Shoulder-Surfing Safe Login in a Partially Observable Attacker Model (Short Paper)

Toni Perković[1], Mario Čagalj[1] and Nitesh Saxena[2]

[1] FESB, University of Split
[2] Polytechnic Institute of New York University

Abstract. Secure login methods based on human cognitive skills can be classified into two categories based on information available to a passive attacker: (i) the attacker *fully observes* the entire input and output of a login procedure, (ii) the attacker only *partially observes* the input and output. Login methods secure in the fully observable model imply very long secrets and/or complex calculations. In this paper, we study three simple PIN-entry methods designed for the partially observable attacker model. A notable feature of the first method is that the user needs to perform a very simple mathematical operation, whereas, in the other two methods, the user performs a simple table lookup. Our usability study shows that all the methods have reasonably low login times and minimal error rates. These results, coupled with low-cost hardware requirements (only earphones), are a significant improvement over existing approaches for this model [9, 10]. We also show that side-channel timing attacks present a real threat to the security of login schemes based on human cognitive skills.

1 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* [1]. Many proposals aimed at countering the threat require the user to perform some form of cognitive tasks - so called *cognitive authentication schemes*. The problem of designing a usable cognitive PIN-entry method secure against eavesdroppers is truly challenging. Indeed, it was recently shown in [4] that the cognitive scheme proposed in [12] and all its variants are fundamentally vulnerable to attacks based on SAT solvers.

We can roughly divide existing PIN-entry methods in two classes based on 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. For example, the PIN-entry method [5] belongs to the first class (fully observable). In this class of methods, 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; a 15 digit long PIN required 166 seconds on average [5].

On the other hand, PIN-entry methods [9, 10] belong to the second class (partially observable). In this method, the user first receives a challenge via a *protected channel*,

and enters the response through a public keypad. In this class of methods, a *passive adversary* eavesdrops on all public communication between the user and the end system. In another solution described in US patent [13], the user receives a challenge in form of a random number from $\{0, 1, \ldots, 9\}$, adds modulo 10 each digit of his PIN to the digits of the random challenge, and finally enters back the outcome via a public keypad. We term this scheme the Mod10 method.

In this paper, we design a novel *Simple Table Lookup* (*STL*) login method aimed at improving the Mod10 method [13]. Unlike Mod10, our method does not require users to perform any mathematical or mentally demanding operations. It requires the user to perform nothing more than a simple table lookup. Our usability study shows that both Mod10 and STL login methods are user-friendly and have reasonably low login times. The obtained results reveal that Mod10 has slightly lower login time at the cost of a higher error rate compared to the STL. Interestingly, the major source of errors with the Mod10 method are cases in which the sum of a challenge and the PIN digit exceeds 10, which indicates that non-math oriented people might need additional assistance when using the Mod10 method. Indeed, by extending this method with a simple lookup table (referred to as *Mod10-table*) the usability study reveals that older people prefer to use the Mod10-table method rather than Mod10.

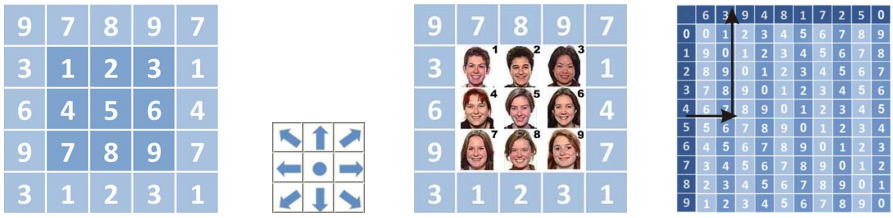
All the methods analyzed in this paper essentially implement the "one-time pad" paradigm. As such they are all perfectly secure against passive observation attacks. However, this is conditional on the fact that proper mechanisms for preventing *side-channel timing attacks* are put in place. Indeed, we show that side-channel timing attacks have to be considered seriously in the context of cognitive authentication schemes.

Other "partially observable" solutions that involve a protected challenge channel, as proposed in [9] and [10], require a fairly sophisticated, non-standard and potentially expensive hardware. In contrast, our methods only require a headset, which are commonly available or can be added with little extra cost (e.g. to ATMs).

2 Shoulder-Surfing Safe Login based on Table Look-ups

Secure PIN-entry with the STL method. STL implements the challenge-response paradigm and comprises three major components: (i) *a protected channel* ensuring secrecy and integrity of challenge values, (ii) *a simple lookup 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 $\{1, \ldots, 9\}$ (Figure 1(a)) and (iii) a set of *response buttons* (Figure 1(b)). The STL method works as follows. The computer will display the STL table on its screen as shown in Figure 1(a). Let us assume that a user wants to authenticate to a computer using the following PIN: 46548[3]. Let us denote PIN digits as $d_0 = 4$, $d_1 = 6$, $d_2 = 5$, $d_3 = 4$ and $d_4 = 8$. At time instant t_0, the user receives a random challenge (one digit long) c_0 selected from $\{1, \ldots, 9\}$, $c_0 = 9$ in our example. The user will receive the challenge over a protected channel (e.g., over *earphones* plugged into the computer). The user looks in the darker area of the STL table (Figure 1(a)) and locates (visually) the PIN digit, $d_0 = 4$. The user then locates (visually) the challenge

[3] With STL method every PIN digit can take one out of 9 values compared with one out of 10 in Mod10 and classical methods; note that $9^5 > 10^4$.



(a) (b) (c) (d)

Fig. 1. User interface: (a) In the STL 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; (c) Strengthening Passfaces graphical password system against shoulder-surfing attacks; (d) The Mod10 lookup table.

$c_0 = 9$ in the immediate (one-hop) neighborhood of previously located digit $d_0 = 4$. 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 d_0. In our example, the user clicks the "south-west" arrow, that is, he/she responds with $r_0 = \swarrow$. It is easily seen that the response r_0 unambiguously links the challenge with the corresponding PIN digit. This procedure repeats for all the remaining PIN digits. For example, for $d_1 = 6$ the user receives $c_1 = 6$ and responds with $r_1 =$ ○. Note that the STL method does not require any numerical computation on the part of the human user. Moreover, the number of challenge-response rounds equals to the size of the PIN. It is these two features that make the STL method highly usable (Section 3).

Passfaces. While there are many forms of graphical passwords, Passfaces [8] is perhaps the simplest and the most attractive solution in this category. However, Passfaces is particularly vulnerable to shoulder surfing attacks [11]. In Figure 1(c) we show that STL naturally complements Passfaces. Using STL, the threat of observation attacks against Passfaces can be mitigated.

Secure PIN-entry with the Mod10-table method. In order to assist non-mathematically oriented people, we propose to extend the Mod10 method with a simple lookup table. The Mod10 lookup table is shown in Figure 1(d). Let us assume that a user wants to enter PIN digit 4 and that she received random challenge 7. The user first looks up (visually) the digit 4 in the first column of the lookup table. Note that number 4 marks the beginning of the sixth row. Then the user looks for challenge 7 in the sixth row and moves up along the corresponding column (the column nine). The top number in this column, number 3, corresponds to the public response she has to enter back into the system[4]. Note that Mod10-table (Figure 1(d)) does not involve mathematical operations.

3 Usability Evaluation

We carried out experiments in order to study different usability aspects of the Mod10, STL and Mod10-table login methods. The usability test is divided into STL vs Mod10

[4] Note that if we order the digits in the top row as $0, 1, \ldots, 9$ we get responses that are consistent with the Mod10 additions; hence the name Mod10-table.

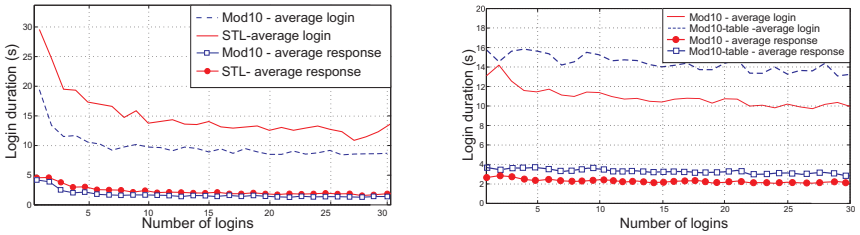


Fig. 2. The average login times and the average user's response time per PIN digit from the experiment with (left) 20 STL and Mod10 users and (right) 38 Mod10 and Mod10-table users.

	Age			Using PC (hours/week)				Using web (hours/week)			
	18-25	26-40	>40	$\geq$ 30	15-30	6-15	$\leq$5	$\geq$ 30	15-30	6-15	$\leq$ 5
STL vs Mod10	18	2	0	11	6	2	1	7	7	4	2
Mod10 vs Mod10 table	22	8	8	6	18	4	10	8	5	15	10

Table 1. Summary of the users' demographics.

and a Mod10 vs Mod10-table study. Each study took 90 minutes per user (30 minutes per method). The users would take a break, of about half an hour in between the tested methods. In the first usability study, we tested both the STL and the Mod10 methods. In Mod10 vs Mod10-table study we wanted to test whether non-mathematically oriented people find easier to use the lookup table compared to the basic Mod10 method. In both STL vs Mod10 and Mod10 vs Mod10-table study the given tests were randomized. A total of 58 participants took part in the usability study: 20 (13 males, 7 females) in the STL vs Mod10 and 38 (26 males, 12 females) in the Mod10 vs Mod10-table study. Table 1 summarizes user's demographics. The test was voluntary and the users were recruited via flyers. No one of the participants have taken part in any of our tests before.

Implementation and Test Procedure. We implemented the STL, Mod10 and Mod10-table methods as a web application. For each participant, the same test statistics (overall login time, error rates) were collected and stored in a central database. The usability evaluation for each of the PIN entry methods consisted of two phases: A *training phase* and an *authentication phase*. In the training phase participants learned how to use the respective methods (five successful logins per method). The authentication phase served as the actual test authentication methods. The participants were asked to successfully login 30 times per method; there have been no other incentives on the part of the testers (e.g. to achieve faster login times). At the end of each usability test for each login method, the users were asked to complete a post-test questionnaire. The System Usability Scale (SUS) [2] test was used to numerically express the usability of each method. The System Usability Scale (SUS) is a ten-item (Likert) scale giving a global view of subjective assessments of usability [2].

3.1 STL vs Mod10 Evaluation Study

Login Time. In Figure 2(left), we plot the average login times taken by the 20 participants over 30 successful logins. Already after the first few successful logins, the login time decreases quickly. The overall login times are only 12.5 (std $=$ 7.41)

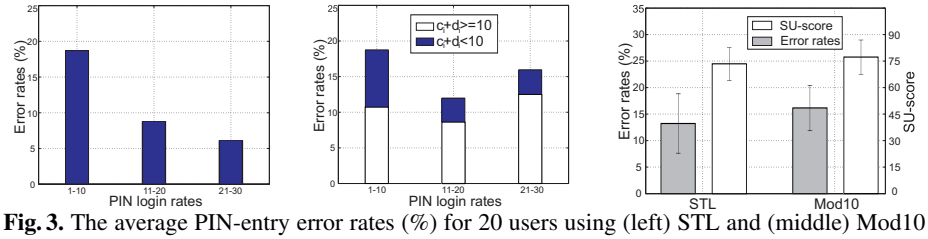


Fig. 3. The average PIN-entry error rates (%) for 20 users using (left) STL and (middle) Mod10 method. (right) The average PIN-entry error rate (%) and SU-score for 20 users.

and 9.5 (std $= 2.54$) seconds on average for STL and Mod10 methods, respectively. Higher login time with STL can be explained through the size of the PIN (5 digit PIN). The average user response time per PIN digit for both STL and Mod10 are given in Figure 2(left). Towards the end of the testing session, the average user response time for STL and Mod10 is 2.25 (std $= 1.17$) and 1.8 (std $= 0.54$) seconds, respectively.

Error Rates. Figure 3 shows average PIN-entry error rates for STL (Figure 3(left)) and Mod10 (Figure 3(middle)) methods over the period of 30 consecutive successful logins. The error rates are shown for 3 equal subsequent periods (10 successful logins per period). In the first period, the error rates are approximately the same for both methods. For the two subsequent periods, Mod10 has a higher error rate than STL. This difference is better seen in Figure 3(right). It is very interesting to observe from Figure 3(middle) that the major source of errors with the Mod10 method are cases in which the sum of the challenge and the respective PIN digit exceeds 10. This type of errors accounts for more than 70% of all errors with Mod10.

Usability score. The SU-scores are shown in Figure 3(right). The average SU-score for STL and Mod10 is 73 and 78 (out of 100). The participants evaluated Mod10 as slightly more usable because of the shorter login time (9.5 vs 12.5 seconds). The majority of participants considered both methods easy-to-use as well as secure (Table 2(left)).

Within User Analysis. Paired t-tests [7] reveal that users achieve significantly higher error rates ($p = 0.0892$) and significantly faster login time ($p = 0.0023$) using the Mod10 method as compared with the STL method. However, users did not consider this method to be significantly more usable than the STL method ($p = 0.1068$).

3.2 Mod10 vs Mod10-table Evaluation Study

Login Time. In Figure 2(right), we plot average login times for 38 participants over 30 successful logins. Already after the first few successful logins the login time decreases. The overall login times are only 10 (std $= 3.92$) and 12.5 (std $= 3.76$) seconds on average for Mod10 and Mod10-table methods, respectively. The average user response time per PIN digit for Mod10 and Mod10-table (Figure 2(right)) is 2 seconds (std $= 0.69$), and 2.7 seconds (std $= 0.72$), respectively.

Error Rates. Figure 4 shows average PIN-entry error rates for Mod10 (Figure 4(left)) and Mod10-table (Figure 4(middle)) methods. Similarly to the results of error rates from STL vs Mod10 evaluation, Mod10 method achieves larger error rates due to the fact that Mod10-table requires only a simple table lookup operations.

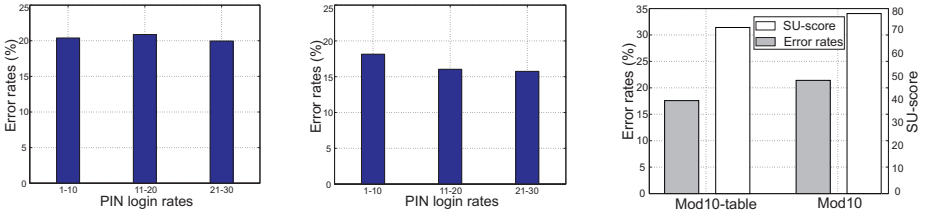


Fig. 4. The average PIN-entry error rates (%) for 38 users using (left) Mod10 and (middle) Mod10-table method. (right) The average PIN-entry error rate (%) and SU-score for 38 users.

	Using					Feel secure				
	5	**4**	**3**	**2**	**1**	**5**	**4**	**3**	**2**	**1**
STL vs	10	4	6	0	0	11	5	4	0	0
Mod10	9	4	7	0	0	9	9	0	0	0
Mod10 vs	11	10	13	3	1	14	12	4	4	4
Mod10-table	9	7	12	6	4	16	13	9	2	3

	Prefer Mod10 or Mod10-table			
	18-25	26-40	over 40	Overall
Mod10	19	6	4	29
Mod10-table	3	2	4	9

Table 2. The table summarizes the responses from users about how acceptable and secure they found using the Mod10 vs Mod10-table and STL vs Mod10 methods (5-strongly agree, 1-strongly disagree), and which method (Mod10 or Mod10-table) they prefer the most.

Usability score. The average SU-score for Mod10-table and Mod10 is 72 and 78 (Figure 4(right)). Thus, in spite of higher error rates, the participants evaluated Mod10 as slightly more usable perhaps because of the shorter login time (10 seconds vs 12.5 seconds). The post-test questionnaire results in Table 2(left) show that majority of participants considered both of the methods easy-to-use and secure. The results in Table 2(right) also indicate that young participants (age 18-25 years) tend to prefer the Mod10 method (87%). However, older participants (over 40 years), are likely to prefer (50%) the login method with a simple lookup table (Mod10-table).

Within User Analysis. Paired t-tests [7] revealed that users achieve significantly higher error rates ($p = 0.0035$), significantly faster login time ($p = 0.000021$) and significantly higher SU-score ($p = 0.01674$) using the Mod10 method as compared with the Mod10-table method. From paired t-tests we conclude that users belonging to the age group 18-25 consider the Mod10 method significantly more usable ($p = 0.0409$) due to the faster login times ($p = 0.001$) achieved with this method despite the higher ($p = 0.001$) error rate. On the other hand, the users belonging to the age group 26-40 and above 40 years do not consider the Mod10 method significantly more usable than the Mod10-table method.

Between User Analysis. For the Mod10 method, unpaired t-tests revealed that users belonging to the age group above 40 take significantly longer time to login compared to the users belonging to the 18-25 group ($p = 0.032$). The means of login times were 9.9756 and 8.9264 seconds, respectively, corresponding to the two age groups. For the Mod10-table method, unpaired t-tests revealed that users belonging to the age group 18-25 years achieve significantly lower error rates to complete the task compared to the users belonging to the group 26-40 years ($p = 0.00117$) and above 40 years ($p = 0.00891$). The means of the error rates are 13.033, 24.663 and 22.994, corresponding to the age groups 18-25, 26-40 and above 40 years.

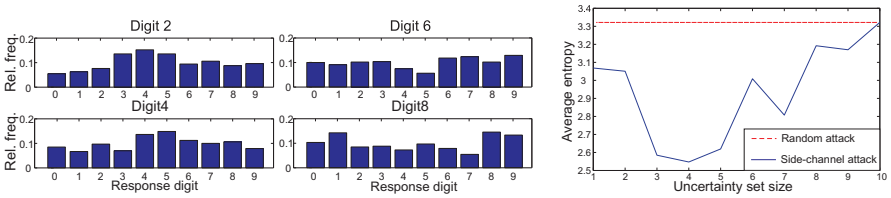


Fig. 5. (left) Relative frequency with which a given response digit appears within $\ell = 4$ fastest response digits, for PIN digits: 2, 4, 6 and 8, (right) Reduction in the average entropy due to the side-channel timing attack.

4 Side-Channel Timing Attacks

As we stated before, Mod10, STL and Mod10-table PIN-entry schemes implement the *one-time pad* paradigm. As such they are all perfectly secure against passive observation attacks. However, this is conditionally on the fact that proper mechanisms for preventing side-channel timing attacks are put in place. Side-channel timing attack represents an interesting vector of attacks on cognitive authentication schemes. A classical 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 [6]. In this section, we analyze the possibility of reducing the entropy of PINs by simply observing the user's reaction time. We consider a passive attacker capable of recording the user's reaction time during the course of the Mod10 procedure (e.g. by using key-logging malware or a simple camera). The attacker records *the user's response time* (the difference between the moment at which the user receives the challenge value and the moment at which the user enters his/her response. We saw earlier (Figure 3(middle)) that the major source of errors in Mod10 scheme are the cases when the sum of two numbers exceeds 10. Consequently, we hypothesize that these additions (over 10) have longer average response times. To verify this hypothesis, for each user (out of 38) we recorded 30 successful logins and calculated the response time taken for entering a given PIN digit. Since there are only 10 different challenge values and they are generated uniformly at random, each challenge has been generated approximately 3 times on average for the fixed PIN digit. For the fixed PIN digit we count how many users (with this PIN digit) have a given response digit within their ℓ "fastest" response digits[5]. We plot the corresponding relative frequency in Figure 5(left) for the PIN digits 2, 4, 6, 8 and for $\ell = 4$. As shown, the shortest response time (large bars in Figure 5(left)) for the respective PIN digit occurs in cases in which the users receive challenges from the set $\{0, 1, \text{or } 2\}$; i.e. for "easy" additions with challenge 0, 1 and 2. Based on these observations, we constructed a set of unique features for each PIN digit. Then, we applied standardized pattern matching techniques (k-nearest neighbor [3]) to a test group of users to classify their PIN digits. The designed algorithm outputs a reduced set of most likely PIN digits. As can be seen from Figure 5(right), with this method we can reduce the entropy from $\log_2 10 \approx 3.3$ bits to 2.55 bits. Similarly, in the STL method the attacker can also observe the correlation between two (or more) equal

[5] Here we refer to ℓ response digits of the corresponding user, which have shortest response times.

PIN digits of the respective user and thereby reduce the entropy of the PIN digit. It is important to emphasize that the side-channel timing attack is not specific to Mod10 and STL only. It is common to any cognitive authentication scheme.

5 Conclusion

We made several contributions in this paper. We studied three simple PIN-entry methods – Mod 10, STL and Mod10-table – designed for the partially observable attacker model. All methods are challenge-response protocols that allow a user to login securely in the presence of an adversary who can observe user input (the response values). A notable feature of the Mod10 method is that the user needs to perform a very simple mathematical operation, whereas, in the other two methods, STL and Mod10-table, the user performs a simple table lookup. Our usability evaluation indicates that all methods have reasonably low login times and minimal error rates. Although Mod10 method is slightly faster compared to STL and Mod10-table, it exhibits slightly higher error rates, and was found to be most suitable for younger users. We showed that the threat of side-channel timing attacks has to be considered seriously in the context of cognitive authentication schemes.

Acknowledgment. We would like to thank our shepherd Philippe Golle, and our anonymous reviewers for their thorough reviews and helpful suggestions.

References

1. M. Backes, M. Drmuth, and D. Unruh. Compromising Reflections - or - How to Read LCD Monitors Around the Corner. In *IEEE Symposium on Security and Privacy*, May 2008.
2. J. Brooke. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*, London, 1996.
3. T. Cover and P. Hart. Nearest Neighbor Pattern Classification. In *Information Theory, IEEE Transactions on*, volume 13, pages 21–27, 1967.
4. P. Golle and D. Wagner. Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract). In *Proc. IEEE Symposium on Security and Privacy*, 2007.
5. N. Hopper and M. Blum. Secure Human Identification Protocols. In *ASIACRYPT*, 2001.
6. P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO*, London, UK, 1996.
7. N. O'Rourke, L. Hatcher, and E. J. Stepanski. *A Step-by-Step Approach to Using SAS for Univariate and Multivariate Statistics, 2nd Edition*. SAS Institute Inc., 2005.
8. The Science Behind Passfaces. http://www.realuser.com/.
9. Kuber R. and Yu W. Authentication Using Tactile Feedback. In *Interactive Experiences, HCI, London, UK*, 2006.
10. H. Sasamoto, N. Christin, and E. Hayashi. Undercover: Authentication Usable in Front of Prying Eyes. In *ACM Conference on Human Factors in Computing Systems*, 2008.
11. F. Tari, A. Ant Ozok, and S. H. Holden. A Comparison of Perceived and Real Shoulder-surfing Risks Between Alphanumeric and Graphical Passwords. In *SOUPS*, 2006.
12. D. Weinshall. Cognitive Authentication Schemes Safe Against Spyware (Short Paper). In *Proc. IEEE Symposium on Security and Privacy*, 2006.
13. G. T. Wilfong. Method and Appartus for Secure PIN Entry. Lucent Technologies, Inc., Murray Hill, NJ, U. S. Patent, Ed. United States, 1999.