals the size of the PIN. It is these two features that make the SSSL easy to use and very user-friendly. We asses the security of the SSSL method in Section V.

The size of the PIN space. The digits in the SSSL table are arranged in a special way in order to ensure that each digit i is an immediate neighbor to all the other digits from the set $\{1, 2, \ldots, i-1, i+1, \ldots, 9\}$. The price that we have to pay to accomplish this is the reduced PIN entropy. Indeed, in our solution every PIN digit can take one out of 9 values compared with one out of 10 in classical methods. In order to compensate for this loss, we suggest using somewhat longer PINs. For example, in our experiments each user has been given a 5 digits long PIN compared to 4 in classical solutions; note that $9^5 > 10^4$. As we report in Section IV, this increase did not affect significantly the usability of our scheme, as the average login time was only 8 seconds. To the best of our knowledge, no other secure PIN-entry method has such a short login time; e.g., [20] reports 32 seconds for five personal images.



9	7	8	9	7
3	1	2	3	1
6	4	5	6	4
9	7	8	9	7
3	1	2	3	1

Username:
PIN:

Fig. 2. The web application used in the evaluation of the SSSL method.

IV. Usability Evaluation of SSSL

We carried out an experiment in order to study different usability aspects of the SSSL login method. In this section, we present the results obtained from our evaluation study.

A. Evaluation Procedure

A total number of 30 participants took part in our evaluation study in a PIN-entry scenario. All the participants involved in the experiment were third year computer science students (early 20s). At the end 15 participants completed the study, with a dropout rate at 50%. This can be explained by two factors: The experiment was voluntary and the students were not paid or motivated to complete the study. The study took part during the exams period where the study was not the priority. In the experiment, each participant was asked to login using SSSL at least once a day during the period of four weeks. The participants were given a short (10 minute) tutorial on the SSSL method before the beginning of their authentication. Each of the 15 SSSL participants was given and asked to remember a 5 digit random PIN from $\{1,\ldots,9\}^5$. At the end, each participant was asked to complete a short questionnaire about the SSSL scheme.

To make the experiment scalable and to enable easier data collection, we implemented the SSSL method as a web application. Figure 2 shows the web interface used in the evaluation of the SSSL method (Section III). Each participant in the experiment had to sign in to the web page prior to performing any SSSL login. For each participant, we recorded different information such as the overall login time, the login error rate, etc. These information have been stored into a database for later processing.

As can be seen in Figure 2, in our implementation of SSSL the user (participant) himself/herself initiate the transfer of a random challenge by clicking on the "Challenge" button. Each time the participant clicks the "Challenge" button and/or enters a response via the response buttons the system (the participant's browser) logs the current time. This information is then transmitted to a central server that stores it in the appropriate data base for later processing. It is important to emphasize that a participant has two alternatives when entering his SSSL response. The user either uses a mouse and clicks on the appropriate response button or he/she enters the response directly via a keyboard. It turned out that while using a

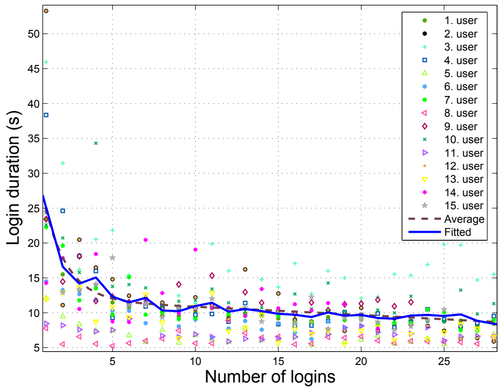


Fig. 3. The PIN-entry time: evaluation results from the experiment with 15 SSSL users for 5-digit PIN.

keyboard is a faster alternative, most of the participants still preferred to use a mouse.

In the next subsection, we present and discuss the results obtained in this experiment.

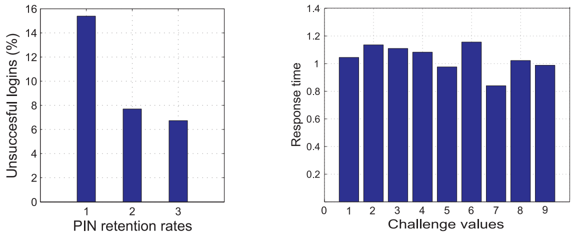
B. Evaluation Results

In Figure 3, we plot actual login times for the 15 PIN-entry participants, as well as the the average login time (over the 15 participants) for 4 consecutive weeks. Here, we take into account only successful logins. The results in Figure 3 reveal two important facts about the SSSL method. First, it has a very steep learning curve, but in a positive sense. Indeed, the login time quickly decreases already after the first few days of the experiment. Second, the overall login time is very short, only 8 seconds on average at the end of the experiment. The fastest PIN entry time was around 3.5 seconds. These two facts about the SSSL scheme are a direct consequence of its simplicity. The SSSL method does not involve mentally demanding (mathematical) operations on the side of the human user. Moreover, entering 5 digits requires only 5 challenge-response rounds.

Figure 4(a) shows an average PIN-entry error rate for the period of 21 consecutive logins organized into 3 equal periods. The participants performance improves significantly from the first to the second period, where it drops to about 7% error rate (1 error in 14 logins).

In Figure 4(b), we show *the average response time* for a single PIN-entry participant. The response time is defined as the time it takes to the user to hear the challenge and in turn to enter his/her response. As can be seen from this figure, the observed participant spent on average 1 second per PIN digit. It is interesting to note that this participant has chosen to use a keyboard for entering SSSL responses. As we already noted, a keyboard is a much faster alternative compared to a mouse. Nevertheless, most of the participants have chosen to use mouse in the experiment. The average response time over all participants is around 1.6 seconds.

Figure 4(b) reveals another interesting fact about SSSL (and other cognitive authentication methods): the response time is a function of a challenge value. We will discuss some important implications of this fact in Section V.



(a) (b)

Fig. 4. a) The average PIN-entry error rate (%) for 15 users for 21 consecutive logins divided into 3 periods. b) Histogram showing the user's response time for different challenge values and the PIN digit 7.

TABLE I
THE TABLE SUMMARIZES THE QUESTIONNAIRE FROM 15 PARTICIPANTS PROVIDED AT THE END OF THE STUDY.

Time to enter a 5-digit PIN					
Grades	1	2	3	4	5
Participants number	4	5	6	0	0
Easy to learn					
Answers	Y	Y/N	N		
Participants number	11	4	0		
Use in high-security situations					
Answers	Y	N			
Participants number	15	0			

Questionnaire. At the end of our study, students were asked to complete a short questionnaire about the SSSL. The participants were asked to rate the ease-of-use and learn of SSSL. From 15 users, 11 of them answered that our method was easy to learn and use. The others answered with moderately hard(i.e., YES/NO).

The participants were also asked to rate the authentication times (from $1 - 5$, 1-acceptable, 5-slow). Recall, students had two alternatives when entering SSSL response, via a mouse or via a keyboard. The participants who used the keyboard achieved smaller total times, and answered 1 or 2. On the other hand the participants who used the mouse rated with 2 or 3. They were also asked to comment the experience of the SSSL scheme. Some of them answered that was "fun", "novel", they felt more "secure", 3 users noted that took them several logins to learn using keyboard, but afterwards it was easier to use. Table I summarizes the answers from 15 participants. Participants preferred a mouse for the SSSL response because it required no additional learning, as opposite to the keyboard, although the scenario with a keyboard achieved faster authentication times.

Finally, the participants agreed they would use SSSL in places which require higher security, like bank transactions or in public places.

V. Security Analysis

In this section, we assess the security of the SSSL method in different attacking scenarios and in the face of different threats (timing attacks, key-loggers, fake ATMs).

A. Camera-Recording (Passive) Adversary

A camera-recording adversary will attempt to learn SSSL-protected PINs and passwords by simply recoding the SSSL login procedure. In this adversarial model the attacker is passive and does not interact in any way with the system that a user tries to authenticate to.

Let us first consider the case in which challenge values remain unknown to the adversary during the course of an attack; later on we discuss likely attacks directed at the protected channel used for delivering secret challenges to the user. Let d, c and r denote the secret PIN digit, the secret challenge value and the public response, respectively. Moreover, let $d = 4$, $c = 9$ and hence $r = ⇙$ (see Figure 1(a)). The question that we ask ourself here is: *What does the attacker learn by observing the response* $r = ⇙$?

Assuming that d and c remain secret, all that the attacker can learn by observing the response $r = ⇙$ is that it has been generated by one of the following nine (d, c) pairs:

d	1	2	3	4	5	6	7	8	9
c	6	4	5	9	7	8	3	1	2
r	⇙	⇙	⇙	⇙	⇙	⇙	⇙	⇙	⇙

As both d and c are selected uniformly at random from $\{1, \ldots, 9\}$, each of the pairs (d, c) above is equally likely to be the true pair. In other words, the adversary learns nothing about the true secret PIN digit d. The same argument applies to any response r.

Eavesdropping on the protected channel. In an attempt to break the SSSL scheme, the attacker could try to attack the protected "challenge" channel. For example, the adversary could try to use a parabolic reflector to collect sound energy produced by earphones through which the user receives challenges. This threat can be mitigated by reducing sufficiently the volume level of an audio challenge. More advanced protection would involve sound and noise reduction techniques. In-ear monitors are a passive counterpart to active noise canceling headphones [7]. They offer portability similar to earbuds, and also act as earplugs to block out environmental noise. According to [1], canalphones may reach isolation levels of -30dB to -40dB, which implies a lower sound level of an audio challenge. Laser beam eavesdropping [16] is another potential threat. Canalpohones can mitigate this threat too. For additional protection, the user can simply cover his/her earphone with a hand.

It is interesting to note that in a recent work [6] Halperin et al. propose an audio channel to securely transfer key between an *implantable cardioverter defibrillator* and an external device (programmer).

B. Side-Channel Timing Attacks

A classic timing attack is a side channel attack in which an attacker attempts to compromise a given cryptosystem by analyzing the time it takes to execute different cryptographic operations [11]. In this section, we analyze the possibility of reducing the entropy of PINs and passwords by simply observing the user's reaction time during the SSSL procedure. This is a similar approach to [28] where Zhuang, Zhou and Tygar propose to reconstruct unknown text (a password) by recording keyboard acoustic emanations produced by the user's typing. In this section, we show that the timing attack does present a threat against the SSSL login method. We conjecture this to be the case with any cognitive authentication method that involves the human user.

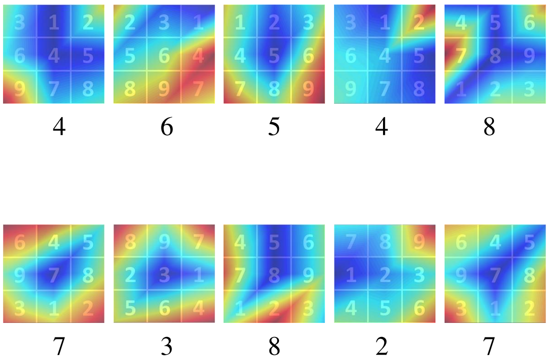


Fig. 5. Patterns showing different response times (Δt) by two users with PINs: 46548 and 73827.

We consider the passive attacker with the capability to record the user's reaction time during the course of the SSSL procedure. To accomplish this, the attacker can for example use any standard key-logging malware. The attacker records *the user's reaction time* (Δt), that is, the time period from the time instant t_c at which the user receives the challenge value to the time instant (t_r) at which the user enters his/her response. Speaking mathematically, $\Delta t = t_c - t_r$. In our initial implementation of SSSL, the user himself/herself initiates the transfer of a challenge value by clicking on the "Challenge" (Figure 2). Therefore, the attacker can easily record the time t_c; the same applies to t_r.

In Figure 5, we plot the response time Δt for two SSSL users from our evaluation study (Section IV). The patterns in Figure 5 are generated as follows. For each user we recorded 30 successful logins and for each successful login we calculated the response time Δt taken for entering a given PIN digit. Recall that for a given PIN digit, the response time is a function of a random challenge (Figure 4(b)). Since there are only 9 different challenge values and we recorded 30 login sessions, each challenge value has been generated approximately 3 times on average for the fixed PIN digit. Next, we average these response times for the fixed PIN digit and each different challenge value. Note that the attacker can do the same, since the same PIN digit and the same challenge value always imply the same public response by the user. These 9 average response times are stored in a 3×3 matrix to which we applied the MATLAB functions `contour()` and `shading()` [10] to finally obtain the patterns as shown in Figure 5. Darker area means the shorter response time.

By analyzing the patterns in Figure 5 an adversary can extract significant amount of information about the secret PINs. For example, for the first user the attacker can observe that patterns corresponding to the first and the fourth PIN digit are highly correlated. Similarly, for the second user, there is a very high correlation between the first and the last PIN digit. Based on this information, the attacker can conclude that the respective PIN digits are the same, therefore reducing the security factor from 9^5 to 9^4 (i.e., approximately $89\%!$).

In order to mitigate this threat, we can apply standard techniques such as introducing random delays in the SSSL challenge-response procedure. The price that we have to pay for this is the increased login time by a few seconds. Note that we can afford this extra time, as the overall login time with SSSL is reasonably short.

It is important to emphasize that the timing attack threat is not specific of SSSL. It is common any cognitive authentication scheme that involves the human user.

C. Active Attacks

In this section we show how to harden SSSL against this threat in a way that is completely transparent to the user.

We consider a very powerful attacker. Thus, the attacker can trick the user to login from a compromised computer or ATM that are under the attacker's control. Clearly, all login methods described in Section II, including our SSSL, can be easily broken in such a model. A safe way to protect against such attacks is to completely bypass the compromised machine. The problem however is how to accomplish this in a user-friendly and cost efficient way?

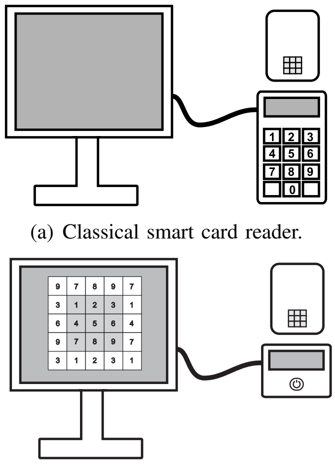
A possible approach based on SSSL applies to scenarios where an end-server authenticates the user who is connected to the server through a potentially compromised machine. A typical scenario is that of an on-line ATM, where the ATM is connected over the Internet to the secure bank server. We assume that the end-server is not compromised. As before, we assume that the user receives challenges over earphones (canalphones). The only difference is that now challenges come from the end-server. By integrating a *tamper resistant microprocessor* capable of performing cryptographic operations, the user and the secure end-server can completely bypass, for example, a compromised ATM. Indeed, the hardened earphones and the bank server will share a secret key that will be used to ensure the confidentiality and the integrity of SSSL challenges. Note that during the user authentication phase, the intermediate ATM will only provide the connectivity between the user and the bank server.

This solution is completely transparent to the user who simply follows the basic SSSL login procedure. Should the attacker tamper with cryptographically protected challenges, the hardened earphones will detect this and discard them. What is more, earphones hardened with a tamper resistant microprocessor can replace a tamper resistant smart card in the ATM scenario without affecting at all the two-factor authentication. Actually, by combining the SSSL method and hardened earphones we implement **the two-factor authentication that is cost-efficient, user-friendly and safe against shoulder surfing attacks**.

VI. Applications of SSSL

The SSSL PIN-entry method is a software-based solution and, as such, can easily be integrated into existing systems that require users to type in some secret value. For example, in this section, we show the advantages of integrating SSSL into smart-card readers (in Internet banking systems) and ceratin types of secure tokens (e.g., SSL/TLS enabled USB sticks).

SSSL in Internet Banking. The strongest authentication model in the context of Internet banking is the one where the user authenticates each transaction by "signing" it with his/her private key [8]. For this, the user is equipped with a smart card (e.g., JavaCard) and a secure online card reader (e.g., FINREAD). The smart card hosts the user's private keys (for signing and authentication) plus some certificates, while the online reader allows the user to access "securely" to his/her private key and to perform signing operations. A card reader must have certain physical properties such as, tamper resistance, a secure display, and **a secure keypad**. The sole purpose of a secure keypad is to allow the user to securely enter his/her PIN. Tamper resistance of course increases significantly the overall cost of the reader.



(a) Classical smart card reader.

(b) SSSL-enabled smart card reader.

Fig. 6. Application of SSSL to smart card readers in Internet banking systems.

Here we show that SSSL can replace a secure keypad in the above Internet banking model, in such a way that the security of the original architecture is not affected. In this way, we can basically halve the size of the original card reader as shown in Figure 6, which also reduces the overall cost of a tamper-proof reader.

In the architecture shown in Figure 6(b), the user will receive SSSL challenges on the reader's display. This design choice is motivated by the following observation. When entering his/her PIN over a secure keypad on a card reader, the user's typing can be easily recorded unless he/she covers the keypad. In the same way, the user can cover the reader's secure display when receiving SSSL challenges. The overall odds of the camera recording attacker are the same in both architectures shown in Figure 6.

The SSSL challenges c_i will be generated by the reader, while the user will enter his/her response r_i using the mouse or the keyboard of the computer to which the reader is connected (Figure 6(b)). When the user enters his/her response, the computer will simply forward it back to the card reader. Since SSSL challenges remain unknown to the computer, it cannot learn the PIN entered by the user (Section III).

By cutting the size and hence the cost of expensive smartcard readers, we can effectively stimulate wider adoption of such advanced security solutions.

"SSL/TLS+SSSL"-enabled Security Tokens. As a software-based solution, SSSL can be easily integrated with secure tokens such as SSL/TLS enabled USB sticks. In this way, we enable the user to establish a secure tunnel from practically any public machine without the fear of revealing his/her secret login credentials. Similarly, many existing solutions require the user to enter a master secret to unlock his/her USB security token. The user enters the master secret on a computer on which the token is being used, thus exposing the master secret to different forms of observation attacks. With SSSL enabled security tokens, this threat is successfully mitigated.

VII. Conclusion

We made several contributions in this paper. The fact that many prominent cognitive authentication methods are vulnerable to some form of a SAT-solver attacks, motivated our approach to the design of a secure login method.

First, we proposed a novel PIN-entry scheme called *Shoulder Surfing Safe Login (SSSL)*. Compared to existing solutions, SSSL is both user-friendly and cost-efficient. In spite of the fact that SSSL is originally developed as a PIN-entry method, we showed that it can be also easily adapted to work with graphical passwords.

Second, we carried out an experiment involving 15 participants, with the goal of studying usability aspects of SSSL. Our study confirmed that the proposed SSSL method is extremely easy to learn and use. The average login time with SSSL is only 8 seconds.

Third, we showed that a hardened version of SSSL can thwart strong active attacks in a way that is completely transparent to the user. Moreover, we showed that the threat of side-channel timing attacks has to be considered seriously in the context of cognitive authentication schemes. Finally, we showed in Section VI a number of advantages of integrating SSSL into systems like Internet banking and secure USB tokens.

Acknowledgment

The authors wish to thank Andrina Granić for discussions concerning the usability evaluation. The authors would also like to thank the anonymous reviewers for their thorough reviews and helpful suggestions.

References

[1] In-ear Monitor. http://en.wikipedia.org, last access, December 2008.

[2] M. Backes, M. Drmuth, and D. Unruh. Compromising Reflections - or - How to Read LCD Monitors Around the Corner. In *Proceedings of the IEEE Symposium on Security and Privacy (SSP), Oakland, CA*, May 2008.

[3] G. E. Blonder. Graphical Passwords. In *Lucent Technologies, Inc., Murray Hill, NJ, U. S. Patent, Ed. United States*, 1996.

[4] R. Dhamija and A. Perrig. Deja Vu A User Study Using Images for Authentication. In *Proceedings of Ninth USENIX Security Symposium*, 2000.

[5] P. Golle and D. Wagner. Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract). In *Proc. IEEE Symposium on Security and Privacy*, 2007.

[6] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-power Defenses. In *Proceedings of the IEEE Symposium on Security and Privacy (SSP), Oakland, CA*, May 2008.

[7] W. Harris. How Noise-canceling Headphones Work. http://www.howstuffworks.com, last access, December 2008.

[8] A. Hiltgen, T. Kramp, and T. Weigold. Secure Internet Banking Authentication. *Security and Privacy*, MARCH/APRIL:pages 24–32, 2006.

[9] N. Hopper and M. Blum. Secure Human Identification Protocols. In *ASIACRYPT*, 2001.

[10] http://www.mathworks.com. last access, december 2008.

[11] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO96*, London, UK, 1996.

[12] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing Shoulder-Surfing by Using Gaze-Based Password Entry. In *Proceedings of Advanced Information Networking and Applications Workshops (AINAW)*, 2007.

[13] B. Malek, M. Orozco, and A. El Saddik. Novel Shoulder-Surfing Resistant Haptic-based Graphical Password. In *Proceedings of EuroHaptics*, 2006.

[14] T. Matsumoto. Human-computer Cryptography: An Attempt. *J. Comput. Secur.*, 6(3), 1998.

[15] T. Matsumoto and H. Imai. Human Identification Through Insecure Channel. In *EUROCRYPT*, 1991.

[16] K. D. Murray. Laser Beam Eavesdropping Sci-fi Bugs? http://www.spybusters.com, last access, December 2008.

[17] Passfaces. http://www.realuser.com/. last access, December 2008.

[18] Kuber R. and Yu W. Authentication Using Tactile Feedback. In *Interactive Experiences, HCI, London, UK*, 2006.

[19] V. Roth, K. Richter, and R. Freidinger. A PIN-entry Method Resilient Against Shoulder Surfing. In *Proceedings of 11th ACM Conference on Computer and Communications Security (CCS)*, 2004.

[20] H. Sasamoto, N. Christin, and E. Hayashi. Undercover: Authentication Usable in Front of Prying Eyes. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2008.

[21] J. Thorpe, P. C. van Oorschot, and A. Somayaji. Pass-thoughts: Authenticating With Our Minds. In *Proceedings of the Workshop on New Security Paradigms*, 2005.

[22] C.H. Wang, T. Hwang, and J.J. Tsai. On the Matsumoto and Imai's Human Identification Scheme. In *EUROCRYPT*, 1995.

[23] D. Weinshall. Cognitive Authentication Schemes Safe Against Spyware (Short Paper). In *Proc. IEEE Symposium on Security and Privacy*, 2006.

[24] S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon. PassPoints: Design and Longitudinal Evaluation of a Graphical Password System. In *International Journal of Human Computer Studies*, 2005.

[25] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget. Design and Evaluation of a Shoulder-surfing Resistant Graphical Password Scheme. In *Proceedings of the working conference on Advanced visual interfaces (AVI)*, 2006.

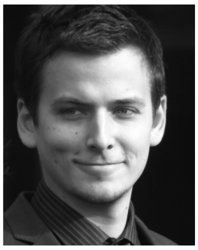
[26] G. T. Wilfong. Method and Appartus for Secure PIN Entry. In *Lucent Technologies, Inc., Murray Hill, NJ, U. S. Patent, Ed. United States*, 1999.

[27] H. Zhao and X. Li. S3PAS: A Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication Scheme. In *Proceedings of Advanced Information Networking and Applications Workshops (AINAW)*, 2007.

[28] L. Zhuang, F. Zhou, and J. D. Tygar. Keyboard Acoustic Emanations Revisited. In *CCS: Proceedings of the 12th ACM conference on Computer and communications security*, 2005.

Mario Čagalj received the Dipl. Ing degree in computer science and electrical engineering from the University of Split, Croatia, in 1998, and the PhD degree in communication systems from the Ecole Polytechnique Federale de Lausanne (EPFL) in February 2006. In 2000 and 2001, he completed the Predoctoral School in Communication Systems, EPFL. From 2001 to 2006, he was a research assistant in the Laboratory for Computer Communications and Applications (LCA) at EPFL. Since September 2006, Mario Cagalj is an Assistant Professor at Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture (FESB), University of Split, Croatia. His research interests include the design and analysis of security protocols for wireless networks, applied cryptography, applications of game theory to wireless (and wired) networks, and the design of energy-efficient communication protocols for wireless networks.

Toni Perković received the Dipl. Ing. degree in telecommunications and electrical engineering from the University of Split, Croatia, in 2007. He is currently working toward the Ph.D. degree at the Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture (FESB), University of Split, Croatia. His research interests include the usability, design and analysis of security protocols for wireless networks, the usability and design of the secure authentication protocols. He is a member of the IEEE and the IEEE Computer Society.



Nikola Rakić received the Dipl. Ing. degree in computer science from the University of Split, Croatia, in 2008. He is currently employed at zoomMediaPlus, Inc. as software engineer. He has been working on several mobile application development projects. He lives and works in Zagreb, Croatia.