Optimizing Decision Tree Attack on CAS Scheme

Toni PERKOVIĆ, Marin BUGARIĆ, Mario ČAGALJ
FESB, University of Split, 21000, Croatia
marin.bugaric@fesb.hr

***Abstract*—In this paper we show a successful side-channel timing attack on a well-known high-complexity cognitive authentication (CAS) scheme. We exploit the weakness of CAS scheme that comes from the asymmetry of the virtual interface and graphical layout which results in nonuniform human behavior during the login procedure, leading to detectable variations in user's response times. We optimized a well-known probabilistic decision tree attack on CAS scheme by introducing this timing information into the attack. We show that the developed classifier could be used to significantly reduce the number of login sessions required to break the CAS scheme.**

***Index Terms*—access control, authentication, classification algorithms, computer security, human factors.**

I. Introduction

The emergence of a whole variety of attacks on various authentication schemes has led to the fact that designers of such schemes spend more and more time trying to improve existing schemes or invent new ones. PIN/password-based authentication schemes, although still the most popular way of user authentication, have proven to be vulnerable to different forms of observation attacks, such as shoulder-surfing [1], keylogging or camera recording attacks [2]. Newly proposed authentication schemes are in most cases challenge-response protocols, where users respond to given challenges in one or more challenge-response rounds.

Appearance of new types of authentication protocols had as a result the invention of even more complicated attacks (such as a side-channel attack). In such attacks, the adversary exploits various information gained from the actual implementation of the system, either physical or virtual. Timing attack, as an example of a side-channel attack, focuses on the time the user invests in order to complete various computations required by the system.

Protocols vulnerable to this kind of attack have one thing in common: subtle variations in the cognitive difficulty of challenges given to users, lead to observable variations in users' response times. Designers of new authentication protocols often invest too much effort on maintaining the usability of the system, unfortunately they often neglect this asymmetry in the user's cognitive load. This, along with the fact that the users respond to given challenges immediately after they calculate the response, results in unsecured protocols. In our recent work [3] we have shown how timing information could be used in cognitive-asymmetry side-channel attacks. Unfortunately, not only the asymmetry in the user's cognitive load can result in a successful timing attack, but also the asymmetry of the physical user interfaces [4] (such as keyboards) and, as we will show in this paper, of the virtual interfaces [5] and graphical layouts.

In order to demonstrate the feasibility of the timing attack that results from the asymmetry of virtual interfaces and graphical layouts, we will use the asymmetry found in a well-known high complexity CAS scheme [6]. In this protocol, the user's secret consists of 30 images extracted from the pool of altogether 80 images. In each challenge-response round, the user is presented with an 8 × 10 grid consisting of randomly scattered 80 images, as can be seen in Fig. 1. In every challenge-response round the user visually forms the path from the upper left corner of the grid (the set of images) to one of the responses based on simple rules and his secret password.

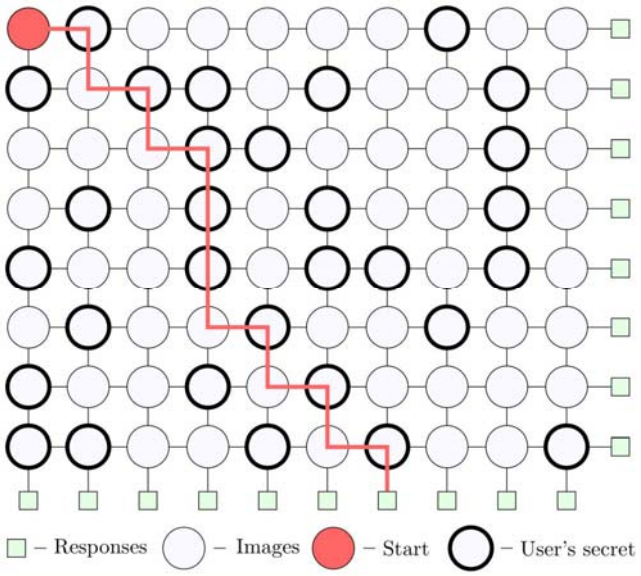

Figure 1. The graphical layout of the high-complexity CAS scheme

From the design of the graphical layout of the high-complexity CAS scheme, it is obvious that not all responses require an equal number of steps the user has to make before obtaining the response. This is of great importance, since differences in the number of steps required to obtain the response are highly correlated with (observable) variations in the user's response times. In this paper we will demonstrate how to build a classifier that exploits these variations and allows the adversary to successfully perform the timing attack on high-complexity CAS scheme. Although we agree that CAS scheme is already insecure against SAT solver attacks [7] and probabilistic decision tree attacks [8], in this paper we show how to increase the speed of such attacks (reduce the number of login sessions the attacker has to observe) by observing the timing information in every challenge-response round.

We show that with timing information it is possible to

Digital Object Identifier 10.4316/AECE.2016.02010

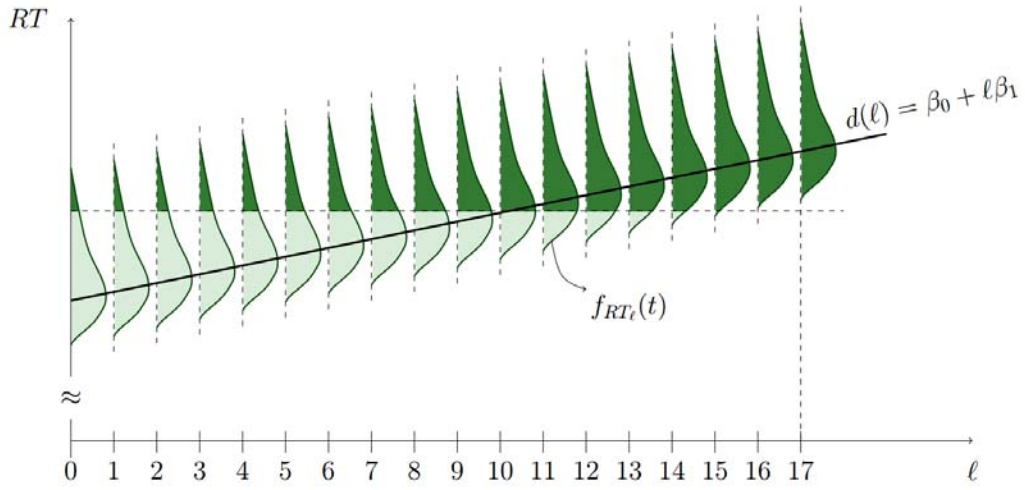

Figure 2. An example of the proposed human model running CAS authentication scheme. In the simplest version of this human model where $d(\ell) = \ell$ and $E[RT_\ell] = \partial \cdot \ell$ we get the same equation for the expected user's response time as in (X): $E[RT] = (0.3694 + 0.0383 \cdot k) \cdot \bar{\ell}$, where $\bar{\ell}$ is the average decision path length

reduce the number of observations required to discover all secret elements with high probability by 150 challenge-response rounds, i.e. by 15 login sessions! We also believe that our attack can be extended to work on plethora of other cognitive authentication schemes that have asymmetry in virtual interfaces and graphical layouts.

II. Related Work

There is a body of research focused on designing secure PIN-entry schemes in face of the threat posed by observation attacks [9-11]. Some researchers design their solutions secure against a short-term memory attackers, using the fact that the human short-term memory has a limited capacity. In these solutions, the user is requested to give answers to a set of challenges during a login procedure. However, the authentication scheme is designed in a way that the user can easily respond to questions, whereas the cognitive capacity exceeds the attacker (human) memory.

Roth et al. [9] developed a scheme where digits are placed in distinct sets. To authenticate, the users have to repeatedly indicate the respective target set. Similar approaches were proposed by Zezschwitz et al. [12], De Luca et al. [11] and Lee [10], exploiting the limitations of the human short-term memory. Unfortunately, scheme proposed by Roth et al. [9] is insecure against recording attacks, as indicated by Kwon and Hong [13]. As a results of their experimental and theoretical analysis, they have proposed a new scheme secure against camera-based recording attacks.

Bianchi et al. proposed a nonvisual unimodal schemes, which uses hidden audio and vibration challenges for user authentication [14]. In another work by Bianch et al. Spinlock, Colorlock and Timelock schemes achieve faster times than Spinlock [15-16]. However, all three schemes have partial leakage of information in the observation attack.

Other solutions assume the existence of stronger attacker that can record the complete login session and try to recover the user's secret PIN/password [17-21]. However, all these schemes are not usable in practice since they all take large authentication time.

Designing a scheme secure against even a simple passive attack in a model where the attacker can observe both challenges and responses appears to be challenging [6], [14]. In Cognitive authentication scheme (CAS) [6], a user mentally computes a path formed by his set of secret images, and gives an answer based on that (mentally) computed path. CAS scheme is vulnerable to SAT solver attacks [7] and probabilistic decision tree based attack [8].

To speed up the login process and keeping the solution safe against observation attacks, some solutions rely on the presence of secondary-based (unobservable) channels. Kuber and Yu [22] and Sasamoto et. al. [23] use a tactile channel as a secure hidden challenge channel.

In VibraPass authentication system user receives hidden challenges via his mobile phone [24] (a vibration telling the user to enter true/false response). Hidden challenges are used to avoid possible manipulations by the attacker. The authors mentioned confused waiting as a potential timing attack.

In the Undercover solution the user simultaneously receives a visual challenge and a hidden tactile challenge via a protected channel and authenticates by answering correctly to several challenges. One of the authors of Undercover, Hasegawa et. al. [25] proposed two alternative designs to Undercover [23], one of which uses an audio channel as the carrier of the hidden challenges. However, the proposed solution is prone to intersection attacks [5]. Unfortunately, Undercover is also prone to intersection attacks as independently demonstrated in [5] and [8]. This problem can be easily mitigated if challenges are fixed instead of being randomized [5].

In recent paper Asghar et al. [26] show how to attack CG protocols [27] and a modified version of Foxtail protocol [18], by transforming them into a system of linear congruences. A as result, the attack places the upper bound on the number of allowable sessions to recover the secret.

Recent discoveries in human user's nonuniform behavior have shown that a lot of information about the user's secret can be discovered only by observing user's response times to (hidden) challenges. For example, timing attack on Undercover [23] is based on design flaws that lead to nonuniform user's behavior that results from the asymmetry of virtual interfaces, i.e. asymmetric graphical layouts. Recently, a timing based side-channel attack has been found

on HB protocols and Mod10 scheme [3] that exploit the difference in response times that result from the difficulty of cognitive operations while calculating the user's response, i.e. cognitive asymmetry side-channel attacks.

III. High-Complexity CAS Scheme

High-complexity CAS scheme is based on *k-out-of-n paradigm*, where the user (*U*) is assigned with *k*=30 images (which form the user's secret set **s**) chosen from the pool of altogether n=80 images. In each challenge-response round, the system (*S*) forms an 8 × 10 grid from independent random permutations of the original 80 images. Positions next to terminal row and column of the grid are exit positions (squares in Fig. 1). Each exit position is associated with a number from the set [0, 1, 2, 3] that represents the response to the given challenge. Please note that every number from the set [0, 1, 2, 3] has approximately the same probability of occurrence. The user mentally computes the path and gives back the obtained response based on the Algorithm 1:

Algorithm 1 High-complexity CAS scheme

1) Starting point (the current cell) is the upper left corner of the 8 × 10 grid (colored circle in Fig. 1).
2) If the current cell belongs to the user's secret set ***s****, move down by one cell, otherwise move right by one cell.*
3) When the user reaches the exit position, respond with the given response.

IV. The Attacker Model

The attacker we consider in this paper is a *passive attacker* who can eavesdrop on all public communication between the user and the end system, and can also measure and record user's reaction time when responding to challenges given by the system.

V. Timing Attack on High-Complexity CAS Scheme

In this section, we will explain how variations arising from the asymmetric graphical layout of the high-complexity CAS scheme could be used to successfully perform the timing side-channel attack. Let us denote with ℓ the number of steps (movements to the right or down by one cell) that the user makes before reaching the exit position. Due to the design of the CAS scheme, the number of steps ℓ may differ with each challenge-response round. Moreover, higher the number of steps the user requires to make to reach the exit position, longer the time he will require to respond to the given challenge (as users tend to enter the response immediately after observing it). This means that for different challenges, user will with high probability respond with observably different response times RT. A passive attacker can observe and record these times, and use this knowledge to gradually learn the user's secret through the attacking algorithm we propose in this paper.

A. Modelling a Human Running CAS Scheme

We now present a general and realistic model describing the user running the CAS scheme. Let RT_ℓ be the probabilistic model for the response time associated with ℓ number of steps the user makes before reaching the exit position (either bottom or right). Equation (1) that we propose is an extended version of equation given in [8]:

$$RT_\ell = \overbrace{(0.3694 + 0.0383 \cdot k)}^{\partial:\ \text{a fixed delay}} \cdot D_\ell \tag{1}$$

where $D_\ell > 0$ represents a random penalty associated with a number of steps ℓ. Let $f_{D_\ell}(t)$ be the probability density function of D_ℓ, i.e. $D_\ell \sim f_{D_\ell}(t)$. We choose $f_{D_\ell}(t)$ to take any form appropriate for modelling human reaction times, such ex-Gaussian [28], ex-Wald, Weibull, etc. We now write:

$$f_{D_\ell}(t) = f(t \mid d(\ell), \mathbf{p}), \quad \text{with support } t > 0, \tag{2}$$

where $d(\ell)$ in equation (2) is the mean of $f_{D_\ell}(t)$, and $\mathbf{p}$ represents parameters such as variance, shape, etc.

Please recall, our main (and reasonable) assumption is that the higher number of steps ℓ the user makes before reaching the exit position, longer the time he requires to respond to the given challenge. This can be formalized in a general way if we assume that $d(\ell)$ is an arbitrary increasing positive function that is strictly increasing for at least one ℓ. We can now derive the distribution of RT_ℓ (equation (3)).

$$f_{RT_\ell}(t) = f(t \mid \partial \cdot d(\ell), \mathbf{p}), \quad \text{with support } t > 0. \tag{3}$$

To build our classifier, we must first derive the mixture distribution that characterizes the observable user's response time $f_{RT}(t)$. To do so, we must look at the exits to the bottom and to the right separately (squares in Fig. 1). It is obvious that for one exit to the bottom there are numerous paths leading to it, however, all these paths share the same number of steps ℓ (starting from the upper-left corner). More notably, ℓ number of steps uniquely identifies a single exit position at the bottom. The same applies for the exits on the right. For each exit position at the bottom, the user makes exactly r down steps, however, for each $(0 \le i \le c-1)$ there are $\binom{r-1+i}{i}$ possible ways i steps to the right exist in the path exiting to the bottom. Similarly, for each exit position to the right, the user makes exactly c steps to the right, however, for each $(0 \le j \le r-1)$ there are $\binom{c-1+j}{j}$ possible ways j steps to the bottom exist in the path exiting to the right. This leads us to our mixture distribution that characterizes the observable user's response time $f_{RT}(t)$ that we now give in equation (4):

$$f_{RT}(t) = \sum_{\ell=r}^{r+c-1} \pi_{1_\ell} f_{RT_\ell}(t) + \sum_{\ell=c}^{r+c-1} \pi_{2_\ell} f_{RT_\ell}(t) \tag{4}$$

where $\sum_{\ell=r}^{r+c-1} \pi_{1_\ell} + \sum_{\ell=c}^{r+c-1} \pi_{2_\ell} = 1$, and:

$$\pi_{1_\ell} = \frac{\binom{\ell-1}{\ell-r}\binom{n-\ell}{k-r}}{\binom{n}{k}}, \pi_{2_\ell} = \frac{\binom{\ell-1}{\ell-c}\binom{n-\ell}{n-k-c}}{\binom{n}{k}}, \tag{5}$$

where π_{1_ℓ} and π_{2_ℓ} in equation (5) represent probabilities that the user will reach a certain exit position (uniquely identified by the number of steps ℓ) to the bottom or to the right, respectively (Fig. 1). In other words, each of these probabilities represents the total number of paths consisting of exactly ℓ steps, multiplied by the probability of occurrence of such paths (which is the same for all of them). Please note that, in order to stay general, we take into account "the human factor" by introducing the aforementioned distribution $f_{RT_\ell}(t)$ into equation (4).

The expected value of RT can be defined as follows: $E[RT] = \sum_{\ell=r}^{r+c-1} \pi_{1_\ell} E[RT_\ell] + \sum_{\ell=c}^{r+c-1} \pi_{2_\ell} E[RT_\ell]$. We show one such human model running the CAS scheme in Fig. 2. In this case we assume that $d(l) := \beta_0 + \ell\beta_1$ and $\beta_1 > 0$.

B. Attacking Algorithm

The timing attack we propose in this section is an enhanced version of the probabilistic decision tree attack that in addition takes into account the timing information. The attacker creates and updates a *t-element score* table (that contains *t* elements) by observing the user's responses to the given challenges, along with the time the user takes to respond. Basically, for every challenge-response observation, the attacker indicates all possible candidate decision paths leading to the observed response. For every possible candidate decision path, the attacker assigns a probability according to the following equation:

$$p_{1_\ell} = \frac{\binom{n-\ell}{k-r}}{\binom{n}{k}},\ p_{2_\ell} = \frac{\binom{n-\ell}{n-k-c}}{\binom{n}{k}}, \qquad (6)$$

where p_{1_ℓ} in equation (6) denotes a probability assigned to every path out of $\binom{r-1+i}{i}$ paths that reach *i*th exit at the bottom of the challenge matrix ($0 \le i \le c-1$), whereas p_{2_ℓ} denotes a probability assigned to every path out of $\binom{c-1+j}{j}$ paths that reach *j*th exit on the right within the challenge matrix ($0 \le j \le r-1$). Please note that probabilities p_{1_ℓ} and p_{2_ℓ} are slightly corrected compared to the initially proposed probabilities given in [8]. More precisely, the probabilities used in [8] were: $p_{1_\ell} = \frac{k^r (n-k)^{\ell-r}}{n^\ell}$, $p_{2_\ell} = \frac{(n-k)^c k^{\ell-c}}{n^\ell}$. According to [8], if the corresponding individual element belongs to the secret set within a decision path, it will be assigned a uniform probability *k/n*, otherwise it will be assigned a probability 1-*k/n*. In our scenario the values of such probabilities depend on the position within the challenge matrix. For example, if the corresponding individual element belongs to the secret set within a decision path, and is placed in the row 3 and column 4, then its probability will be 28/77. This means that from the overall *n*=80 elements, there still have left 77 candidates while traveling through the decision path (form the upper left corner), but also 28 overall candidate images for the secret vector, since user has already moved two positions down (as can be seen in Fig. 1). As a final result the average decision path length has been slightly changed as compared to the average path length being 14.5539 in [8].

In the case of a high-complexity CAS scheme, where *k*=30, *n*=80, *r*=8 and *c*=10, there are 43758 possible decision paths in total, with average decision path length being $\bar{\ell} = 14.7318$, obtained from the following equation:

$$\bar{\ell} = \sum_{\ell=r}^{r+c-1} \binom{\ell-1}{\ell-r} p_{1_\ell} \ell + \sum_{\ell=c}^{r+c-1} \binom{\ell-1}{\ell-c} p_{2_\ell} \ell, \qquad (7)$$

where $\binom{\ell-1}{\ell-r}$ and $\binom{\ell-1}{\ell-c}$ in equation (7) represent the total number of paths consisting of exactly ℓ steps to the bottom and right, respectively.

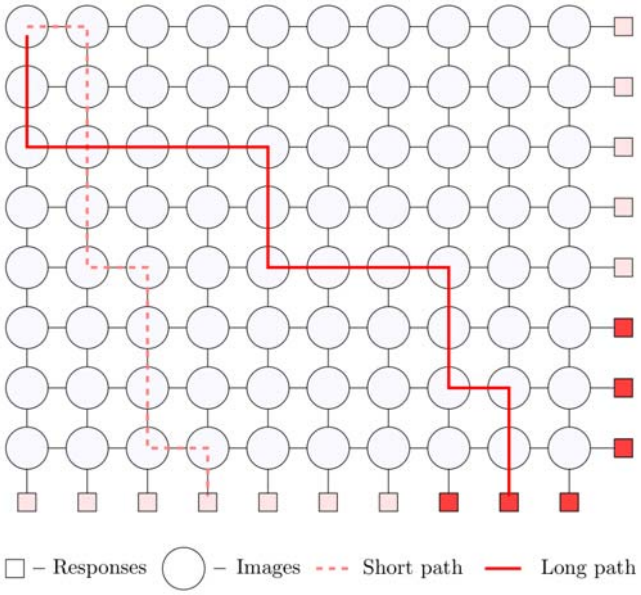

Figure 3. High-complexity CAS scheme. Light shaded responses present responses smaller than $\bar{\ell}$ steps, whereas dark shaded responses are the ones that require more than $\bar{\ell}$ steps

Please note in Fig. 3 dark shaded squares represent responses that require more steps from user to perform than the average $\bar{\ell}$ number of steps, whereas light shaded responses require from user more than the average number of steps $\bar{\ell}$. Recall, each response is associated with a number from the set [0, 1, 2, 3] that has approximately the same probability of occurrence. At the same time, the probability of reaching either light or dark shaded responses in Fig. 3 is approximately the same.

According to the human model presented in previous section, the user will require more time to respond if he has to conduct more steps before reaching the exit position. In the simplest version of the human model (as explained below Fig. 2, where $E[RT] = (0.3694 + 0.0383 \cdot k) \cdot \bar{\ell}$), we can easily calculate the expected value for the user response time *E*[*RT*]=22.3687*s*. We now present the classifier that

defines what responses should be taken into account by the adversary based on the timing information:

Classifier 1: if the user's response time RT takes more time more than $E[RT]$, take into account responses that require more than $\bar{\ell}$ steps to be obtained from the user (long paths). Otherwise, take into account responses that require $\bar{\ell}$ or less steps (short paths).

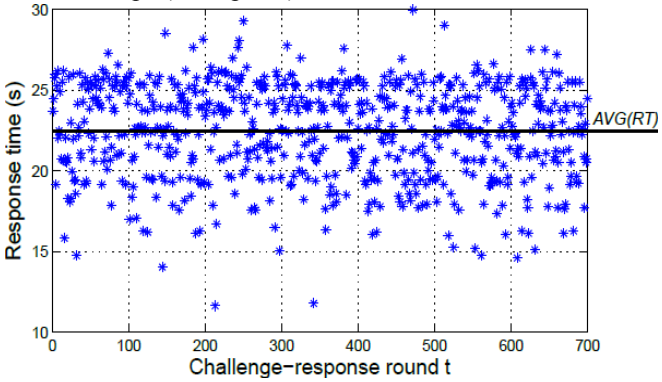

Figure 4. A trace of response times RT generated by the proposed generative model of the human behavior. We used the average of the collected response times as a threshold

Fig. 3 shows an example of short path (dashed line) and long response path (solid line). The adversary can now build and update a *t-element score table* using Algorithm 2.

Algorithm 2 Building t-element score table

1) Enumerate all consistent decision paths leading to all exit positions with the current response.

2) Eliminate those decision paths that lead to responses that do not satisfy the classifier based on timing information.

3) For all remaining decision paths X, calculate the probability of the decision path P(X), where $P(X) = p_{1_\ell}$ *or* $P(X) = p_{2_\ell}$*, depending on whether the current path ends at the bottom or to the right side of the grid, respectively.*

4) Calculate p_{sum} *- the sum of all P(X) of the remaining decision paths.*

5) Update the t-element score table according to the conditional probability $P(X \mid response) = P(X) / p_{sum}$*; i.e. in the currently observed decision path, add score by* $P(X \mid response)$ *if the image is followed by the step to the bottom, otherwise deduce the score by* $P(X \mid response)$*.*

6) Repeat the procedure until all k images from the secret set are revealed (have the highest scores).

By following Algorithm 2, the adversary can gradually learn, and finally discover the exact user's secret **s** by observing a polynomial number of challenge-response rounds.

C. Results of the Attack

Our goal is to estimate the number of rounds T required to discover all k elements of the user's secret set of n images. Our attack is the extension of the previously introduced probabilistic decision tree attack introduced in [8]. However, in our attack the adversary can, along with all public challenges and responses, also observe response times the users require to enter the response. This way, by introducing a classifier based on response times into probabilistic decision tree attack, we hypothesize that the attacker can speed up the attack and thus reduce the number of observed challenge-response rounds (login sessions).

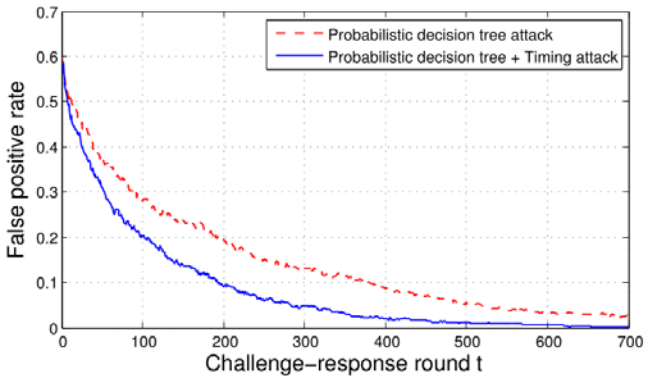

Figure 5. The average false positive rate of the high-complexity CAS scheme for probabilistic decision tree attack [8] and the probabilistic decision tree attack aided with timing attack

To model the human running the high-complexity CAS scheme we used a generative model introduced in Section IV-A. Recall, in our model we assume that the probability distribution function of the response time is given by a linear positive function $d(l) := \beta_0 + \ell\beta_1$, however in our model we used $\beta_0 = 0$ and $\beta_1 = 1$, which is the simplification that leads us to the similar equation as the one given in [8]. The density parameters in **p** (the distribution shape, variance) are modeled by using ex-Gaussian [28] distribution used for modeling human reaction times.

To estimate the efficiency of our attack compared to the probabilistic decision tree attack and to estimate the number of rounds T required to break the high-complexity CAS scheme, we implemented and compared both attacks: the probabilistic decision tree scheme attack introduced in [8] and probabilistic decision tree scheme attack aided with our timing-based classification attack. The main difference between our attack and the one introduced in [8] relies on the fact that in our observation we can differentiate between long or short response paths, i.e. paths that lead to the observed response which fall either within $\bar{\ell}$ steps, or require from user to perform more than $\bar{\ell}$ steps. By observing only fast or slow response times we consider only such decision paths that lead to all exit positions with the current response which fall within short or long response path, respectively. This way we eliminate unwanted decision paths used to calculate/update *t-element score table* (Section IV-B) and finally speed up the attack by significantly reducing the number of challenge-response rounds required to break the high-complexity CAS scheme.

Fig. 4 shows a trace comprising of 700 challenge-response rounds. Since in our human behavior model we consider a linear positive function $d(\ell)$, where the response time is a linear function of the number of steps the user has to make before obtaining the response, we used the expected response time $E[RT]$ as a threshold to differentiate between slow and fast responses (response times). More precisely, as a threshold we used the average of the generated response time of 22.46s (which is close to the expected resp. time $E[RT] = 22.3687s$ obtained from the expression

$E[RT] = (0.3694 + 0.0383 \cdot k) \cdot \bar{\ell}$, where $\bar{\ell} = 14.7318$).

We can see from Fig. 5 that the timing attack significantly increases the speed of the probabilistic decision tree attack. For comparison, in the original probabilistic decision tree attack, to discover 90% of secret elements it is sufficient to observe around T=350 challenge-response rounds (35 sessions), where by adding our timing-based classification algorithm we can reduce the number of observations to T=200 challenge-response rounds (to 20 sessions).

The above results indicate that asymmetry of virtual interfaces and graphical layouts found in a well-known CAS scheme [6] can result in nonuniform human behavior which can be further exploited to fully recover the secret **s**. As we show in this paper, timing attack in combination with probabilistic decision tree attack can increase the speed by 15 login sessions in the case of a high-complexity CAS scheme.

VI. Conclusion

In this paper we have shown the vulnerability of high-complexity cognitive authentication scheme (CAS) to side-channel timing attacks. CAS scheme is a representative of an authentication scheme in which users visually/mentally form a path to reach the response. As we show, the vulnerability comes from design flaws of visual interfaces that lead to detectable timing variations in human behavior. The attack is based on asymmetry in the visual interface of CAS scheme. We show that timing information can increase the speed of the probabilistic decision tree attack by 150 challenge-response rounds and thus significantly reduce the number of logins the attacker is required to observe. In future work we will focus on finding timing attacks on CAS-like designs and other human authentication systems.

REFERENCES

[1] F. Tari, A. A. Ozok, and S. H. Holden, “A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords,” in Proc. Symposium on Usable Privacy and Security, ser. SOUPS. ACM, 2006, pp. 56–66

[2] M. Backes, M. Durmuth, and D. Unruh, “Compromising reflections-or- ¨ how to read LCD monitors around the corner,” in IEEE Symposium on Security and Privacy, 2008, pp. 158–169.

[3] M. Cagalj, T. Perkovic, M. Bugaric, “Timing attacks on cognitive authentication schemes,” in IEEE Transactions on Information Forensics and Security, Vol. 10(3), pp. 584 – 596, 2015. doi:10.1109/TIFS.2014.2376177

[4] L. Zhuang, F. Zhou, and J. D. Tygar, “Keyboard acoustic emanations revisited,” in CCS: Proc. ACM Conf. Computer and Communications Security, 2005.

[5] T. Perkovic, S. Li, A. Mumtaz, S. A. Khayam, Y. Javed, and M. Cagalj, “Breaking Undercover: exploiting design flaws and nonuniform human behavior,” in SOUPS - Symposium On Usable Privacy and Security, 2011, p. 15.

[6] D. Weinshall, “Cognitive authentication schemes safe against spyware (short paper),” in Proc. IEEE Symposium on Security and Privacy, ser. SP, 2006, pp. 295–300.

[7] P. Golle and D. Wagner, “Cryptanalysis of a cognitive authentication scheme (extended abstract),” in Proc. IEEE Symposium on Security and Privacy, ser. S&P, 2007, pp. 66–70.

[8] Q. Yan, J. Han, Y. LI, and R. Deng, H., “On limitations of designing usable leakage-resilient password systems: attacks, principles and usability,” in Network & Distributed System Security Symposium (NDSS), Distinguisged Paper Award, 2012.

[9] V. Roth, K. Richter, and R. Freidinger, “A PIN-entry method resilient against shoulder surfing,” in Proc. ACM Conf. Computer and Communications Security, ser. CCS. ACM, 2004, pp. 236–245.

[10] M.-K. Lee, “Security notions and advanced method for human shoulder-surfing resistant PIN-entry,” in IEEE Trans. Inf. Forensics and Security, vol. 9, no. 4, 2014. doi:10.1109/TIFS.2014.2307671

[11] A. D. Luca, K. Hertzschuch, and H. Hussmann, “ColorPIN: securing PIN entry through indirect input.” in CHI. ACM, 2010.

[12] E. Zezschwitz, A. D. Luca, B. Brunkow and H. Hussmann, “SwiPIN: fast and secure PIN-entry on smartphones,” in ACM Proceedings of the Conference on Human Factors in Computing Systems, CHI. pp. 1403-1406, 2015.

[13] T. Kwon and J. Hong, “Analysis and improvement of a PIN-entry method resilient to shoulder-surfing and recording attacks”, in IEEE Trans. Inf. Forensics and Security, vol. 10, no. 2, pp. 278–292, 2015. doi:10.1109/TIFS.2014.2374352

[14] A. Bianchi, I. Oakley, and D. S. Kwon, “The secure haptic keypad: a tactile password system,” in Proc. SIGCHI Conf. Human Factors in Computing Systems, ser. CHI. ACM, 2010, pp. 1089–1092.

[15] A. Bianchi, I. Oakley, and D.-S. Kwon, “Counting clicks and beeps: exploring numerosity based haptic and audio PIN entry.” Interacting with Computers, vol. 24, no. 5, pp. 409–422, 2012. doi:10.1016/j.intcom.2012.06.005

[16] A. Bianchi, “Spinlock: a single-cue haptic and audio PIN input technique for authentication.” in Haptic and Audio Interaction Design, vol. 6851. Springer, 2011, pp. 81–90. doi:10.1007/978-3-642-22950-3_9

[17] N. Hopper and M. Blum, “Secure human identification protocols,” in Proc. Int. Conf. on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ser. ASIACRYPT, 2001.

[18] S. Li and H.-Y. Shum, “Secure human-computer identification (interface) systems against peeping attacks: SecHCI. Cryptology ePrint Archive, Report 2005/268,” 2005.

[19] H. J. Asghar, J. Pieprzyk, and H. Wang, “A new human identification protocol and coppersmith’s baby-step giant-step algorithm.” in Applied Cryptography and Network Security. Springer, 2010. doi:10.1007/978-3-642-13708-2_21

[20] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, “Design and evaluation of a shoulder-surfing resistant graphical password scheme,” in Proc. Working Conf. Advanced Visual Interfaces, ser. AVI ’06. ACM, 2006, pp. 177–184.

[21] H. Zhao and X. Li, “S3PAS: A scalable shoulder-surfing resistant textual-graphical password authentication scheme,” in Proc. Int. Conf. Advanced Information Networking and Applications Workshops - Volume 02, ser. AINAW, 2007.

[22] R. Kuber and W. Yu, “Authentication using tactile feedback,” in Interactive Experiences, HCI, London, UK, 2006.

[23] H. Sasamoto, N. Christin, and E. Hayashi, “Undercover: authentication usable in front of prying eyes,” in Proc. Conf. Human Factors in Computing Systems, ser. CHI ’08, 2008.

[24] A. De Luca, E. von Zezschwitz, and H. Hussmann, “VibraPass: secure authentication based on shared lies,” in Proc. SIGCHI Conf. Human Factors in Computing Systems, ser. CHI. ACM, 2009, pp. 913–916.

[25] M. Hasegawa, N. Christin, and E. Hayashi, “New directions in multisensory authentication,” in Proc. Int. Conf. Pervasive Computing (Pervasive), 2009.

[26] H. J. Asghar, R. Steinfeld, S. Li, M. Ali Kaafar and J. Pieprzyk, “On the linearization of human identification protocols: attacks based on linear algebra, coding theory and lattices,” in IEEE Transactions on Information Forensics and Security, Vol. 10, no. 8, pp. 1643-1655, 2015. doi:10.1109/TIFS.2015.2421875

[27] L. Catuogno and C. Galdi, “A graphical PIN authentication mechanism with applications to smart cards and low-cost devices,” in WISTP, Lecture Notes in Computer Science, 2008.

[28] R. Whelan, “Effective analysis of reaction time data,” The Psychological Record, vol. 58, pp. 475–482, 2008.