Breaking Undercover: Exploiting Design Flaws and Nonuniform Human Behavior

Toni Perković*
FESB, University of Split, Croatia
toperkov@fesb.hr

Asma Mumtaz
National University of Science and Technology (NUST), Pakistan
asma.mtz@gmail.com

Yousra Javed
National University of Science and Technology (NUST), Pakistan
yousra.javed@seecs.edu.pk

Shujun Li*
University of Konstanz, Germany
shujun.li@uni-konstanz.de

Syed Ali Khayam
National University of Science and Technology (NUST), Pakistan
ali.khayam@seecs.edu.pk

Mario Čagalj
FESB, University of Split, Croatia
mcagalj@fesb.hr

ABSTRACT

This paper reports two attacks on Undercover, a human authentication scheme against passive observers proposed at CHI 2008. The first attack exploits nonuniform human behavior in responding to authentication challenges and the second one is based on information leaked from authentication challenges or responses visible to the attacker. The second attack can be generalized to break two alternative Undercover designs presented at Pervasive 2009. All the attacks exploit design flaws of the Undercover implementations.

Theoretical and experimental analyses show that both attacks can reveal the user's password with high probability with $O(10)$ observed login sessions. Both attacks were verified by using the login data collected in a user study with 28 participants. We also propose some enhancements to make Undercover secure against the attacks reported in this paper.

Our research in breaking and improving Undercover leads to two broader implications. First, it reemphasizes the principle of "devil is in details" for the design of security-related human-computer interface. Secondly, it reveals a subtle relationship between security and usability: human users may behave in an insecure way to compromise the security of a system. To design a secure human-computer interface, designers should pay special attention to possible negative influence of any detail of the interface including how human users interact with the system.

Categories and Subject Descriptors

D.4.6 **[Security and Protection]**: Access controls, Authentication; H.1.2 **[User/Machine Systems]**: Human factors.

General Terms

Security, Human Factors

Keywords

Passwords, Observation Attack, Undercover, Tactile Device, Audio Channel, Timing Attack, Intersection Attack

* Corresponding authors.

Symposium On Usable Privacy and Security (SOUPS) 2011, July 20-22, 2011, Pittsburgh, PA, USA.

1. INTRODUCTION

Any reasonably-sensitive computer system starts from a user authentication process where a human user has to prove her identity. Contemporary systems use one or a combination of the following authentication methods: "what you know" (e.g., passwords), "what you have" (e.g., hardware tokens) or "who you are" (e.g., biometrics like our fingerprints) [39]. The user authentication process plays a key role in the security of the whole system since it is the first (and often the only) means to prevent unauthorized access by illegitimate users.

Despite the existence of more advanced user authentication methods, the simplest one based on static passwords/PINs has been the most widely adopted method since its birth in the 1960s. This is because other more advanced methods either require additional costs or decrease the usability. A salient drawback of static passwords/PINs is that they are extremely sensitive to replay attacks: they can be stolen and then simply replayed by attackers to impersonate legitimate users. In other words, when a user's identity is protected by a static password/PIN, stealing this password/PIN means stealing the user's identity.

There are many different ways to steal a user's static password/PIN. One of the simplest ways is shoulder surfing [45], which can be automated by installing hidden cameras or fake keypads or even fake terminals (like fake ATMs) [3]. Other ways of identity theft include social engineering attacks like phishing [24] and malware-based attacks like keylogging and Trojan horses [6,17]. These attacks are often described as "observation attacks" in literature, to highlight the fact that the attacker can observe communications between the user and the verifier computer.

Since the early 1990s many solutions have been proposed to fight observation attacks. With the exception of a few specialized hardware based solutions, most solutions are challenge-response user authentication protocols based on shared secrets. In each authentication session, the user is asked to give responses to a number of random challenges based on her knowledge of the shared secret. Let the shared secret, the challenges, and the responses be denoted by **S**, **C** and **R**, respectively. The user will be accepted only when $\mathbf{R}=f(\mathbf{C},\mathbf{S})$. In an observation attack, we assume that the attacker can observe both **C** and **R** but does not have access to **S**. The main task of the attacker is to solve **S** from **C** and **R**. Accordingly, the task of the user authentication system is to design a mapping f such that the attacker cannot (partially or

completely) recover **S** from **C** and **R**. Different solutions use different mappings and generate random challenges in different manner. Unfortunately, some solutions have been found insecure against multiple observations and others are not usable in terms of the average login time. A solution that is both secure and usable remains an open problem.

While most previous efforts were based on the assumption that random challenges **C** and responses **R** are fully observable to the attacker, some researchers proposed to make **C** and **R** completely or partially unobservable to increase the complexity of solving **S**.

The idea of unobservable challenges was proposed by several different groups of researchers independently in 2006 [11,27,36]. The unobservable challenge is transmitted via a tactile device that can be sensed by the user but not visible to an observer. Later at CHI'2008, Sasamoto et al. proposed another design called Undercover [43], in which part of the challenges is sent to the user via a moving trackball covered by the user's hand. Hayashi et al. claimed that Undercover is secure against multiple observations as long as the hidden challenges are truly unobservable to attackers. Hasegawa et al. from the same research group proposed two alternative designs in [21], one of which uses an audio channel as the carrier of the hidden challenges. Some other researchers have also been inspired to propose similar solutions [8,9,14,41].

In this paper, we report two attacks on the original design of Undercover in [43]. One attack can also be generalized to the alternative designs in [21]. Our attacks are based on flaws in the Undercover design, and one attack exploits human users' nonuniform behavior on how they respond to different hidden challenges. To be more precise, an average user tends to respond faster to one specific hidden challenge, thus the fastest response exposes this hidden challenge with a considerably high probability. Both theoretical and experimental analyses show that the two attacks can recover the password or part of it with considerably high probability with $O(10)$ observed login sessions. When less than ten login sessions are observed, the attacker is still able to get a reduced password space for launching a random guess attack with a better chance or a brute force attack with less complexity. The human behavior based attack was validated by a user study with a total of 28 users performed at two distinct geographical locations.

We also propose some effective enhancements to make Undercover secure against the proposed attacks. Our investigation on the enhancements revealed more nonuniformities of human behavior in interacting with the user interface, which further highlight some unique principles we need to follow for the design of secure human-computer interfaces.

The rest of the paper is organized as follows. First we give a brief survey of related work, and then detail different designs of Undercover. Afterwards we describe our attacks and demonstrate their real performance with experimental results. Then, we propose some enhancements and show how the Undercover system can be made secure against our attacks. Finally, we discuss how the timing attack may be generalized to other human authentication systems. Theoretical analyses of the two attacks are covered in the Appendix.

2. RELATED WORK

In the challenge-response protocol we described above, the key is to find a good mapping f so that the computation $\mathbf{R}=f(\mathbf{C},\mathbf{S})$ can be easily handled by an average human user while at the same time maintain the expected security level. If a hardware device is available to assist the human user, it is not difficult to choose a strong trapdoor one-way function as f, thus leading to a cryptographically strong system. Unfortunately, to protect the device from unauthorized access, a password/PIN is still needed, which is again vulnerable to observation attacks. If the hardware device is a general-purpose one like a mobile phone, then mobile malware can be another potential threat [17].

If auxiliary hardware devices cannot be used, the mapping f has to be sufficiently simple for human users to mentally calculate the correct responses. While the user has his/her own brain as the only computational resource, the attacker can access a supercomputer or even a large number of distributed computing resources (e.g., a botnet). Furthermore, to make a human authentication system usable in reality, the average login time and the error rate should be small. In contrast, the attacker can wait for a long time to break a victim's secret. Intuitively argued, it is non-trivial to find a mapping f that makes the constructed human authentication system both sufficiently secure and highly usable. Since the 1990s, there have been a number of attempts in this field, but they are either insecure or not usable in terms of average login time.

To the best of our knowledge, the first solution against observation attack was proposed by Matsumoto and Imai in 1991 [38]. The solution tries to hide the user's secret in the response by using a question alphabet and a randomized answer alphabet. Unfortunately, a few years later Wang et al. pointed out [46] that Matsumoto-Imai scheme is not sufficiently secure if the same challenge can be replayed several times by an active adversary. In addition, to achieve a high level of security, Matsumoto-Imai scheme has to use large question and answer alphabets, thus compromising usability [33]. Wang et al. also proposed an improved scheme to enhance the security, but the usability is much worse.

Matsumoto later proposed several other solutions based on inner products of secret and public vectors [37]. As pointed out in [23,33], these solutions cannot resist multiple observations because the secret vector can be solved from $O(N)$ observations, where N is the size of the secret vector.

Li and Teng proposed a new solution based on lexical shifting and matching in [35]. Although no cryptanalysis has been reported so far, its usability is not good enough since the user needs to remember a long 3-tuple secret.

Hopper and Blum proposed two solutions based on hard mathematical problems in [23]. The main problem with these solutions is again about usability: the password has to be long enough to ensure security, which makes usability relatively low. According to the user study reported in [23], the average login time of one solution (the less complicated one) is around 160 seconds, which is too long for a practical system. One solution also requires the users to make intentional errors with probability h, which may not be an easy task for them.

Sobrado and Birget proposed several novel graphical password schemes against observation attacks in [44]. One typical scheme called CHC (convex hull click) asks the user to click a random point inside a convex hull formed by three or more secret icons in the password. This scheme was later tested in a user study reported by Wiedenbeck et al. in [48]. A similar scheme called S3PAS was proposed in [49]. Two attacks on CHC were recently reported in [4]. In addition, the usability shown in [48] is not encouraging: the average login time is longer than 70 seconds. The user study was performed on a small password space of size

$C(112,5) \approx 2^{27}$, so the usability will be much worse if the password space has to be enlarged significantly.

In [34], Li and Shum suggested some basic principles of designing challenge-response protocols against observation attack. They also proposed two general protocols called Twins and Foxtail, which are based on making balanced errors and hiding direct responses to attackers, respectively. A Foxtail protocol and a graphical implementation were also reported. No cryptanalysis has been reported, but the usability of the graphical implementation is questionable, since the average login time is considerably long.

Jameel et al. proposed a new image-based solution in [25] and shortly after extended it for devices with limited display [26]. This solution is based on a hidden rule classifying an image pool into two different sets. One major problem with this design is the conflict between the automation of the classification process and security against automated attack. However, if the classification process has to be done manually by the user, the usability will be low since the image pool needs to be large.

In [47], Weinshall proposed two new solutions based on image recognition capabilities of humans. Golle and Wagner showed that both solutions are insecure against SAT (satisfiability solver) attack [11]. This attack requires only a small number of observations. In addition to the security problem, the usability of Weinshall's solutions is also questionable: the user has to remember more than 30 pictures as the password.

Bai et al. proposed a new observation-resistant human authentication scheme called PAS in [7]. PAS uses different parts of the password for different login sessions and the user's responses are obfuscated by randomized challenge and response tables. In [31], Li et al. show that part of the password can be revealed with a number of observations, thus leading to a degradation of the PAS scheme to a common OTP (one-time-password) system but with worse usability.

In [30] Lei et al. proposed a virtual password system against observation attack. They base the system on a randomized linear function. However, in [32] Li et al. pointed out that this virtual password system is not secure because an equivalent password can always be derived with only two or a few more observations.

Very recently Asghar et al. proposed a scheme in [5], which is based on a many-to-one nonlinear mapping to hide the direct response. While it is still too early to say if this solution is indeed secure, its usability does not seem to be very encouraging: the average login time was estimated to be 213 seconds, even slower than Hopper-Blum protocol proposed in [23].

Instead of trying to design a solution secure against general observation attack, some solutions relax the security requirements to target only the weakest observation attack: shoulder surfing with a very limited number (say, three) of passive observations. Examples of these solutions include some graphical passwords [12,16,42], which offers limited security against observation attack by exposing only partial information of the password in each login session. An interesting comparative study on simple shoulder surfing performed by human observers on Passfaces (a commercial graphical password scheme [40]) and textual passwords was reported in [45], which reveals that Passfaces with keyboard input is the strongest setting and strong textual password is the weakest one.

While most previous work does not require any hardware device, some other solutions employ special devices so that the challenges and/or the responses are completely or partially unobservable. Devices of this kind include eye-gazing devices [19], haptic/tactile input devices [8-10,15,21,27,28,36,43], headphone/earphone [9,21,41], mobile phones [9,14], and so on. The use of eye-gazing devices can obviously make the responses **R** invisible to human observers, but it is still possible to install hidden eye-tracking devices to read the user's eye movements. Solutions based on other partly/completely unobservable devices have close links to Undercover [43,21], the observation-resistant solution studied in this paper, and are described in the next section.

Human users are known for being unreliable to behave properly to protect their passwords [1]. Previous research has also shown that different kinds of insecure human behavior can compromise the security of some password systems [13,14,29]. However, how human behavior influence the security of many ad hoc designs of human authentication systems remains largely unexplored.

3. UNDERCOVER AND SIMILAR SOLUTIONS

Since observation attacks are mainly performed in visual form, most solutions based on unobservable challenges and/or responses aim at preventing the attacker's visible access to challenges and/or responses. Mainly two kinds of devices are employed to achieve this goal: haptic/tactile devices, and audio devices. In the following, we first introduce Undercover [43,21] in detail and then briefly overview some similar solutions [2,8-10,14,15,27,28,36,41].

3.1 Undercover: Original Design

Undercover is based on the idea of "partially observable challenges": the challenges **C** are split into public challenges $\mathbf{C}_p$ and hidden challenges $\mathbf{C}_h$. For Undercover, the relationship between the challenge, the response and the shared secret becomes $\mathbf{R}=f(\mathbf{C}_p,\mathbf{C}_h,\mathbf{S})$, where only $\mathbf{C}_p$ and **R** are observable to the attacker. Clearly, if $\mathbf{C}_h$ and **S** have the same number of possible values and the same entropy, it is possible to conceal **S** perfectly with $\mathbf{C}_h$.

The device used in Undercover is a haptic device covered by the user's palm, which is supposed to be unobservable to passive attackers. In the prototype system reported in [43], a trackball driven by two servo motors is used as the haptic device. The trackball has five different "vibrate" modes (i.e., hidden challenges): upward rotation, downward rotation, leftward rotation, rightward rotation, and vibration. The five hidden challenges are referred to as "Left", "Right", "Up", "Down" and "Center" in this paper, respectively. Each hidden challenge corresponds to a different layout of five buttons as shown in Figure 1, which is used to input the response by pressing one of the five buttons "1", "2", "3", "4" and "5" locating near the trackball. The input device of the Undercover prototype is a box as shown in Figure 2.

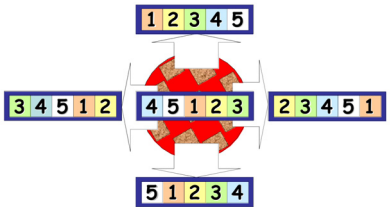

Figure 1: Five button layouts of the Undercover prototype, corresponding to the five hidden challenges (Fig. 7 in [43]).

The Undercover prototype is built on top of a graphical password scheme. The user selects five pass-pictures from an image pool to form his/her password. The system selects 23 more distractor

pictures to create the user's portfolio. Each login session is composed of seven challenges, and each challenge contains: 1) a hidden challenge transmitted via the trackball, and 2) a public challenge – four pictures and a "no pass-picture" icon shown on the monitor of the terminal computer (see Figure 3).

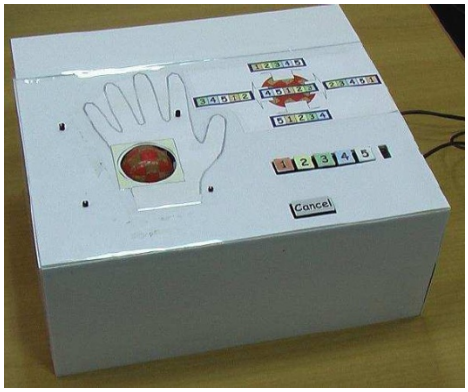

Figure 2: The input device box of the Undercover prototype (Fig. 5a in [43]).

Figure 3: A public challenge composed of four pictures and a "no pass-picture" icon (Fig. 9b in [43]).

To avoid potential security problems, the Undercover prototype system is designed so that five public challenges contain one pass-picture and the other two contain no pass-picture. Each pass-picture and distractor picture in a user's portfolio is shown *once and only once* in a login session. However, [43] does not make it clear how the seven public challenges should be generated in each login session. One may understand that the public challenges are fixed over all login sessions or randomized from session to session. In Sec. 4.3.1 we will show insecurity against an intersection attack when randomized public challenges are used.

To make a correct response to a challenge, the user needs to derive a "hidden response" first: 1) if there is a pass-picture in the public challenge, derive the hidden response (1, 2, 3, or 4) according to the position of the pass-picture among the four pictures; 2) if there is no pass-picture, the hidden response is 5 (i.e., the position of the "no pass-picture" icon). Then, the user looks for the hidden response in the button layout corresponding to the hidden challenge and presses the button matching the location of the hidden response in the correct button layout. For instance, if the hidden response is 3 (i.e., the third picture in the public challenge is a pass-picture) and the hidden challenge is "Right", the user needs to press button "2" because the hidden response appears on the 2nd button of the "Right" button layout.

Given one observed login session, the password space of the Undercover prototype is $C_7^5 \times 4^5 = 20480$, which is larger than a 4-digit PIN. Under the assumption that the hidden challenges are unobservable, the Undercover system is believed secure even if an infinite number of login sessions are observed. From an information-theoretic point of view, this is equal to the claim that the password-related information leaked in each login session is 0.

The median login time of the Undercover prototype system is 32 seconds, which is much better than previous solutions. The overall failure rate is 26%, which is rather high but could be significantly reduced after the user becomes more familiar with the system.

Hayashi et al. also proposed to show distorted pictures in the public challenge to increase the security of the system against human observers (as proposed in [22]). However, this method is not very useful for attacks performed by hidden cameras so it will not be considered in this paper.

3.2 Undercover: Alternative Designs

In addition to the original Undercover design, in [21] Hasegawa et al. from the same research group proposed two alternative designs. The main goal is to reduce the size of the system. To further simplify the design, a 4-digit PIN is used as the underlying password and the public challenge $\mathbf{C}_p$ is removed.

One design is based on six vibrating tactile devices covered by the user's five fingers and his/her palm. To generate a hidden challenge, one of the tactile devices covered by the user's five fingers will vibrate to select a column of the 2×5 matrix shown in Figure 4. The tactile device covered by the user's palm vibrates to determine the row of the 2×5 matrix. The vibrating statuses of all the six tactile devices then determine a hidden challenge – a specific element of the 2×5 matrix. Note that the ten elements of the 2×5 matrix are labeled with numbers from 0 to 9. So the hidden challenge is actually a number between 0 and 9, which will henceforth be referred to as the "hidden digit" in this paper.

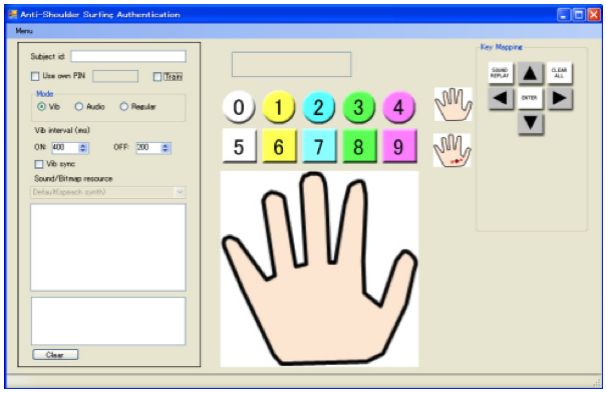

Figure 4: An alternative Undercover design (Fig. 1b in [21]).

To make a correct response, the user needs to find out his/her current PIN digit in the 2×5 matrix and then presses the four arrow buttons in Figure 4 to show a route from the PIN digit to the hidden digit. This process repeats four times so that the user can input all the four PIN digits.

Another design proposed in [21] is similar to the tactile one, but the hidden challenges are sent to the user via an audio channel, i.e., via a headphone set.

3.3 Similar Solutions

There are some other early designs for human authentication based on haptic/tactile devices with/without using the concept of hidden challenges. The main goal of these systems is mainly to resist shoulder-surfers, the simplest form of observation attacks. The solution in [36] involves pressure of a haptic pen as part of the password, thus making the password input partly unobservable to shoulder surfers. The solution called TAS (Tactile Authentication System) in [27,28] uses the VT Player tactile mouse to transmit hidden challenges to the user for entering the password without the worry of being observed. The design reported in [15] is very similar to TAS but the VT player tactile mouse is replaced by solenoids pins that can raise and lower their

positions. The solution in [2] analyzes haptic information in handwritten signatures to achieve the goal of user identification.

Some more solutions were inspired by Undercover. At CHI'2009, De Luca et al. proposed a scheme called VibraPass, which uses the user's mobile phone as the receiver of hidden challenges (a signal telling the user to make a true or false response) to avoid possible manipulation of the haptic devices by attackers [14]. De Luca et al. noticed a possible timing attack related to "confused waiting" (the user responds slower to "false" hidden challenges due to confusion) that can lead to password disclosure.

At CHI'2010, Bianchi et al. proposed a solution called Secure Haptic Keypad (SHK), which combines the tactile device and input buttons to make a uni-modal haptic password [10]. SHK can achieve similar usability to the original Undercover design in terms of average login time. In [8,9], Bianchi et al. proposed a number of other uni-modal designs based on haptic and audio hidden cues (i.e., hidden challenges) to achieve user identification. Their user studies showed that a shorter average login time and a login error rate can be achieved with the uni-modal designs.

At FC'2010, Perković et al. proposed three alternative designs based on audio channels, which have a much shorter average login time (less than 13 seconds) [41]. Perković et al. also pointed out that a side channel timing attack can reduce the PIN digit entropy, due to the user's nonuniform response time to challenges.

4. PROPOSED ATTACKS

In theory, Undercover-like solutions can achieve perfect secrecy since the shared secret **S** can be perfectly "encrypted" by the hidden challenge $\mathbf{C}_h$. Unfortunately, this is not always true because careless designs can leak information about **S** and/or $\mathbf{C}_h$. Our studies on the original Undercover design in [43] and the two alternative designs in [21] led to the discovery of such design flaws, which allow an attacker to completely break the password with considerably high probability with only $O(10)$ observed login sessions or reduce the password space if an insufficient number of login sessions are collected. We have developed two attacks: a timing attack on the original Undercover design and an intersection attack on all Undercover designs. The timing attack is based on a careless design flaw of the button layout, which leads to nonuniform behavior of the user's responses to hidden challenges. In the following, we separately describe the two attacks and their real performance verified via user studies on our own implementation of Undercover.

4.1 Our implementation of Undercover

Before introducing the two attacks, we first briefly describe how we implemented Undercover and how we collected the data to analyze the performance of the attacks.

To ease our study, we avoided using any special hardware and implemented the whole system in software. We use the audio channel to transmit the hidden challenges and Passfaces [40] as the underlying graphical scheme. The same button layouts in the original Undercover design are used. The five buttons are shown as press buttons on our software GUI. Users are allowed to make responses via mouse (by pressing one of the push buttons) or keyboard (by pressing <1>, <2>, <3>, <4> or <5>). Those changes have no influence on the security of Undercover against the proposed attacks. Figure 5 shows what a public challenge looks like in our implementation. We mask the faces in Figure 5 to avoid violating the affected people's privacy.

We performed user studies on our implementation at two universities located in two countries: the University of Split in Croatia and the National University of Science and Technology (NUST) in Pakistan. Neither university requires IRB reviews on research work involving human subjects, so the user studies were carried out without such a review. The University of Konstanz has no established policy on usable security research, but an approval from the Chair of the Ethics Committee was secured. Although a formal IRB review was not required, we took all possible measures to make sure that all legal and ethical issues we could think of were properly handled. For instance, all users were well informed in advance (before the user studies) about the purpose of the study and how the data would be processed and used in our paper. All the data collected from users was shared only among the coauthors of the paper.

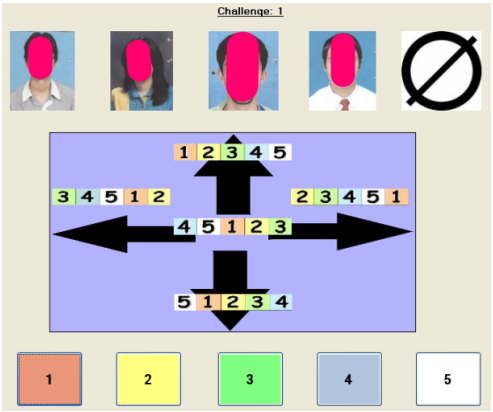

Figure 5: Our Undercover implementation.

Part of the reason why we ran the user studies in two different countries is to see if users with different cultural backgrounds and of different races share similar nonuniform human behavior that makes the timing attack a universal attack. In total, 28 users participated. All users are university students and staff members in departments of electronic engineering and computer science. Among the 28 users, 19 performed the study at the University of Split in Croatia and 9 at the National University of Science and Technology (NUST) in Pakistan. The gender ratio is 22:6 (22 males and 6 females). The ages of the 28 users range from 20 to 40 years old. All the participants were voluteers who were asked to help our research, and none of them was economically compensated/motivated, so we believe that our data is not biased towards positive results of our proposed attacks.

At the beginning of the user studies, the users were given a short tutorial of the system. A questionnaire was issued to each user to collect personal information and knowledge on computer/web technology and password security. Then, they were asked to log in at least once a day during a one-month period. To have a better control over the environment of the user studies, we set up a computer running the Undercover system in our labs and users needed to come to our labs physically for performing the logins. Users who forgot to come within 24 hours were automatically reminded via emails. Despite the reminding mechanism, not all users followed our request strictly, so at the end of our user studies different users have different numbers of recorded login sessions, but no user dropped during the course of the user studies. The minimum number of login attempts made by a user is 20, the maximum number is 66, and the median is 26.5. In total we collected 918 login attempts, among which 771 are successful ones, leading to an overall login success rate around 84%. The

login success rates of all users range from 66.67% to 100% and the median rate is 84.82%. Among all the 28 users, 18 used the keyboard as the input device while others used the mouse.

Login data of the 28 users were stored in an XML database for further processing. The login data provide information including public/hidden challenges and responses, the response time of each challenge, the overall login time, input device used to make each response, if a login attempt is successful, at which location the login attempt was made, which user made each login attempt.

Compared with the original Undercover implementation in [43], our implementation has comparable usability in terms of average login time. The median login time is 30.1 seconds, slightly shorter than the original Undercover implementation (32 seconds). Note that the average login time steadily decreased as the users became more familiar with the system. After 20 logins, the median login time decreased to 21.8 seconds.

The data collected from Croatian and Pakistani participants show some statistical differences, e.g., most Pakistani participants used mouse while most Croatia participants used keyboard as the input device, but our analysis showed that such differences do not have a major impact on the effectiveness of the proposed attacks. The different choices on input devices may be partly explained by the personal choices of our coordinators to demonstrate the system during the introduction stage: the Pakistani and Croatian coordinators used the mouse and the keyboard, respectively.

4.2 Timing Attack

4.2.1 Nonuniform human behavior in responding to different hidden challenges

Observing the five button layouts in Figure 1, we can see that the button layout corresponding to "Up" hidden challenge is "12345", exactly the original layout of the five buttons that the user needs to press. In comparison, the other four button layouts are all circularly rotated editions of the original button layout. Since users do not need to do button rotation for the original button layout, we hypothesized that they may make responses to "Up" hidden challenges faster and with a lower error rate, compared to the other four hidden challenges. Our user study confirmed this hypothesis. Figure 6 and Figure 7 show the average response times and error responses rates of all users to the five hidden challenges, respectively. Paired *t*-tests revealed that the difference between the user's responses to "Up" hidden challenges and to other hidden challenges is significant at 5% level.

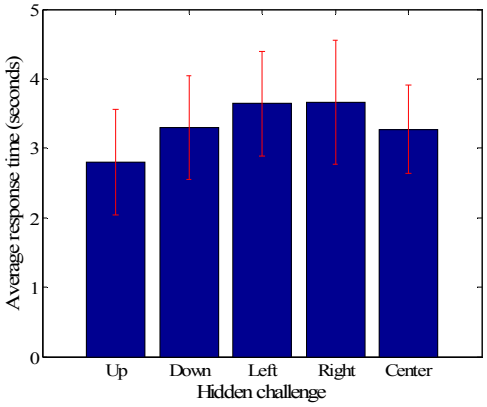

Figure 6: The nonuniform human behavior in the average response time to different hidden challenges.

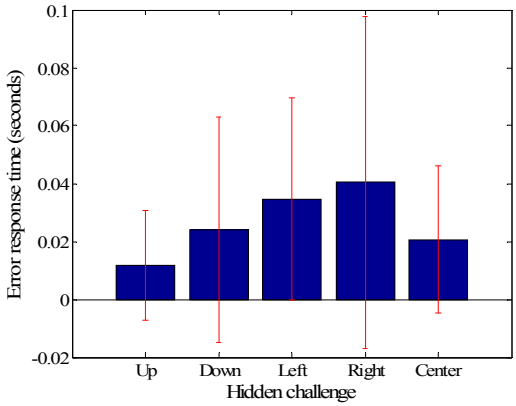

Figure 7: The nonuniform human behavior in the average error response rate to different hidden challenges.

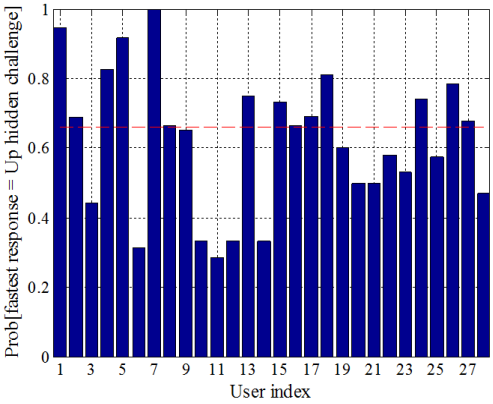

Figure 8: Probability that the fastest response in a login session corresponds to an "Up" hidden challenge.

This nonuniform human behavior in responding to different hidden challenges inspired us to propose a timing attack. Denoting the 28 pictures by 28 integers (from 1 to 28), the attack works as follows.

- *Step 1*: Create 28 counters, $C_1,\ldots,C_{28}$, for the 28 pictures, and initialize all of them with 0.
- *Step 2*: For each observed login session, take the fastest response and assume that it corresponds to an "Up" challenge. Then, if the corresponding public challenge contains a pass-picture i, increase C_i by one.
- *Step 3*: Rank all the pictures according to the values of the 28 counters, and take the top five pictures as the five pass-pictures forming the password. If there is more than one way to select the top five pictures (which can happen when some pictures have the same counter value), random shuffle all pictures with the same counter value as the fifth one, re-rank all the 28 pictures, and then take the new top five as the pass-pictures. The random shuffling process is to avoid the bias towards pictures with smaller indices.

Note that the random shuffling process in Step 3 means the timing attack may produce different results for different runs. In Section 10.1 of the Appendix, we theoretically explain why the above timing attack works and then estimate its performance.

To further improve the performance of the above timing attack, two additional measures can be further adopted: 1) negative penalty mechanism – for each distinguished decoy picture i in

Step 2, decrease C_i by one; 2) multiple fastest responses – use the fastest m=2 or 3 responses in *Step 2*. Both measures can potentially increase differences between counter values of pass-pictures and decoy pictures. Theoretical analysis of the generalized timing attack is very complex, so we only show experimental results in the next sub-section.

In the following, we describe the performance of the timing attack by applying it to the real login data collected in our user studies.

4.2.2 Real performance of the timing attack

We applied the timing attack to the login data collected in our user study, in order to verify its real performance under the following 3×2×2=12 different settings:

- Number of fastest response(s): m=1, 2, 3;
- Negative penalty mechanism: on, off;
- Login sessions used: all, successful ones only.

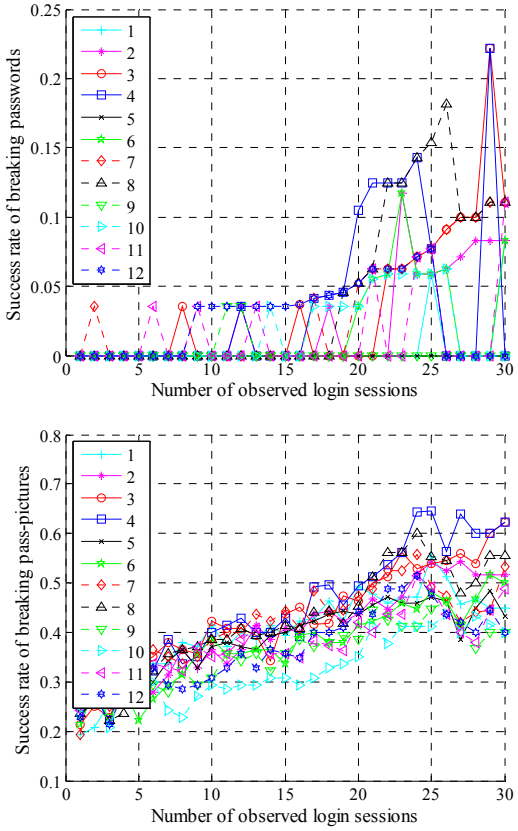

Figure 9: Success rates of breaking passwords and pass-pictures applying the timing attack to real login data.

The performance of the above 12 settings of the timing attack on real login data is shown in Figure 9. The two sub-figures correspond to p_{t5} and p^*_{t5}, respectively. It is interesting that the real performance is similar to the one estimated in the theoretical analysis. Among all the 12 settings, Settings 3 (solid line marked with "○"), 4 (solid line marked with "□") and 8 (dashed line marked with "△") have a better performance, which correspond to "m=1, without negative penalty, successful logins only", "m=1, with negative penalty, successful logins only", and "m=2, with negative penalty, successful logins only", respectively.

4.2.3 Yet another potential timing attacks

The human behavior has many different kinds of nonuniformities we may exploit. Yet another nonuniformity we noticed is that most users tend to respond more slowly to public challenges with no pass-picture. Figure 10 shows the average response times of all users with respect to the five different hidden responses. One can see that the average response time is longer when the hidden response is "5", i.e., when the public challenge does not contain a pass-picture. This phenomenon can be explained by the fact that the user has to look at all the four pictures (potentially twice) to make sure there is indeed no any pass-picture. We tested this new timing attack using the same strategy: 1) pick the m slowest responses in each login session; 2) assuming this response correspond to a public challenge with no pass-picture, decrease the counters of the four distinguished decoy pictures by one; 3) rank the counters of all pictures and pick the top five ranked pictures to form the password. Simulated attacks on the real login data did not produce good results. None of the users' passwords was completely broken, and the success rate of breaking pass-pictures ranges from 0 to 0.4. We attribute the failure of the attack to the larger variance of the response time to the public challenge with no pass-picture (which can be seen in Figure 10). Although this timing attack was unsuccessful on our dataset, it remains a potential threat since some pass-pictures may still be broken.

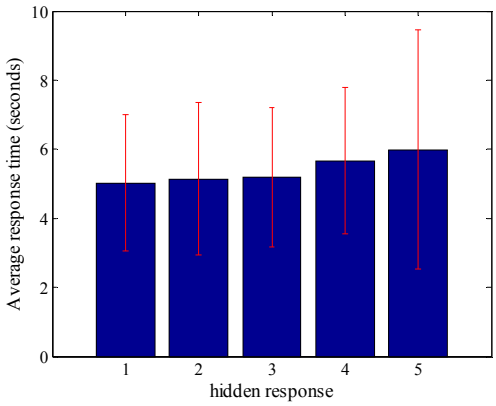

Figure 10: Average response times with respect to different hidden responses.

4.3 Intersection Attack

Intersection attack is not new and has been reported in previous research on other human authentication systems especially graphical passwords [16]. The basic idea behind intersection attack is to fuse the information obtained in multiple observed login sessions to reduce the space of password space (i.e., the password entropy). This subsection presents intersection attacks on the original and alternative designs of Undercover in [43,21].

4.3.1 Breaking the original Undercover design with randomized public challenges

In [43], the system is designed so that each pass-picture and decoy picture is shown once and only once in a single authentication process. Unfortunately, showing each picture only once is not a sufficient condition to maintain the security. In fact, how the public challenges are generated also matters. In this sub-subsection, we show that the password can be exposed with $O(10)$ observed login sessions if randomized public challenges are used.

In [43] it was not made clear how public challenges should be generated. Our communications with the authors of [43] revealed

that they implemented their prototype system with fixed public challenges, so their prototype does not suffer from the security problem discussed in this sub-subsection. However, since this issue was not discussed in [43], a reader might assume that randomizing public challenges is still fine or even beneficial because randomness often helps enhance the security of a system. Therefore, the intersection attack in this sub-subsection shows how important such small design details are for a secure system.

Each public challenge exposes a significant amount of information about the password due to the following fact: each public challenge (i.e., a set of four pictures) contains at most one (i.e., either none or one) pass-picture. This means that a candidate password can be excluded if two or more pass-pictures in this candidate password appear in a public challenge. In other words, observation of one public challenge can lead to a reduction of the password space. Therefore, as the number of observed public challenges increases, the password space will become smaller and smaller and finally the real password will be revealed after a number of login sessions are observed. For n given observed public challenges, the reduced password space can be mathematically calculated as the intersection of the reduced password spaces corresponding to the n public challenges; hence we call this attack an "intersection attack". The real attack is performed in a simpler way:

- *Step 1*: Set **P** to be the space of all possible passwords.
- *Step 2*: For each observed public challenge, reduce the space of candidate passwords **P** by checking each password in **P** and removing invalid ones.
- *Step 3*: Repeat *Step 2* until all observed challenges are processed or the size of **P** becomes 1.

The above attack can be theoretically analyzed to get how quickly the password space is reduced and how many observed login sessions may be needed to get the password with high probability. See Section 10.2 of the Appendix for such a theoretical analysis.

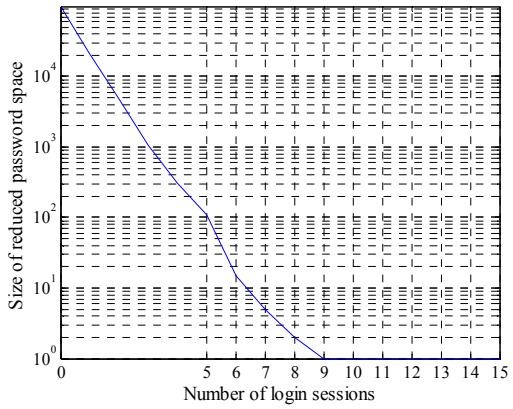

Figure 11: The size of reduced password space in an intersection attack on the original Undercover design.

To verify the real performance of the above intersection attack, we did MATLAB simulations on the original Undercover system with 15 randomly generated login sessions (i.e., 105 public challenges). The experimental results showed that the actual number of observed login sessions for uniquely revealing the password is seven to ten in most cases. A typical simulation result is shown in Figure 11. We performed the intersection attack on real login data collected in our user studies, and the passwords of all 28 users were successfully broken. The number of required login sessions ranges from eight to eleven, and the median number is nine.

4.3.2 Breaking alternative Undercover designs

For the two alternative designs proposed in [21], the same intersection attack still works but in a slightly different way. Now no public challenge is available, but the user's response becomes the source of information leakage. This is due to a flaw in the alternative designs: for different PIN digits and hidden digits, the user needs to press different sequences of arrow buttons to make a correct response. As a result, the buttons presses and their order can leak information of the PIN and hidden digits.

This problem can be best explained by an example. Assume the PIN digit is 2 and the hidden digit is 6. To make a correct response, the user needs to press Button "Left" (◄) and Button "Down" (▼) (the order does not matter). Obviously, pressing Button "Down" leaks the information that the PIN digit is in the first row. Similarly, pressing Button "Left" reveals that the PIN digit must not be 0. As a whole, the number of possible PIN digits is reduced from ten to only four (1, 2, 3 or 4).

Button press pattern	*Possible PIN digits*	*Possible hidden digits*
▼	0, 1, 2, 3, 4	5, 6, 7, 8, 9
▲	5, 6, 7, 8, 9	0, 1, 2, 3, 4
◄	1, 2, 3, 4, 6, 7, 8, 9	0, 1, 2, 3, 5, 6, 7, 8
►	0, 1, 2, 3, 5, 6, 7, 8	1, 2, 3, 4, 6, 7, 8, 9
◄◄	2, 3, 4, 7, 8, 9	0, 1, 2, 5, 6, 7
►►	0, 1, 2, 5, 6, 7	2, 3, 4, 7, 8, 9
◄◄◄	3, 4, 8, 9	0, 1, 5, 6
►►►	0, 1, 5, 6	3, 4, 8, 9
◄◄◄◄	4, 9	0, 5
►►►►	0, 5	4, 9

Table 1: Information leaked from different button presses.

PIN digit	*Combinations of button press patterns*	*Occurrence probability in n responses*
0	▼ + ►►►►	$(1-0.5^n)(1-0.8^n)$
4	▼ + ◄◄◄◄	
5	▲ + ►►►►	
9	▲ + ◄◄◄◄	
1	▼ + ►►► + ◄*	$(1-0.5^n)(1+0.6^n-2\times0.8^n)$
3	▼ + ► + ◄◄◄*	
6	▲ + ►►► + ◄*	
8	▲ + ► + ◄◄◄*	
2	▼ + ►► + ◄◄*	
7	▲ + ►► + ◄◄*	
* For the combinations revealing PIN digits 1, 2, 3, 6, 7 and 8, the second and the third button press patterns should appear in two different responses to a challenge at the same position of two different login sessions; otherwise they will completely or partly cancel each other. The first button press pattern can appear in the same response as the other two.[1]		

Table 2: Combinations of button presses that are sufficient to uniquely reveal the ten PIN digits.

Table 1 shows a list of different button press patterns that can leak information about PIN digits, where the occurrence probability of

[1] For instance, we may have two responses to the second challenge of two different login sessions: ▼►►► and ▼◄, which reveal that the second PIN digit is 1.

each case assuming that each PIN digit and each hidden digit distribute uniformly in {0,…,9}. Here, we ignore button presses that cancel each other, e.g. one "Left" followed by one "Right" or one "Up" followed by one "Down". From Table 1, we can see that a combination of some button press patterns can lead to a unique determination of the PIN digit. Such combinations of button press patterns are enumerated in Table 2, with their occurrence probability in n responses to n random challenges (see Section 10.3 of the Appendix for the calculation of the occurrence probabilities). Note that there is only one response corresponding to each PIN digit in a single login session.

Based on the occurrence probabilities in Table 2, we can estimate how many login sessions are needed to uniquely recover a PIN digit with probability q:

- When the PIN digit is 0, 4, 5, or 9: $(1-0.5^n)(1-0.8^n) \geq q$. This inequality can be solved numerically to get $n \geq n_1(q)$.
- When the PIN digit is 1, 2, 3, 6, 7, or 8: $(1-0.5^n)(1+0.6^n-2\times 0.8^n) \geq q$. This inequality can be solved numerically to get $n \geq n_2(q)$.

When each PIN digit is uniquely determined with probability q, the whole 4-digit PIN is uniquely determined with probability q^4. To make $q^4 \geq 0.5$, we need to have $q \geq 0.8409$. When $q=0.8409$, we can calculate $n_1(q)=9$ and $n_2(q)=12$. This means that, given twelve login sessions, the 4-digit PIN can always be uniquely determined with probability no less than 0.5. If the PIN is composed of 0, 4, 5 and 9 only, nine observed login sessions will be enough.

We did a large number of MATLAB simulations to test the real performance of the intersection attack. For a PIN "1236", 1000 random attacks showed that the median number of login sessions needed to uniquely reveal the whole PIN is eleven. For a PIN "0459", the median number is nine. One can see that the attack works very well and our theoretical analysis is very accurate. Since we did not implement this alternative design of Undercover, the attack was not validated by real login data from a user study. But the attack does not depend on human behavior at all, so a re-validation via a user study is not really necessary.

5. ENHANCING UNDERCOVER

As we described above, the two proposed attacks are based on some flaws in the original and alternative designs of Undercover. By removing these design flaws, we can enhance the security of Undercover. To simplify our discussion, here we only focus on how to enhance the original Undercover design.

To resist the intersection attack, we should avoid information leaked from public challenges. This means that we need to use a fixed set of seven public challenges in all login sessions (as the authors of [43] implemented their Undercover prototype). If the order of the seven public challenges in each login session and the order of the four pictures in each public challenge should also be fixed is an issue for future study. It remains a question if fixing either order or both can lead to other new attacks.

To resist the timing attack, we need to make the five button layouts equally difficult for human users to handle. Randomly shuffling them is a simple way to achieve this goal. The shuffling can be done dynamically for each challenge to minimize any potential nonuniformity of human users' response time to different hidden challenges.

We developed an enhanced edition of our Undercover implementation by adopting the above two measures. A two-week user study with 22 participants was performed to verify its performance against the timing attack, i.e., if human users can now respond to hidden challenges more uniformly. The same protocol as the user studies on the original Undercover scheme was followed for this new user study. All participants also attended the previous user study, except one new user who was recruited for testing this new enhanced design. The average response times become flatter as shown in Figure 12. However, paired t-tests showed that the average response time to "Up" hidden challenges is still significantly shorter than the average response time to hidden challenges "Left", "Right" and "Center" (although with much smaller p-values).

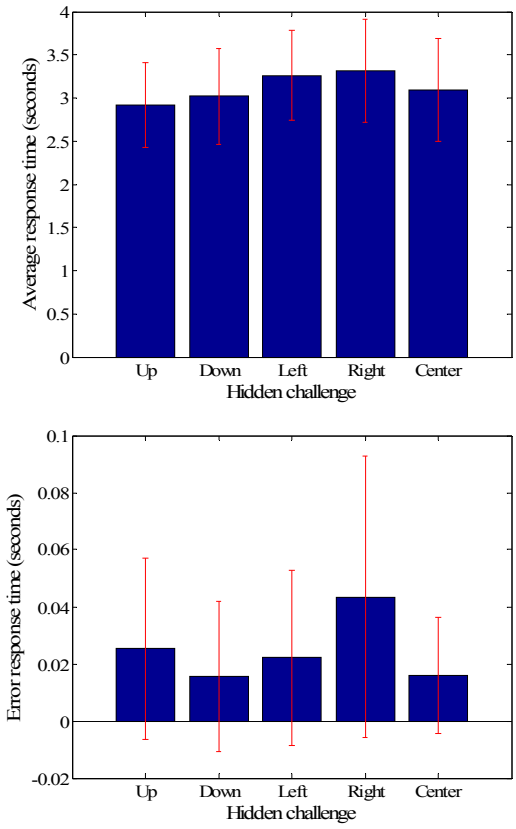

Figure 12: Average response times and error response rates to different hidden challenges of the Undercover implementation enhanced by shuffling button layouts.

After discussing with some participants, we noticed a possible explanation to the still shorter response time to "Up" hidden challenges. Observing Figure 1, we can see that the button layout corresponding to "Up" hidden challenges is the closest to the public challenge. Some participants recalled that they had spent more time in locating other button layouts and verifying them.

To further remove the new kind of nonuniformity in human behavior, we realized that it is important to re-arrange the user interface so that the distance between each button layout and the public challenge is equal. Further analysis showed that we should also equalize the distance between the pass-picture and the button layout used by the user to make the public response. This led to a new design of the interface of the Undercover as shown in Figure 13. Now we distribute the four pictures in each public challenge uniformly on a circle, and the "no pass-picture" icon at the center of the circle. The five button layouts are located in the same way as the five pictures. To further simplify the user interface, we also

changed the hidden responses to "1", "2", "3", "4" and "5" and the user is asked to: 1) find the hidden response in the button layout near to the pass-picture or the "no pass-picture" icon; 2) press the button at the same location as the hidden response to make the public response. The above changes make the interface tighter and the user's task simpler, so we expect the usability of the system can also be improved.

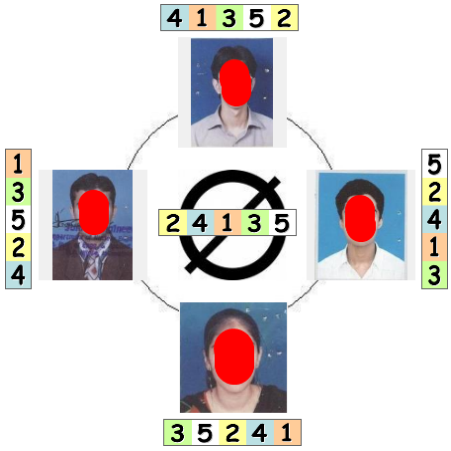

Figure 13: The new layout of our enhanced Undercover implementation.

In the design process of the new enhanced Undercover implementation, we noticed a new kind of nonuniformity that may lead to a new timing attack: if all the 5 pass-pictures have appeared in the first five or six public challenges, the user knows that all the remaining (one or two) public challenge(s) will contain no pass-picture so that he/she may be able to respond faster than the usual case. To avoid this problem, we changed the design so that the last public challenge always contains one pass-picture. This measure has a side effect on the success rate of random guess, which is increased from $1/\left(C_7^2 \cdot 4^5\right) = 1/21504$ to $1/\left(C_6^2 \cdot 4^5\right) = 1/15360$, around 1.4 times larger. Since 1/15360 is still smaller than 10^{-4}, the side effect is acceptable.

A one-week user study with 19 users was then performed to check if this new enhancement works. All 19 users are old users who had participated in previous user studies. Unlike our previous user studies, each user was asked to login five to ten times per day so that we can collect enough data for analysis. Figure 14 shows the results obtained from real login data. Now the paired *t*-test fails to reject the null hypothesis that the response time to Hidden Challenge "1" has the same mean as the response time to other hidden challenges, thus leading us to believe that the response times to different hidden challenges are not significantly different. Simulated attacks on the enhanced Undercover implementation showed that none of the user passwords was broken. The success rate of breaking pass-pictures is always below 50%. In addition, as we expected, the average login times and the login error rates are both improved compared to the original Undercover design: the average login time is reduced to less than 19 seconds after 20 logins and the error rate over all 19 users is just around 6%. We believe that the average login time can be reduced to within 10 seconds after the user becomes more familiar with the system and her password. While this is still significantly longer than the average time of entering a 4-digit PIN, it is likely that we have to pay some additional costs for getting the additional security against passive observers. It remains a question if an even better design can be made to further reduce the average login time.

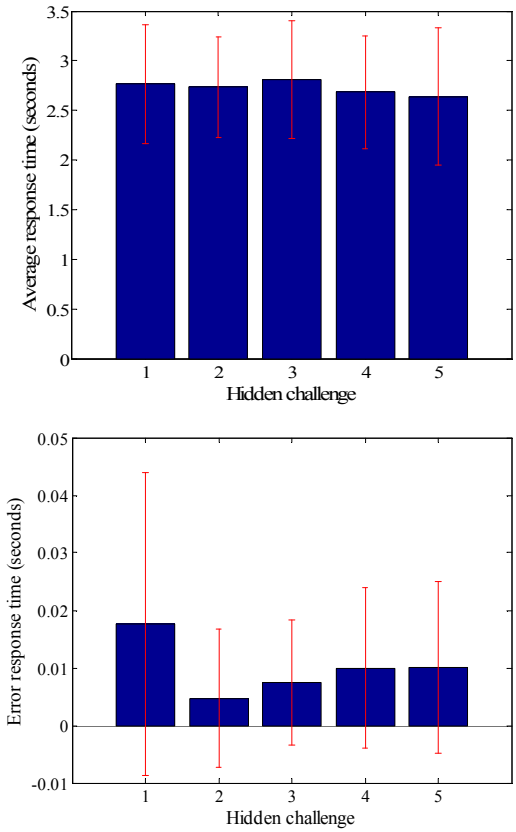

Figure 14: Average response times and error response rates to different hidden challenges, of the Undercover implementation enhanced with the new interface in Figure 13.

6. Generalizing Timing Attack

The idea of timing attack may also be generalized to break other human authentication systems based on hidden challenges. For instance, the uni-modal designs proposed in [8,9] ask the user to rotate an input device to match a target (which is a secret password item) cued via a tactile or audio channel. It is obvious that the response time depends on how far the current cue is from the target. In addition, for different targets, the average response time should be different because the average distance from a random cue to the target is different. For instance, assuming that the list of possible cues/targets are 0,…,9 and they follow a predetermined fixed order, then the targets 4 and 5 have the minimum average response time. By further considering the direction of the overall rotation, one can further distinguish 4 and 5. In the same manner one can distinguish all targets. If such a timing attack works in practice will be part of our future research.

Although the timing attack is proposed to break Undercover-like human authentication schemes as ad hoc designs, the reason why it works for Undercover has its root in the way how a normal human user responds to visual challenges that require mental efforts: she needs to first look for visual patterns of her interest, interpret it properly, then compute the correct response and finally makes the response by moving her body and/or finger(s). If any of the four steps has a dependency on the contents of the challenge, the user may respond differently to different challenges. Such a behavioral difference may lead to an effective timing attack. If

there is more than one kind of such human behavior, different combinations of them may lead to different timing attacks. In case the password cannot be completely broken, the information leaked may be useful to reduce the password space thus making a brute force attack feasible. Considering the fact that human behavior can be nonuniform and highly nonlinear in many aspects, the exploitation space of attack based on human behavior may be much larger than what we think of. Note that timing attack may not be the only form of human behavior based attacks. In the following, we briefly discuss different aspects that may lead to human behavior attacks on human authentication systems.

Response time. This has been shown clearly by the timing attack on Undercover designs proposed in this paper and previous research on some other systems like PIN input devices and VibraPass [13,14,29]. There are different sources of nonuniformity that may be exploited by an attacker to launch a successful timing attack. For instance, for graphical password systems based on an image pool [16,22,23,25,26,34,40,44,47], the user's response to a challenge may be faster if the (average) distance of the pass-picture(s) to the left upper corner of the displayed challenge and/or the pass-picture(s) are more visually attractive or eye-catching (due to their colors or patterns or semantics). In addition, depending on the personal nature of a given user, she may be more sensitive to specific challenges and response slower or faster than average. As a typical example, color blind users will respond slower to color patterns that fall into their color vision deficiency, so a careless design of the challenges may lead to an additional risk that does not exist for users with normal color vision. Considering the fact that a considerable percentage of the whole population are suffering from color blindness (e.g., 8% Caucasian males and 0.5% Caucasian females [18]), this effect may not be negligible for graphical password systems. Recall that Undercover was also designed to work with distorted images which will likely make the distinguishability of pass-pictures from decoy pictures more dependent on color differences and thus may lead to a higher risk of a timing attack on color blind users.

Response error rate. Similar to nonuniform response time, the nonuniformity of response errors may also be used to develop a similar human behavior based attack. For instance, some graphical password systems (e.g., those reported in [23,34] require the user to count the number of pass-pictures in each challenge, which implies that for some (if not all) users the response error rate may increase as the number of pass-pictures. By observing if a user failed a login session and how many times she re-tried, an attacker may get some useful information to reduce the password space. Note that once having made a mistake, the user may be more careful and be slower in the second login attempt and spend more time on confusing challenges, therefore, an attacker may further get more useful information about the challenge(s) for which the user made wrong responses. In case the attacker is allowed to impersonate the server, he can present carefully constructed challenges to induce login failures.

Mental computation. One of the reasons why our proposed timing attack works for Undercover is that human users need different amounts of mental efforts to handle different hidden challenges. Since all human authentication systems require the user to do some mental computation (recalling, counting, recognizing, comparing, calculating, etc.), there is always a potential risk that some kind of nonuniformity exists so that an effective timing attack can be developed based on it.

Temporal variation. The response time and the response error rate of a human user may vary and evolve during the course of using the system. It is also possible that a user becomes "smarter" after using a password system for a long time so that she creates some shortcuts to make faster responses to some challenges with a higher probability. This may create new attacks or improving the performance of existing attacks. For instance, the unsuccessful timing attack described in Section 4.2.3 may start working after the user becomes very familiar with the system and her pass-pictures and the seven fixed public challenges if she can locate pass-pictures faster and make quicker responses than before.

Personal preference. Previous research [11] has shown the important role of personal preference in the security of graphical password systems. It is likely that some users may suffer from a higher risk of timing attack if they select weak passwords linked to their personal preference. One consequence is a possible change of the response pattern. For the original Undercover design, the time gap between the response times to public challenges with and without pass-pictures may become larger or some new time gaps may appear. Note that it may not be easy (if not impossible) to completely avoid personal preference for graphical passwords since the users do need some semantic clues to help them remember their passwords. As a consequence, in principle there is always an exploitable personal preference that can potentially be used by an attacker. In addition, it deserves noting that the cultural and religious backgrounds normally play an important role in a user's preference. In our user studies, although no exploitable difference was observed between the Croatian and Pakistani groups for the proposed timing attack, it remains a question if a different attack can be developed by exploiting some statistical differences we missed during the user study or such an attack exists for other human authentication systems.

Facial expression and hand/body movement. For human authentication systems against passive observers, the attacker can install a hidden video camera to record the login sessions. He may also install a secondary hidden camera to record the facial expression and hand/body movement of the user during the login sessions. This point is also discussed in [43], where Sasamoto et al. observed that some users moved their hands improperly to leak information about the hidden challenge or the pass-picture. In addition to hand movement, the user's facial expression may also leak information about the hidden challenge or the pass-picture. For instance, when a public challenge without any pass-picture is shown, the user may look less relaxed than when a public challenge with a pass-picture is presented. Similarly, when all the five pass-pictures have been shown so the user knows the last public challenge will not contain any pass-picture, she may appear very relaxed and move her eyes towards the button layout without looking at the computer screen before making the last response.

The above discussion is very general and can in principle apply to *all* human authentication systems. In our opinion, *every* human authentication system must be carefully evaluated against human behavior attacks by considering all the above points. As a general rule, the user interface should be designed in such a way that most human users will not have distinguishable nonuniform behavior. In some cases, educating users may also help mitigate the risk, but it is desirable to avoid user education since users are well-known for not behaving very well even after being educated. In our future work, we will investigate if similar human behavior based attacks exist in other human authentication systems especially other recognition and recall based graphical password systems.

7. CONCLUSION

This paper reports two practical attacks to Undercover, a human authentication system proposed at CHI'2008 which was believed to be secure against observation attacks. We reveal security weaknesses in Undercover due to some design flaws and insecure human behaviors. We also proposed some enhancements to make Undercover more secure against the proposed attacks. User studies were carried out to verify both our proposed attacks and the performance of the suggested enhancements.

Our work has implications beyond gauging the security of Undercover as an ad hoc design. Our results reemphasize that designers of security systems should pay special attention to the human-computer interfaces of their systems. More specifically, the attacks proposed in this paper demonstrate that, if meticulous care is not exercised in measuring how human users will perceive and operate a security system, user behavior can reveal sensitive information that can be used to break the system. Our work on enhancing Undercover also showed that usable solutions to insecure human behaviors are not always intuitively obvious.

In our future work, we plan to generalize the timing attack to other Undercover-like designs and other human authentication systems. We will also look for new Undercover designs that can lead to a shorter login time and a lower login error rate. One possible direction is to explore uni-modal designs that remove the public challenges since some previous work suggests that they unnecessarily increase the mental work load of users.

8. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and the shepherd of the paper, Alexander De Luca, for their comments on further enhancement of the paper. The authors also thank the participants of our user studies who made this research possible. Shujun Li was supported by a fellowship from the Zukunftskolleg, University of Konstanz, Germany, as part of the "Excellence Initiative" Program of the DFG (German Research Foundation).

9. REFERENCES

1. A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40-46, 1999.
2. F. A. Alsulaiman, J. Cha and A. El Saddik. User identification based on handwritten signatures with haptic information. In *Haptics: Perception, Devices and Scenarios, 6th International Conference, EuroHaptics 2008, Proceedings*, Volume 5024 of *Lecture Notes in Computer Science*, 114-121, Springer, 2008.
3. R. J. Anderson. Why cryptosystems fail. *Communications of the ACM*, 37(11): 32-49, 1994.
4. H. J. Asghar, S. Li, J. Pieprzyk and H. Wang. Cryptanalysis of the Convex Hull Click human identification protocol. In *Information Security, 13th International Conference, ISC 2010, Revised Selected Papers*, Volume 6531 of *Lecture Notes in Computer Science*, 24-30, Springer, 2011.
5. 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, 8th International Conference, ACNS 2010, Proceedings*, Volume 6123 of *Lecture Notes in Computer Science*, 349-366, Springer, 2010.
6. J. Aycock. *Computer Viruses and Malware*. Springer, 2006.
7. X. Bai, W. Gu, S. Chellappan, X. Wang, D. Xuan and B. Ma. PAS: Predicate-based Authentication Services against powerful passive adversaries. In *Proceedings of the 24th Annual Computer Security Applications Conference* (ACSAC'2008), 433-442, IEEE Computer Society, 2008.
8. A. Bianchi, J. K. Lee, I. Oakley and D. S. Kwon. The Haptic wheel: design & evaluation of a tactile password system. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems: Extended Abstracts* (CHI EA'2010), 3525-3530, ACM, 2010.
9. A. Bianchi, I. Oakley, V. Kostakos and D. S. Kwon. The Phone Lock: audio and haptic shoulder-surfing resistant PIN entry methods for mobile devices. In *Proceedings of the 5th International Conference on Tangible, Embedded, and Embodied Interaction* (TEI'2011), 197-200, ACM, 2011.
10. A. Bianchi, I. Oakley and D. S. Kwon. The Secure Haptic Keypad: a tactile password system. In *Proceedings of the 28th ACM International Conference on Human Factors in Computing Systems* (CHI'2010), 1089-1092, ACM, 2010.
11. D. Davis, F. Monrose and M. K. Reiter. On user choice in graphical password schemes. In *Proceedings of the 13th USENIX Security Symposium*, 151-164, USENIX, 2004.
12. A. De Luca and B. Frauendienst. A privacy-respectful input method for public terminals. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges* (NordiCHI'2008), 455-458, ACM, 2008.
13. A. De Luca, M. Langheinrich and H. Hußmann. Towards understanding ATM security: a field study of real world ATM use. In *Proceedings of the 6th Symposium on Usable Privacy and Security* (SOUPS'2010), Article 16, ACM, 2010.
14. A. De Luca, E. von Zezschwitz and H. Hußmann. VibraPass – secure authentication based on shared lies. In *Proceedings of the 27th ACM International Conference on Human Factors in Computing Systems* (CHI'2009), 913-916, ACM, 2009.
15. T. Deyle and V. Roth. Accessible authentication via tactile pin entry. *Computer Graphics Topics*, Issue 3, 2006. http://www.zgdv.de/PDFs/topics/2006/topics3_2006.pdf.
16. R. Dhamija and A. Perrig. Déjà Vu: a user study using images for authentication. In *Proceedings of the 9th Conference on USENIX Security Symposium*, 45-58, USENIX, 2000.
17. K. Dunham (Technical Editor). *Mobile Malware Attacks and Defense*. Syngress Publishing, 2008.
18. L. Fleming Fallon. Color blindness. In *Gale Encyclopedia of Children's Health: Infancy through Adolescence*, Volume 1, 449-452, Gale, 2005.
19. A. Forget, S. Chiasson and R. Biddle. Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In *Proceedings of the 28th ACM International Conference on Human Factors in Computing Systems* (CHI'2010), 1107-1110, ACM, 2010.
20. P. Golle and D. Wagner. Cryptanalysis of a cognitive authentication scheme. In *Proceedings of 2007 IEEE Symposium on Security and Privacy* (S&P'2007), 66-70, IEEE Computer Society, 2007.
21. M. Hasegawa, N. Christin and E. Hayashi. New directions in multisensory authentication. In *Adjunct Proceedings of the*

Seventh International Conference on Pervasive Computing (Pervasive 2009) – Late Breaking Results, 2009.

22. E. Hayashi, N. Christin, R. Dhamija and A. Perrig. Use Your Illusion: secure authentication usable anywhere. In *Proceedings of the 4th Symposium on Usable Privacy and Security* (SOUPS'2008), 35-45, ACM, 2008.
23. N. J. Hopper and M. Blum. Secure human identification protocols. In *Advances in Cryptology – ASIACRYPT 2001*, Volume 2248 of *Lecture Notes in Computer Science*, 52-66, Springer, 2001.
24. M. Jakobsson and S. Myers (Editors). *Phishing and Countermeasures*. John Wiley & Sons, 2007.
25. H. Jameel, R. Shaikh, H. Lee and S. Lee. Human Identification through image evaluation using secret predicates. In *Topics in Cryptology – CT-RSA 2007*, Volume 4377 of *Lecture Notes in Computer Science*, 67-84, 2007.
26. H. Jameel, R. Shaikh, L. Hung, Y. Wei, S, Raazi, N. Canh, S. Lee, H. Lee, Y. Son and M. Fernandes. Image-feature based human identification protocols on limited display devices. In *Information Security Applications, 9th International Workshop, WISA 2008, Revised Selected Papers*, Volume 5379 of *Lecture Notes in Computer Science*, 211-224, Springer, 2009.
27. R. Kuber and W. Yu. Authentication using tactile feedback. In *Proceedings of the 20th British HCI Group Annual Conference on People and Computers* (HCI'2006), Volume 2, 141-145, British Computer Society, 2006.
28. R. Kuber and W. Yu. Feasibility study of tactile-based authentication. *International Journal of Human Computer Studies*, 68(3):158-181, 2010.
29. D. F. Kune and Y. Kim. Timing attacks on PIN input devices. In *Proceedings of the 17th ACM Conference on Computer and Communications Security* (CCS'2010), 678-680, ACM, 2010.
30. M. Lei, Y. Xiao, S. V. Vrbsky, C.-C. Li and L. Liu. A virtual password scheme to protect passwords. In *Proceedings of 2008 IEEE International Conference on Communications* (ICC'2008), 1536-1540, IEEE, 2008.
31. S. Li, H. J. Asghar, J. Pieprzyk, A.-R. Sadeghi, R. Schmitz and H. Wang. On the security of PAS (Predicate-based Authentication Service). In *Proceedings of the 25th Annual Computer Security Applications Conference* (ACSAC'2009), 209-218, IEEE Computer Society, 2009.
32. S. Li, S. A. Khayam, A.-R. Sadeghi and R. Schmitz. Breaking randomized linear generation functions based virtual password system. In *Proceedings of 2010 IEEE International Conference on Communications* (ICC'2010), IEEE, 2010.
33. S. Li and H.-Y. Shum. Secure Human-Computer Identification against peeping attacks (SecHCI): A survey. Technical report, 2003. http://www.hooklee.com/Papers/SecHCI-Survey.pdf.
34. S. Li and H.-Y. Shum. Secure Human-Computer Identification (Interface) systems against peeping attacks: SecHCI. IACR's Cryptology ePrint Archive: Report 2005/268, 2005.
35. X.-Y. Li and S.-H. Teng. Practical human-machine identification over insecure channels. *Journal of Combinatorial Optimization*, 3(4):347-361, 1998.
36. B. Malek, M. Orozco and A. El Saddik. Novel shoulder-surfing resistant haptic-based graphical password, In *Proceedings of EuroHaptics'2006*, 179-184, EuroHaptics Society, 2006.
37. T. Matsumoto. Human-computer cryptography: an attempt. In *Proc. 3rd ACM Conference on Computer and Communications Security* (CCS'96), 68-75, ACM, 1996.
38. T. Matsumoto and H. Imai. Human identification through insecure channel. In *Advances in Cryptology – EUROCRYPT'91*, Volume 547 of *Lecture Notes in Computer Science*, 409-421, Springer, 1991.
39. A. J. Menezes, P. C. van Oorschot and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
40. Passfaces Corporation. Passfaces: Two Factor Authentication for the Enterprise. http://www.passfaces.com, last visited on 6th June 2011.
41. T. Perković, M. Čagalj and N. Saxena. Shoulder-surfing safe login in a partially observable attacker model. In *Financial Cryptography and Data Security: 14th International Conference, FC 2010, Revised Selected Papers*, Volume 6052 of *Lecture Notes in Computer Science*, 351-358, Springer, 2010.
42. V. Roth, K. Richter and R. Freidinger. A PIN-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM Conference on Computer and Communications Security* (CCS'2004), 236-245, ACM, 2004.
43. H. Sasamoto, N. Christin and E. Hayashi. Undercover: authentication usable in front of prying eyes. In *Proceeding of the 26th ACM International Conference on Human Factors in Computing Systems* (CHI'2008), 183-192, ACM, 2008.
44. L. Sobrado and J. C. Birget. Graphical passwords. *The Rutgers Scholar*, vol. 4, 2002.
45. F. Tari, A. A. Ozok, and S. H. Holden. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of the 2nd Symposium on Usable Privacy and Security* (SOUPS'2006), 56-66, ACM, 2006.
46. C.-H. Wang, T. Hwang and J.-J. Tsai. On the Matsumoto and Imai's human identification scheme. In *Advances in Cryptology – EUROCRYPT'95*, Volume 921 of *Lecture Notes in Computer Science*, 382-392, Springer, 1995.
47. D. Weinshall. Cognitive authentication schemes safe against spyware. In *Proceedings of 2006 IEEE Symposium on Security and Privacy* (S&P'2006), 295-300, IEEE Computer Society, 2006.
48. S. Wiedenbeck, J. Waters, L. Sobrado and J.-C. Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of International Working Conference on Advanced Visual Interfaces* (AVI'2006), 177-184, ACM, 2006.
49. H. Zhao and X. Li. S3PAS: a scalable shoulder-surfing resistant textual-graphical password authentication scheme. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops* (AINAW'2007), 467-472, IEEE Computer Society, 2007.

10. APPENDIX

10.1 Theoretical analysis of the timing attack

In this subsection of the appendix, we give a theoretical analysis of the performance of the timing attack. The success of the timing attack depends on if a pass-picture has a higher probability of being distinguished as a pass-picture in *Step 2* than a decoy picture. If so, a pass-picture will have a larger counter value than a decoy picture. Thus, pass-pictures will likely be ranked higher than decoy pictures in *Step 3*. Intuitively, increasing the number of login sessions will increase the probability that all pass-pictures are ranked as the top five pictures in *Step 3*, thus increase the success rate of the timing attack.

In each login session, denote the probability that a pass-picture's counter is increased by p_1, the probability that a decoy picture's counter is increased by p_2, and the probability that no counter is increased by p_3. Based on the original Undercover design, $5p_1+23p_2+p_3=1$ should hold. There are eight events we need to consider for calculating the three probabilities:

- *Event 1*: The login session includes at least one "Up" hidden challenge, which happens with a probability $p_{E1}=1-(1-1/5)^7\approx 0.7903$.
- *Event 2*: The fastest response corresponds to an "Up" hidden challenge, which happens with a user-dependent probability p_{E2}.
- *Event 3*: The public challenge corresponding to the fastest response includes a pass-picture, which happens with a probability $p_{E3}=5/7\approx 0.7143$.
- *Event 4*: Given that we are observing a public challenge with a pass-picture, the probability that a specific pass-picture appears in the challenge is $p_{E4}=1/5=0.2$.
- *Event 5a*: The user makes a correct response to an "Up" hidden challenge, which happens with a user-dependent probability p_{E5a}.
- *Event 5b*: The user makes a correct response to an non-"Up" hidden challenge, which happens with a user-dependent probability p_{E5b}.
- *Event 6*: An incorrect response made by the user to a non-"Up" hidden challenge matches the pass-picture if we consider the hidden challenge as "Up", which happens with a user-dependent probability p_{E6}.
- *Event 7*: An incorrect response made by the user to an "Up" hidden matches a public challenge without a pass-picture, which happens with a user-dependent probability p_{E7}.

The pass-picture under consideration will be distinguished as a pass-picture under the following two situations:

- *Events 1*, *2*, *3*, *4* and *5a* happen;
- *Event 2* does not happen, *Events 3*, *4* and *6* happen.

Assuming that the events are independent of each other, we have $p_1=p_{E1}p_{E2}p_{E3}p_{E4}p_{E5a}+(1-p_{E2})p_{E3}p_{E4}p_{E6}$. This probability is user dependent because p_{E2} and p_{E5} are both user dependent. To ease our discussion, we use the median probabilities of *Events 2* and *5* obtained in our user studies: $p_{E2}=0.6583$, $p_{E5a}=0.9871$, $p_{E5b}=0.9652$. For *Events 6* and *7*, we assume that the user makes incorrect responses randomly, so $p_{E6}=p_{E7}=1/4=0.25$. With all the above values of those the user-dependent probabilities, $p_1=0.0865$. The probability p_3 is equal to $(1-p_{E3})+p_{E3}(p_{E2}(1-p_{E5a})+(1-p_{E2})(1-p_{E5b}))p_{E7}\approx 0.2894$. From p_1 and p_3, we can derive $p_2=(1-5p_1-p_3)/23\approx 0.0123$. Since p_1 is $p_1/p_2\approx 6.96$ times larger than p_2, we expect that the timing attack should work well in practice.

We estimated the values of p_1 and p_2 of each user from our real login data. The results are shown in Figure 15, from which we can see that the values of p_1, p_2 and p_3 are indeed user dependent and also time varying. The actual value of p_1/p_2 is less than the above theoretical estimate, but still significantly larger than 1 for all users and over the whole course of logins. The inaccuracy of the estimate might be attributed to the inaccuracy of the theoretical model itself and/or some probabilities involved in the model.

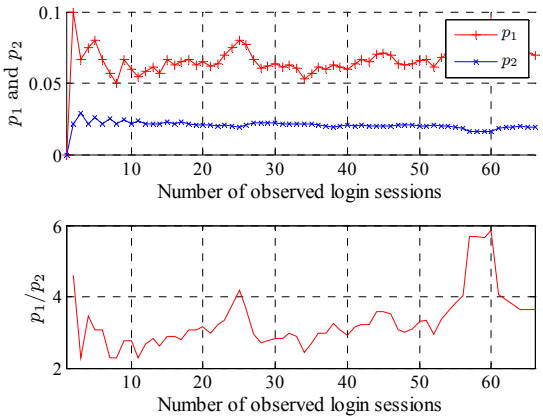

Figure 15: Values of p_1 and p_2 of the median user, estimated from real login data.

Although p_1 is much larger than p_2, the probability that all the five pass-pictures are ranked as top five pictures may not be high when the number of observed login sessions (denoted henceforth as n) is small. In fact, when $n<5$, this probability is 0 because not all pass-pictures can appear. In general, this probability can be re-formulated as follows.

Randomly make n attempts of picking a ball from a box of infinite number of balls labeled with Numbers 1,…,28. *With probability* p_1, *we take a ball with a label between 1 and 5, and with probability* p_2, *we take a ball with a label between 6 and 28. With probability* $p_3=1-5p_1-23p_2$, *we fail to get a ball. At the end, what is the probability that the number of balls with label i is larger than the number of balls with label j for any* $i\in\{1,\ldots,5\}$ and $j\in\{6,\ldots,28\}$*?*

Denote the above probability by p_{t5} and the number of Objects i by C_i, it can be written as a sum as follows:

$$p_{t5}=\sum_{\min_{i=1}^{5}(C_i)>\max_{i=6}^{28}(C_i),\,D=n-\sum_{i=1}^{28}C_i}\frac{n!}{C_1!\cdots C_{28}!D!}p_1^{C_1+\cdots+C_5}p_2^{C_6+\cdots+C_{28}}p_3^{D}.$$

It is not trivial to get an explicit form of p_{t5}, so we used the Monte Carlo method to estimate p_{t5} for a set of values of n, which are shown in Figure 16 (the line marked with "x"). We can see p_{t5} keeps increasing as n increases. Although it is not very high when n is small, the value is not negligible either. For instance, when n=30, $p_{t5}\approx 0.0842$, which means that 8.42% of passwords can be recovered.

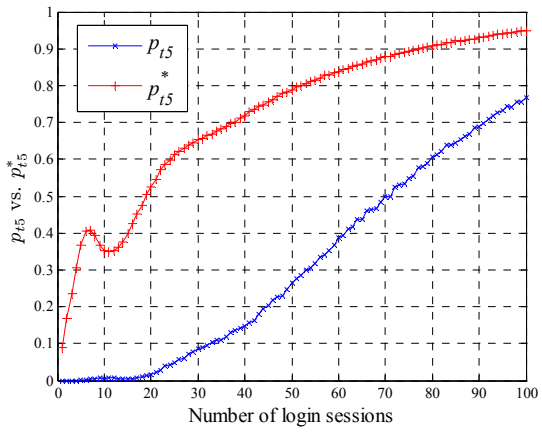

Figure 16: Values of p_{t5} and p_{t5}^* for *n*=1,...,100.

While the success rate of breaking the whole password (i.e., all the five pass-pictures) is not high when n is small, our simulations showed that the probability that a picture in the top five ones is a pass-picture (denoted by p_{t5}^*) is significantly high. The line marked with "+" in Figure 16 shows the results. When n=30, although p_{t5} is only 0.0842, p_{t5}^* is much larger: 0.6508>0.5.

10.2 Theoretical analysis of the intersection attack on the original Undercover design

We can estimate how quickly the size of **P** reduces as n increases. Assume that the reduction rate of the password space $\mathbf{P}_i / \mathbf{P}_{i-1}$ remains stable for all i, where $\mathbf{P}_i$ denotes the reduced password space after i public challenges are checked. To avoid unnecessarily complicating our theoretical analysis, we ignore the fact that there are exactly five public challenges with one pass-picture. This will lead to a slightly larger key space (since less information leakage is counted), but the final result remains fairly accurate as shown in our experiments (see Sec. 4.3.1). Based on the above assumptions, the reduction rate is the following:

$$r = \#(\mathbf{P}_1) / \#(\mathbf{P}_0) = \left(C_4^0 C_{24}^5 + C_4^1 C_{24}^4\right) / C_{28}^5 = 0.865 .$$

The size of $\mathbf{P}_n$ will be $\#(\mathbf{P}_0) r^n = C_{28}^5 r^n$. To uniquely reveal the password, the size of the reduced password space needs to be small enough. Since there has to be at least one element (the true password) in the reduced password space, when $C_{28}^5 r^n < 1.5$ (meaning that the number of wrong passwords in the reduced password space is smaller than 0.5) the probability that the final reduced password has only one element will become high, which leads to $n > \left\lfloor \ln\left(C_{28}^5 / 1.5\right) / \ln\left(1/r\right) \right\rfloor + 1 = 77$. Note that one login session includes seven public challenges, so $\lceil 77/7 \rceil = 11$ observed login sessions will be enough for an attacker to uniquely reveal the password with a considerably high probability.

The computational complexity of the intersection attack is determined by the sum of the sizes of all reduced password spaces: $O\left(\sum_{i=0}^{n-1} \#(\mathbf{P}_i)\right) = O\left(\sum_{i=0}^{n-1} C_{28}^5 r^i\right) = O\left(C_{28}^5 \left(1-r^n\right)/(1-r)\right)$.

Since the value of n will not be much larger than 11, the complexity will be upper bounded by $O\left(C_{28}^5 \left(1-r^{11}\right)/(1-r)\right) \approx O\left(2^{19.2}\right)$.

10.3 Occurrence probabilities of combinations of button press patterns in alternative Undercover designs

The occurrence probability of a specific combination of button press patterns in n responses can be calculated based on the probability of each button press pattern in the combination. In the following, we show how the two values in the last column of Table 2 are derived.

When the PIN digit is 0, the combination includes two button press patterns: the first one is ▼, and the second is ►►►►. To calculate the occurrence probability of the pattern combination, we need to know the occurrence probabilities of the two patterns. They can be derived from the ten possible patterns combinations corresponding to the ten hidden digits: 1) none; 2) ►; 3) ►►; 4) ►►►; 5) ►►►►; 6) ▼; 7) ▼►; 8) ▼►►; 9) ▼►►►; 10) ▼►►►►. Assuming the hidden digit distributes uniformly over {0,…,9}, each of the above ten pattern combinations appears in one response with probability 0.1. This means that the occurrence probability of ▼ and ►►►► will be 0.5 and 0.2, respectively. Given n responses, the probabilities that ▼ and ►►►► appear at least once are $1-(1-0.5)^n = 1-0.5^n$ and $1-(1-0.2)^n = 1-0.8^n$, respectively. Further assuming that the two patterns can appear independently in the n responses, the occurrence probability of the pattern combination ▼+►►►► becomes $\left(1-0.5^n\right)\left(1-0.8^n\right)$.

When the PIN digit is 4, 5 or 9, following a similar process to the above one, we can derive that the occurrence probability of the pattern combination of interest is also $\left(1-0.5^n\right)\left(1-0.8^n\right)$.

When the PIN digit is 2, the ten pattern combinations corresponding to the ten hidden digits are: 1) none; 2) ►; 3) ►►; 4) ◄; 5) ◄; 6) ▼; 7) ▼►; 8) ▼►►; 9) ▼◄; 10) ▼◄◄. Thus, the occurrence probabilities of the patterns ▼, ►► and ◄◄ are 0.5, 0.2 and 0.2, respectively. Given n responses, ▼ appears at least once with probability $1-(1-0.5)^n = 1-0.5^n$. The probability that both ►► and ◄◄ appear at least once is a bit more complicated. Let us consider its complement event: ►► does not appear and ◄◄ does not appear. The probability of this complement event can be calculated as:

$$(1-0.2)^n + (1-0.2)^n - (1-0.2-0.2)^n = 2\times 0.8^n - 0.6^n ,$$

where $(1-0.2-0.2)^n$ is the probability that neither ►► nor ◄◄ appears (which has to be subtracted because it is counted twice in the other two terms of the probability). Now we can immediately derive the probability that both ►► and ◄◄ appear at least once: $1-\left(2\times 0.8^n - 0.6^n\right) = 1+0.6^n - 2\times 0.8^n$. Then, combining the probabilities of ▼, ►► and ◄◄, the final occurrence probability of the pattern combination ▼+►►+◄◄ is $\left(1-0.5^n\right)\left(1+0.6^n - 2\times 0.8^n\right)$.

When the PIN digit is 1, 3, 6, 7 or 8, following a similar process to the above one, we can derive that probability of the pattern combination of interest is also $\left(1-0.5^n\right)\left(1+0.6^n - 2\times 0.8^n\right)$.