5-2013

Designing leakage-resilient password entry on touchscreen mobile devices

Qiang YAN
Singapore Management University, qiang.yan.2008@smu.edu.sg

Jin HAN
Institute for Infocomm Research

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

Jianying ZHOU
Institute for Infocomm Research

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Published in ASIA CCS '13, May 8-10, 2013, Hangzhou. pp. 37-48.
https://doi.org/10.1145/2484313.2484318

Designing Leakage-Resilient Password Entry on Touchscreen Mobile Devices

Qiang Yan†, Jin Han‡, Yingjiu Li†, Jianying Zhou‡, Robert H. Deng†
†School of Information Systems, Singapore Management University, Singapore
‡Cryptography and Security Department, Institute for Infocomm Research, Singapore
{qiang.yan.2008, yjli, robertdeng}@smu.edu.sg, {hanj, jyzhou}@i2r.a-star.edu.sg

ABSTRACT

Touchscreen mobile devices are becoming commodities as the wide adoption of pervasive computing. These devices allow users to access various services at anytime and anywhere. In order to prevent unauthorized access to these services, passwords have been pervasively used in user authentication. However, password-based authentication has intrinsic weakness in password leakage. This threat could be more serious on mobile devices, as mobile devices are widely used in public places.

Most prior research on improving leakage resilience of password entry focuses on desktop computers, where specific restrictions on mobile devices such as small screen size are usually not addressed. Meanwhile, additional features of mobile devices such as touch screen are not utilized, as they are not available in the traditional settings with only physical keyboard and mouse. In this paper, we propose a user authentication scheme named CoverPad for password entry on touchscreen mobile devices. CoverPad improves leakage resilience by safely delivering hidden messages, which break the correlation between the underlying password and the interaction information observable to an adversary. It is also designed to retain most benefits of legacy passwords, which is critical to a scheme intended for practical use. The usability of CoverPad is evaluated with an extended user study which includes additional test conditions related to time pressure, distraction, and mental workload. These test conditions simulate common situations for a password entry scheme used on a daily basis, which have not been evaluated in the prior literature. The results of our user study show the impacts of these test conditions on user performance as well as the practicability of the proposed scheme.

Categories and Subject Descriptors

K.6.5 [**Management of Computing And Information Systems**]: Security and Protection— *Authentication*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces— *Evaluation/methodology, User-centered design, Haptic I/O*

Keywords

User Authentication; Leakage-Resilience; Mobile Devices

1. INTRODUCTION

Mobile devices are becoming essential tools in modern life, which seamlessly connect human beings to the cyberspace. A user can now use his smartphone or tablet to access not only general informative services but also sensitive services such as mobile banking and corporate services. In order to prevent unauthorized access to these services, user authentication is required to verify the identity of a user. Among existing user authentication mechanisms, passwords are still the most pervasive due to their significant advantage in usability over other alternatives such as smartcards and biometrics [28]. However, password-based user authentication has intrinsic weakness in password leakage, which may lead to financial loss or corporate data disclosure. This threat could be more serious in scenarios when mobile devices are involved, as mobile devices are widely used in public places.

Password leakage is a classic problem in password-based authentication. Since password leakage usually happens during authentication when a user inputs his password, we focus on the problem of improving leakage resilience of password entry in this work. Most prior research [19, 27, 38, 39, 5, 26, 33, 13, 24] on this problem focuses on desktop computers, where specific restrictions on mobile devices are usually not addressed. These restrictions mainly include: 1) a mobile device usually has a smaller screen size than a desktop computer; 2) a mobile device needs to be operable in non-stationary environments such as on public transit. On the other hand, mobile devices provide additional features such as touch screen, which may not be available in traditional settings. These new features can be utilized to support advanced security properties that were difficult to achieve before.

In this paper, we propose a *concise yet effective* authentication scheme named CoverPad, which is designed for password entry on touchscreen mobile devices. CoverPad improves *leakage resilience* of password entry while *retaining most benefits* of legacy passwords. Leakage resilience is achieved by utilizing the gesture detection feature of touch screen in forming a *cover* for user inputs. This cover is used to safely deliver hidden messages, which break the correlation between the underlying password and the interaction information observable to an adversary. From the other perspective, our scheme is also designed to retain the benefits provided by legacy passwords. This requirement is critical, as Bonneau et al. [10] conclude that any user authentication is unlikely to gain traction if it does not retain comparable benefits of legacy passwords. Our scheme approaches this requirement by involving only intuitive cognitive operations and requiring no extra devices in the design.

We implement three variants of CoverPad and evaluate them with an extended user study. This study includes additional test conditions related to *time pressure*, *distraction*, and *mental workload*. These test conditions simulate common situations for a daily-used password entry scheme, which have not been evaluated in the prior literature. We design new experiments to examine their influence based on previous work in psychology literature [23, 12, 21]. Experimental results show the influence of these conditions on user performance and the practicability of our proposed scheme.

The contributions of this paper are summarized as follows.

- We propose CoverPad to protect password entry on touch-screen mobile devices. It achieves leakage resilience and retains most benefits of legacy passwords by involving only intuitive cognitive operations and requiring no extra devices.
- We implement three variants of CoverPad to address different user preferences. Our user study shows the practicability of these variants.
- We extend user study methodology to examine the influence of various additional test conditions. Among these conditions, time pressure and mental workload are shown to have significant impacts on user performance. Therefore, it is recommended to include these conditions in the evaluation of user authentication schemes in the future.

2. THREAT MODEL

Passwords are the most pervasive user authentication that allows a human *user* to be authenticated to a (local or remote) computer *server*. *Password leakage* is a threat that a user's password is directly disclosed or indirectly inferred. It usually happens during *password entry*, when a user inputs his password in order to prove his identity. In the case of legacy passwords, a user directly enters his plaintext password so that the password may be captured via various eavesdropping attacks including key logger, hidden camera, and malware. We classify these attacks into two types, *external* or *internal*, according to whether an adversary can access the internal states of a device for password entry, such as device memory.

An external eavesdropping attack is an attack exploiting a leakage channel outside a device. This type of attacks includes *vision*-based eavesdropping such as hidden camera, *haptics*-based eavesdropping such as physical key logger, and *acoustics*-based eavesdropping such as tone analysis. Compared to traditional scenarios involving only desktop computers, an adversary has more opportunities to launch an external eavesdropping attack against mobile devices, as mobile devices are widely used in public places. In a crowded area, an adversary may observe password entry in a close distance without being noticed (see Figure 1).

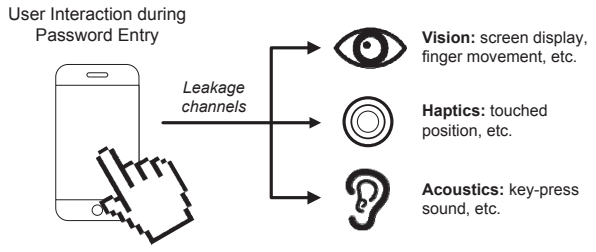


Figure 1: Attack scenarios

For vision-based attacks, an adversary may infer the actual password by observing the movement of fingers even without direct line-of-sight on the screen display. This capability is significantly enhanced with emerging augmented-reality accessory like Google Glass [18], which is a small wearable glass transferring real-time video captured by a tiny camera to a server and displaying the analyzed results received from the server.

Haptics-based attacks are most likely to happen when users use public mobile devices. Mobile devices, such as iPad, have been used as public computer kiosks as observed in museums, restaurants, and hotels [22, 20, 41]. In addition, many existing kiosks are also equipped with touch screen similar to mobile devices. This provides an incentive for an adversary to install a physical "touch" logger. Although such touch logger has not been observed in the wild, it is technically feasible to implement as other physical key loggers [36]. Considering that the thickness of touch screen in Samsung Galaxy S3 is just 1.1mm [2], it may not be noticeable to users if an extra physical touch logger is installed on a normal touch screen.

The effectiveness of acoustics-based attacks depends on whether user actions can be distinguished by their tone patterns. For example, different tones are played when a user dials different numbers on an old-style phone. Due to environmental noises, acoustics-based attacks are usually not as effective as vision-based attacks and haptics-based attacks.

The other type of attacks that cause password leakage is the internal eavesdropping attack. Such attacks exploit a leakage channel inside a device, where an adversary is allowed to access the internal states such as reading device memory. This type of attacks include *logic key logger*, *malware*, and *network eavesdropping*, which are common to all password-based user authentication schemes. Like most prior research [26, 33, 13, 15, 24, 8, 7], our scheme design does not address these attacks for the following reasons: 1) Existing solutions [30, 3, 6, 37] such as application sandbox are available to effectively defend against these attacks, though it takes time for them to replace legacy vulnerable systems; 2) these solutions are independent on user interaction during password entry so that they can be adapted to any user authentication schemes. Compared to external eavesdropping attacks, the threat from internal eavesdropping attacks can be effectively mitigated if a user uses a computer system that is properly updated and configured [16], while it is not easy to defend against external eavesdropping attacks as they are caused by *inevitable* exposure of human interaction during password entry. These external eavesdropping attacks impose realistic threats leading to password leakage. We will thus focus on external eavesdropping attacks in our scheme design.

Besides the above attacks which happen during password entry, password leakage may also be caused by other types of attacks including social engineering and phishing [28]. Although their mitigation technologies such as secure URL checker and spam filter have been widely deployed in modern computer systems, some of these attacks may not be completely preventable by technical solutions alone. Another example is the database reading attack, where the back-end databases are intruded so that all user passwords are compromised. Since these attacks are orthogonal to the password entry problem, they are out of the scope of this paper.

3. COVERPAD DESIGN

In this section, we present the design of CoverPad. First, we describe our design objectives from both security and usability perspectives. Then, we introduce the conceptual design of CoverPad. Lastly, we present three variants in implementing CoverPad.

3.1 Design Objectives

CoverPad is designed to improve leakage resilience of password

entry while retaining most benefits of legacy passwords. We describe our design objectives as follows.

First, in terms of security, a scheme should minimize password leakage during password entry under realistic settings. To achieve this objective, a user should 1) input obfuscated response derived from his password, and/or 2) input his password in a protected environment. A recent study [40] shows strong evidence on the infeasibility of using obfuscated response solely based on human cognitive capabilities. Therefore, it is necessary to rely on certain protected environment to achieve this security objective. However, a fully protected environment may be difficult to establish in practice, which requires to protect all messages delivered between user and server. Therefore, we choose a hybrid solution in our scheme design, where the requirement on a protected environment is significantly reduced with the assistance of simple obfuscation. Such environment is referred to as *partially* protected environment.

In the presence of a partially protected environment, it is possible to achieve the optimal security objective – *no password leakage*. As long as the partially protected environment is not compromised, CoverPad provides the same leakage resilience as *one-time pad* [31], where the most efficient attacks for an adversary to learn the password are online dictionary attacks. We will show how this security objective is achieved in our scheme in the following sections.

Second, in terms of usability, a scheme should preserve the benefits of legacy passwords in order to gain traction [10]. The major benefits of legacy passwords include no extra devices required, and only intuitive cognitive operations performed. We further consider additional restrictions on mobile devices including that 1) a mobile device usually has a *smaller* screen size compared to a desktop computer; 2) a mobile device needs to be operable in a *non-stationary* environment such as on public transit. So we minimize the number of visual elements that are displayed simultaneously on the screen, and also simplify the involved operations to make them suitable in a non-stationary environment.

3.2 Conceptual Design

The conceptual design of CoverPad is shown in Figure 2, where a hidden transformation $T_i(\cdot)$ is a random mapping $\Omega \to \Omega$, where Ω is the set of all individual elements contained in the password alphabet.

Setup:
*A server and a user agree on a k-length password $pwd = (a_1, a_2, \ldots, a_k)$, where a **password element** $a_i = pwd[i]$ belongs to an alphabet with size w. It is allowed that $a_i = a_j$, for $i \neq j$.*

Password Entry:
For each i from $[1, k]$:

Step 1: *The touch screen shows a keypad with all the elements in the alphabet.*

Step 2: *The user is asked to perform a hand-shielding gesture to read the hidden transformation $T_i(\cdot)$ protected by the hand-shielding gesture. $T_i(\cdot)$ will immediately disappear if the gesture is no longer detected.*

Step 3: *The user clicks on response element e_i, where $e_i = T_i(a_i) = (a_i + r_i \mod w)$, where r_i is a random number drawn from a uniform distribution. A new random number r_i is generated for each round i. The hand-shielding gesture is not required for this step.*

Figure 2: Conceptual design of CoverPad

An example of using CoverPad is given as follows. Suppose a user has a k-length password. At the beginning of password entry, the user performs the hand-shielding gesture to view the current hidden transformation T_1 for the first character a_1 in his password. Then, he applies T_1 to a_1 and enters the transformed response e_1. This procedure repeats for each password element a_i. During the whole password entry, T_i disappears immediately once the gesture is not being detected. A user can always view T_i by performing the gesture again before inputting e_i.

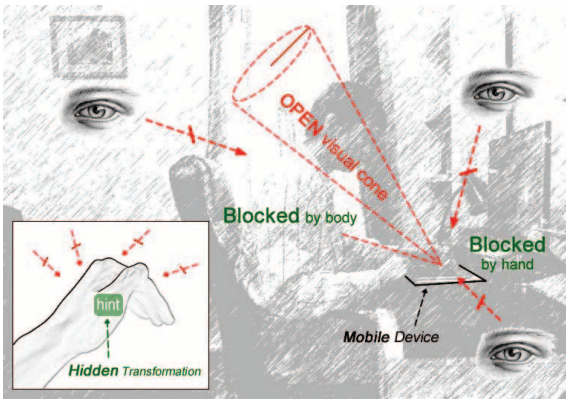


Figure 3: The hand-shielding gesture and its effectiveness

Figure 3 shows how to correctly perform a hand-shielding gesture. This gesture restricts the vision channel to a small visual cone. This visual cone is not accessible to an adversary unless the adversary's eyes are close enough to the user's head, which makes the adversary easily exposed. A hidden camera near the line of sight may help capture the hidden transformation. However, it needs to be adjusted according to the user's height and current position, which may lead to user's awareness. On the other hand, the observable responses for the same password element are uniformly randomized. Thus, CoverPad is also immune to haptics-based eavesdropping. Further analysis is provided in the next section.

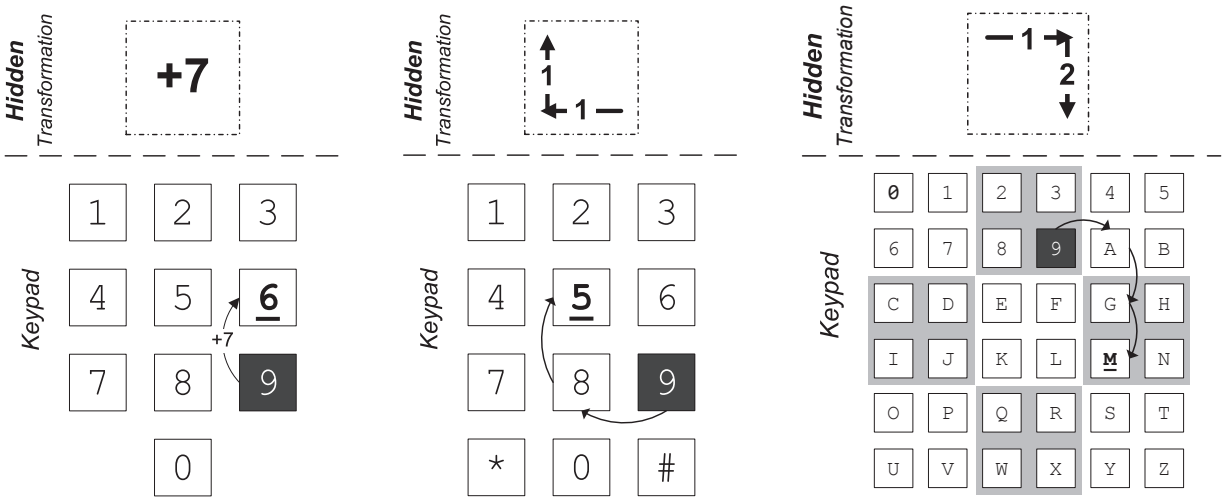
Therefore, it is difficult to compromise the partially protected environment formed by the hand-shielding gesture from external eavesdropping attacks in practice, though the use of this gesture is simple. If the protective gesture is not being detected by the touch screen, the hidden transformation will not be displayed such that the hidden transformation is always protected under the required gesture. Note that a hidden transformation alone does not leak any information about the password. As long as the hidden transformation is not revealed together with the corresponding response, observed interaction provides no valuable information for an adversary to infer the actual password. A proof about this security property will be given in Section 4.

3.3 Implementation Variants

We provide three variants of CoverPad that implement different features tailored for users with various skill sets, which are described and illustrated as follows (see Figure 4).

3.3.1 NumPad-Add

In NumPad-Add, the alphabet of password consists of digits 0 to 9 only. The hidden transformation is performed by *adding* a random digit to the current password element and then $\mod 10$ if the sum is larger than 9, where the value of the random digit ranges from 0 to 9. For example, the correct response for the first round



(a) NumPad-Add (b) NumPad-Shift (c) LetterPad-Shift

Figure 4: Demonstration of three implementation variants

is $6 = (9 + 7) \mod 10$ given password $\underline{9}34567$ and the hidden message '*plus 7*'.

3.3.2 NumPad-Shift

In NumPad-Shift, the alphabet of password consists of digits 0 to 9 only. The hidden transformation is performed by *shifting* the location of the current password element by X-offset and Y-offset, where the offset values are randomly taken from $\{-1, 0, 1\}$ for X-offset, and $\{-1, 0, 1, 2\}$ for Y-offset. For a 3×4 keypad design shown in Figure 4(b), the transformed response for a_i is calculated as $pad[x(a_i) + \Delta x \mod 3][y(a_i) + \Delta y \mod 4]$, where Δx is the X-offset, Δy is the Y-offset, and $x(a_i)$ is the X-index of a_i, and $y(a_i)$ is the Y-index of a_i. For example, the correct response for the first round is 5 if the password is $\underline{9}34567$ and the hidden message is '*move left by 1 step and move up by 1 step*'.

Note that two extra keys $*$ and $\#$ are added to the keypad; otherwise, the distribution of hidden transformations is not uniform on the keypad layout. The proof for the necessity of these two keys is given as follows. Assuming $*$ and $\#$ keys are removed, the keypad now contains only 10 keys for digits 0 to 9. To provide a full transformation from a secret key to a random key, the minimum value set is $\{-1, 0, 1\}$ for X-offsets and $\{-1, 0, 1, 2\}$ for Y-offsets. There are twelve combinations between X-offsets and Y-offsets, but only ten keys on the keypad. If the offset values are drawn from a uniform distribution, certain response keys for a given password element would have a higher frequency compared to others (it is similar as placing twelve balls in ten buckets in a deterministic way). The exact distribution of response keys is decided by the underlying password element, thus it discloses valuable information about the password. From the other perspective, if response keys are drawn from a uniform distribution, the offset values will not be uniformly distributed due to similar reason. Therefore, it is necessary to add these two extra keys to the NumPad-Shift keypad.

3.3.3 LetterPad-Shift

In LetterPad-Shift, the alphabet of password consists of letters a to z and digits 0 to 9 (36 elements in total). The hidden transformation is the same as NumPad-Shift. The offset values are randomly taken from $\{-2, -1, 0, 1, 2, 3\}$ for both X-offset and Y-offset for a 6×6 keypad design. The transformed response for a_i is calculated as $pad[x(a_i)+\Delta x \mod 6][y(a_i)+\Delta y \mod 6]$ in a similar way as for NumPad-Shift. A background grid is added to ease the calculation of shifting, as shown in Figure 4(c).

4. SECURITY ANALYSIS

4.1 External Eavesdropping Attacks

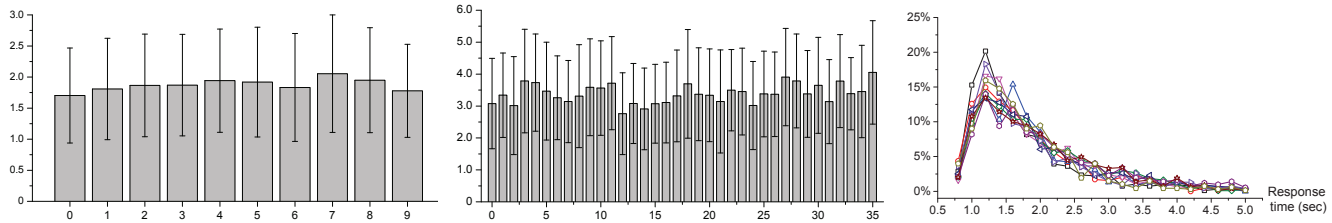
Common external eavesdropping attacks leading to password leakage may exploit vision, haptics, or acoustics channel as analyzed in Section 2. For CoverPad, an adversary using these attacks can observe *at most* a complete response key sequence pressed by a user, while the hidden transformation is protected by our design. From this key sequence, the adversary knows the i-th pressed key is decided by the i-th element in the password. However, the adversary cannot further infer what the i-th password element is, as proved as follows.

Proof: Given a pressed key e_i, and two password elements a_x and a_y in a w-sized password alphabet, let $Pr(e_i|a_x)$ and $Pr(e_i|a_y)$ be the probabilities for e_i being pressed when the underlying password element are a_x and a_y, respectively. We have $Pr(e_i|a_x) = Pr(e_i = a_x + r_i \mod w) = Pr(r_i = e_i - a_x \mod w) = Pr(r_i = C \mod w) = 1/w = Pr(e_i|a_y)$ for any i, x, and y, where C is a constant integer randomly drawn from a uniform distribution. Therefore, a sequence of pressed keys observed by an adversary is equivalent to a random sequence, which is similar to a ciphertext generated by a one-time pad. □

In a partially protected environment where the hidden transformation is protected by the hand-shielding gesture, our scheme achieves no password leakage. As long as the hidden transformation is not disclosed together with the corresponding response, an adversary cannot infer any information about the underlying password (except password length) even after an infinite number of observations.

4.2 Side-channel Attacks

In reality, it is possible for an adversary to exploit subtle side-channels to collect password information during password entry. These attacks are not usually considered in common threat mod-



(a) Timing deviation for NumPad-Add (b) Timing deviation for LetterPad-Shift (c) Timing Distribution for NumPad-Add

Figure 5: Timing deviations and distributions for entering each password element. The results of NumPad-Shift are similar to the results of NumPad-Add shown in these figures.

els [19, 27, 38, 39, 5, 26, 33, 13, 15, 24, 8, 7]. A typical side-channel attack is timing analysis [35], which analyzes the patterns in the response time of entering individual password elements. The preliminary results of our scheme against timing analysis are given in Figure 5. For the timing deviation shown in Figure 5(a) and 5(b), each bar with x-value i represents the average response time for entering the transformed responses for a specific password element i. For the timing distribution shown in Figure 5(c), each line in the figure represents the distribution of the response time for entering the transformed responses for a specific password element. These results show the range and the distribution of the response time for entering different password elements are almost overlapped. This indicates that timing analysis is not a major concern for our scheme, though it is difficult to completely prevent such attacks due to inevitable human behavior patterns during password entry. Detailed analysis on side channel attacks is out of the scope of this paper.

5. USABILITY EVALUATION

5.1 Methodology

The participants in our user study are recruited from undergraduate students in our university. There are 61 participants in total, 30 male and 31 female, with age range between 20 and 25. These participants come from five different departments, in which 42 of them have a social science or business related background, and the remaining have a computer science or information technology related background. Each participant is paid with 10 dollars as compensation for their time. We establish a *ranking* system from which a participant can see a *performance score* representing how well he performs compared to other participants. This ranking system provides a moderate level of motivation for the participants to do their best in tests. A numerical identifier is assigned to each participant in order to protect user privacy.

The user study is conducted in a quiet room. The experiments use a within-objects design. Each participant is asked to use all three variants as three *test groups*. These variants are implemented on Apple iPad, which are referred to as *schemes* in this section. The order of the schemes is randomized to avoid the learning effect that affects the performance for a specific scheme. For each test group, a user is required to memorize a *randomly generated* password in the beginning. The password strength is set to be equivalent to 6-digit PIN, where the password length is four for LetterPad-Shift, and six for both NumPad-Shift and NumPad-Add. The same password will be used for the same test group and a "*show my password*" button is provided in case a participant forgot his password. The participants learn how to use a scheme by an interactive step-by-step tutorial. The participants are required to go through the whole tutorial for the first scheme appearing in the tests, and they may skip the tutorial for the second and third schemes after learning the basic scheme design. In the end of each tutorial, there is a short pretest for the participant to exercise. If a participant fails to pass the pretest, the researchers will provide help to him to ensure that he understands how to use the scheme before the tests start.

In each test group, there are six tests simulating additional *test conditions* that evaluate the influence of time pressure, distraction, and mental workload. The details of these test conditions are described in the next subsection. The order of these tests is also randomized in order to avoid the learning effect.

All three test groups consist of 18 tests in total. To avoid the participants from feeling exhausted and bored, each test is designed to be short and can be finished within one or two minutes. The participants are given a short break after each test group. At the end of the user study, the participants are given a questionnaire using 5-point Likert scale to collect their perception on the schemes. The whole user study takes $35 \sim 50$ minutes to complete.

5.2 Simulating Various Test Conditions

In order to simulate various test conditions related to time pressure, distraction, and mental workload, we introduce two extra experimental tools, timer and secondary task. A *timer* is used to create time pressure by showing a participant how much time is left for the current test condition. It is implemented as a progress bar whose length increases every second with a countdown text field showing how many seconds are left. *Secondary tasks* are used to simulate unexpected distraction and persistent mental workload. We use CRT (*choice reaction time*) tasks as secondary tasks, which is a standard technology in experimental psychology [23, 12, 21]. CRT tasks usually work as secondary tasks that occupy the central executive[1] in human brain when evaluating the performance of a primary task in the presence of a secondary task. CRT tasks require participants to give distinct responses for each possible stimulus. In our implementation, the participants are asked to press the correct button among N buttons, where the correct button should have the same color as the stimulus. For example, if the stimulus shows a red button, a participant should press the red button among N buttons with different colors. We use $N = 2$ for tests in the distraction condition as the major focus is to unexpectedly disrupt password entry with a CRT task. We use $N = 8$ for tests in the mental workload condition so as to create a considerable mental workload, which is the same as in the classic Jensen Box setting [23].

Based on the above experimental tools, we simulate six test conditions for each test group by combining the two modes and three statuses. Two modes related to a timer are described as follows:

- **Relaxed mode:** A participant is asked to minimize the error rate in a fixed number of login attempts where time is not

[1]The central executive is a control system that mediates attention and regulation of processes occurring in working memory [4].

considered in performance score calculation. The number of login attempts is 5 for no-extra-task status and 3 for distraction and mental workload statuses.

- **Timed mode:** A participant is asked to perform as many successful logins as possible within 1 minute, where both time and accuracy are considered in performance score calculation. The countdown of a timer creates time pressure.

Three statuses related to secondary tasks are described as follows:

- **No-extra-task status:** A participant is asked to perform the login task only.
- **Distraction status:** A simple CRT task may appear with $1/3$ probability each time when a participant presses a response key. This task is used to create unexpected distractions during password entry.
- **Mental workload status:** A relatively complex CRT task appears every time when a participant presses a response key. This task is used to create continuing mental workload during password entry.

Among six conditions, we referred to the combination of *relaxed* mode and *no-extra-task* status as the **normal condition**, which is the common condition usually tested in prior work [19, 27, 38, 39, 5, 26, 33, 13, 15, 24, 8, 7]. The short names for the other five conditions are given in Table 1.

Short name	Full specification
normal	*relaxed* mode + *no-extra-task* status
timed	*timed* mode + *no-extra-task* status
distraction	*relaxed* mode + *distraction* status
distraction+timed	*timed* mode + *distraction* status
mental workload	*relaxed* mode + *mental workload* status
mental workload+timed	*timed* mode + *mental workload* status

Table 1: Short names for test conditions

The hypotheses related to these test conditions are described as follows.

(H1) *Compared to the normal condition,* ***login time*** *will be significantly shorter when* ***time pressure*** *is present.*

(H2) *Compared to the normal condition,* ***login accuracy*** *will be significantly lower when* ***time pressure*** *is present.*

(H3) *Compared to the normal condition,* ***login time*** *will be significantly longer when unexpected* ***distraction*** *is present.*

(H4) *Compared to the normal condition,* ***login accuracy*** *will be significantly lower when unexpected* ***distraction*** *is present.*

(H5) *Compared to the normal condition,* ***login time*** *will be significantly longer when persistent* ***mental workload*** *is present.*

(H6) *Compared to the normal condition,* ***login accuracy*** *will be significantly lower when persistent* ***mental workload*** *is present.*

(H7) *Compared to a condition in* ***relaxed*** *mode with* ***secondary tasks****,* ***login time*** *will be significantly shorter for its counterpart in* ***timed*** *mode.*

(H8) *Compared to a condition in* ***relaxed*** *mode with* ***secondary tasks****,* ***login accuracy*** *will be significantly lower for its counterpart in* ***timed*** *mode.*

5.3 Learning Curve

Although our scheme design involves intuitive operations only, it requires a different process for password entry compared to legacy passwords. While we expect the participants can learn this process with the tutorial and pretests, we observed that some participants were impatient to read all instructions and keep pressing the next button. These participants proceeded to the evaluation stage before they fully understand our scheme design.

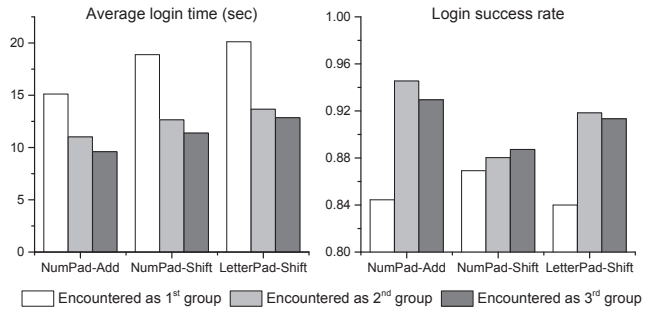


Figure 6: Learning curve of our schemes

Figure 6 compares user performance under the normal condition for different positions where a scheme appears in the study. These results show the user performance in terms of login time and login success rates is significantly worse when the tested scheme is the first scheme which a participant encountered in the user study. But the differences on user performance are not significant if a scheme is encountered as the second or third test group, as all our schemes are similar due to the fact that they are based on the same conceptual design. As shown in the learning curve in Figure 6, most participants get familiar with our scheme design after the first test group. Therefore, we consider the first test group as part of the learning process, and use the performance data collected from the second and third test groups only in the following analysis.

5.4 Experimental Results

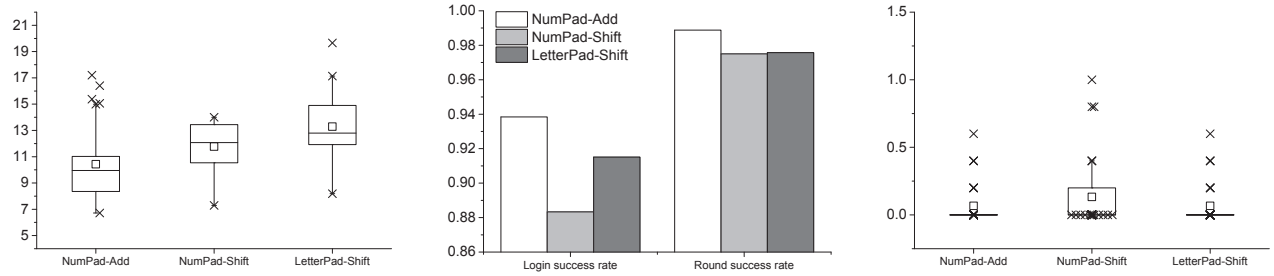
We measure user performance with the following metrics: average login time, login success rates, round success rates, and average edit distances. A *round* success rate is the average success rate for a user to correctly input one password element by applying a hidden transformation. An *edit distance* is the minimum number of insertions, deletions, substitutions, and adjacent transpositions required to transform an input string into the correct password string so that an *average* edit distance is the average value of edit distances calculated from all login attempts of a user under a test condition. Among these metrics, login success rates, round success rates, and average edit distances are used to evaluate *login accuracy*.

We use the following statistical tools to test the significance of our experimental results, where a significance level of $\alpha = .05$ is used. For each comparison, we run an *omnibus* test across all test conditions for each scheme. Since all our performance data are quantitative, we use *Kruskal-Wallis* (KW) test for omnibus tests, which is an analogue of ANOVA but does not require normality. If the omnibus test indicates significance, we further use *Mann-Whitney* (MW) U test to perform pair-wise comparisons so as to identify specific pairs with significant differences. The detailed results of our statistical tests are given in Appendix A.

5.4.1 Performance under Normal Condition

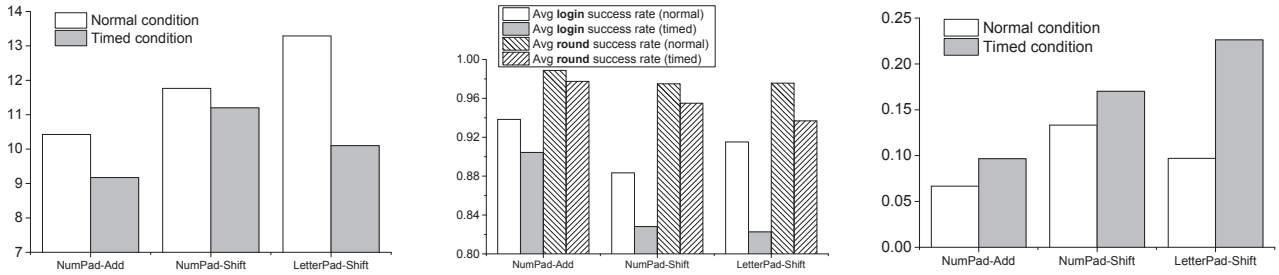
In the normal condition, a participant is only asked to perform login tasks without any time pressure or secondary tasks. It corresponds to the combination of relaxed mode and no-extra-task status, which is used as a *baseline* in our tests.

Figure 7(a) shows the average time for a successful login attempt



(a) Login time distribution (sec) (b) Average success rate (c) Edit distance distribution

Figure 7: Average login time, success rate, and edit distance under the normal condition



(a) Average login time (sec) (b) Average success rate (c) Average edit distance

Figure 8: Impact of time pressure

in the normal condition. For all the three schemes, most participants are able to finish the login within 13 seconds. Figure 7(b) and 7(c) show the corresponding login accuracy. Since our experiment limits the number of login attempts to 5 in order to prevent the participants from feeling exhausted or bored, even a single mistake would take the login success rate down to 80%. Our results indicate that most participants make *at most* one mistake when they use our schemes for the first time after a short training. This is shown by 97.5% average round success rate and 0.13 average edit distance in the worst case. Particularly, for the distribution of average edit distance of NumPad-Shift, 27 participants among 40 samples (after removing the experimental data when NumPad-Shift appears as the first test group) has an average edit distance of zero (i.e. no mistakes during all tests under the test condition), which are shown as a cluster of *outliers* at the bottom of the box chart. The login accuracy is expected to increase after the participants get more familiar with the schemes.

5.4.2 *Influence of Time Pressure*

Figure 8 shows the impact of time pressure without any secondary tasks. The results show that the participants behave much hastily in the presence of time pressure. The average time for a successful login attempt becomes shorter and the login accuracy is decreased. The statistical tests show the difference in login time is significant (p=.017 for NumPad-Add and p<.001 for LetterPad-Shift) but the difference in login accuracy is not. Therefore, **H1** is supported while **H2** is not.

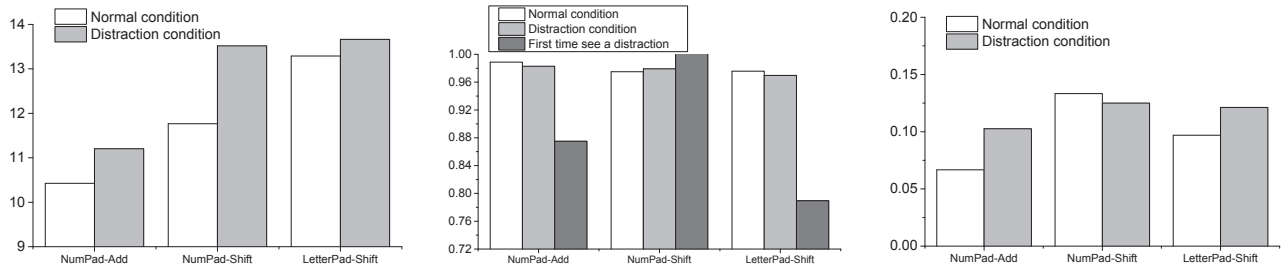
The insignificant results in login accuracy are due to the *ceiling* effect [1], which implies the tests are not sufficiently difficult to distinguish the influence of different test conditions. This effect could be caused by our scheme design, which is not difficult for the participants to use so that the majority of the participants did not make any mistakes during all the tests. This effect will be further discussed in Appendix A. However, even without statistical significance, we still observe the average results of login accuracy become worse for all three tested schemes. Considering the simple design of our schemes, this indicates that time pressure may have a larger influence on the login accuracy of a more complex scheme.

5.4.3 *Influence of Distraction*

Figure 9 shows the impact of distraction without time pressure. Many participants made a mistake when they saw a distraction task for the first time (however, NumPad-Shift is an exception). For NumPad-Add and LetterPad-Shift shown in Figure 9(b), the round success rate returns to a comparable level as the normal condition, after the first time the distraction task appears. This indicates that the distraction task is no longer a surprise for the participants. However, even after the participants get familiar with the distraction tasks, compared to the normal condition, the success rate is still lower, the average edit distance is larger, and the average login time is longer. But the statistical tests show these differences are not significant. Therefore, **H3** and **H4** are not supported in our experiments.

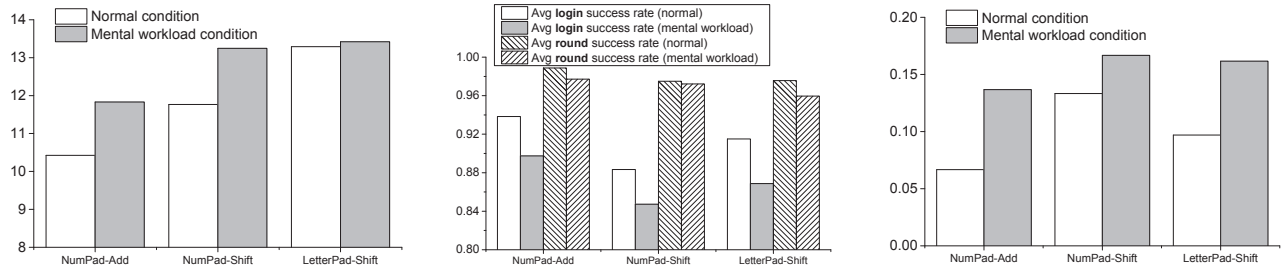
5.4.4 *Influence of Mental Workload*

Figure 10 shows the impact of mental workload without time pressure. The average login time becomes significantly longer with mental workload (p=.003 for NumPad-Add) due to context switch in users' mind between password inputs and secondary CRT tasks. An extra startup time is required to release the central executive after each CRT task. Our experiment simulates the case when users cannot get rid of other thoughts during password entry. The actual effect of mental workload depends on the status of users' mind. The impact may be elevated when the actual mental workload is higher than our CRT tasks. On the other hand, the login accuracy is lower compared to the normal condition but the difference is not significant due to the same ceiling effect mentioned in Section 5.4.2. Therefore, **H5** is supported and **H6** is not. These results



(a) Average login time (sec) (b) Average **round** success rate (c) Average edit distance

Figure 9: Impact of distraction



(a) Average login time (sec) (b) Average success rate (c) Average edit distance

Figure 10: Impact of mental workload

show that persistent mental workload significantly slows the process of password entry for our schemes.

5.4.5 Performance under Combined Conditions

We also examine the overall impact when distraction or mental workload appears together with time pressure. As expected, compared to their counterparts without time pressure, the average login time becomes shorter (from 10.3 seconds to 11.7 seconds on average), the login success rate becomes even lower (from 81.3% to 87.5%), and the average edit distance becomes larger (from 0.151 to 0.243). The statistical tests show the difference in login time is significant (p=.009 for NumPad-Add, p=.019 for NumPad-Shift, and p<.001 for LetterPad-Shift) and the difference in login accuracy is still not significant due to the ceiling effect explained in Section 5.4.2. Therefore, **H7** is supported but **H8** is not. These results show time pressure is still an effective stimulus to speed password entry even in the presence of secondary tasks.

5.4.6 Effectiveness of Secondary Tasks

Figure 11 shows the distribution of the accuracy rate which represents the percentage of secondary tasks being correctly performed by a participant under certain test condition. The overall average accuracy rate is 98.3% across all these test conditions. It implies that the participants did pay attention to these tasks, as they were told that the performance of these tasks also contributes to their scores in the ranking system. Therefore, these CRT tasks work as intended in disturbing participants' mind during password entry.

5.4.7 Memory Interference by Mental Calculation

Figure 12 shows how frequently a participant presses the "*show my password*" button during all tests in a test group. Note that the participants are not allowed to write down their assigned passwords, but they can always click that button in case they forgot their passwords. The overall average value for the total number of

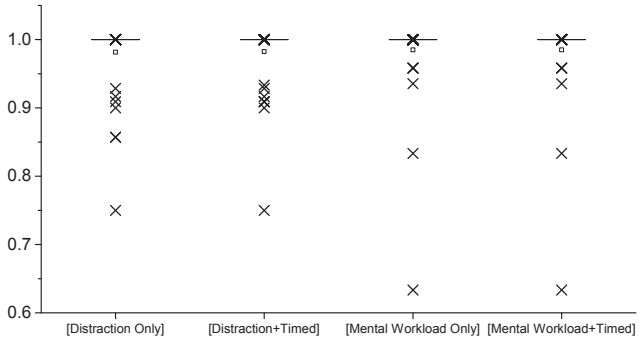


Figure 11: Accuracy rate of performing secondary tasks

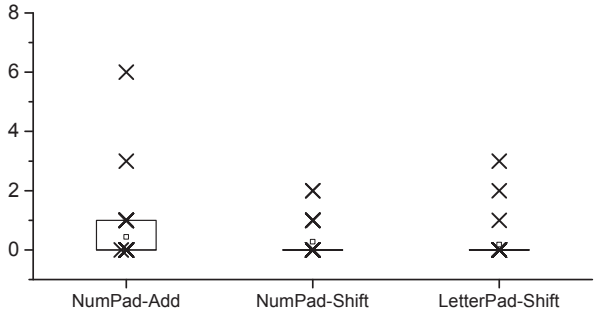


Figure 12: Total number of times for each participant to press the "*show my password*" button

times to press the "*show my password*" button is only 0.31 across all three test groups. As shown in Figure 12, most users did not use

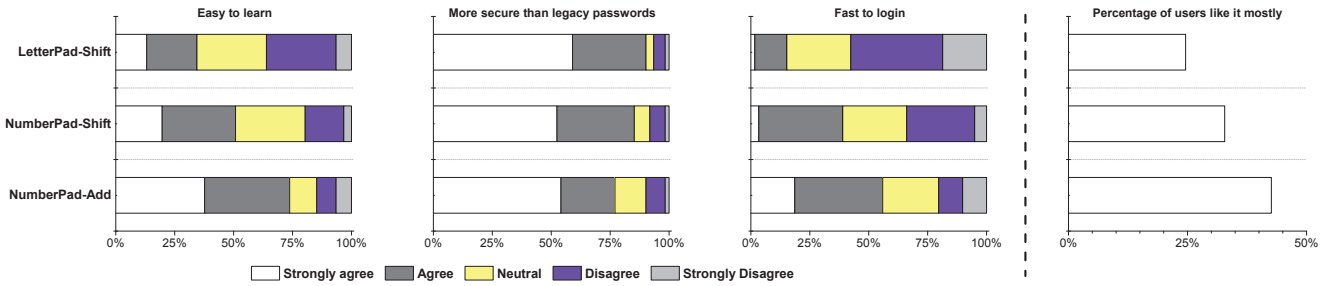


Figure 13: Perception of participants

this button during the tests. This implies that the mental calculation involved in the hidden transformation of our schemes does not pose a significant interference on participants' capability of recalling their passwords.

5.4.8 User Perception

Figure 13 shows the perception of participants collected from questionnaires. The results indicate that the participants generally feel that our schemes are secure and easy to use. While NumPad-Add is the most popular, the other two schemes also have their favorite users.

5.5 Comparison with Legacy Passwords

Table 2 gives a comparison between CoverPad and legacy passwords based on the *usability-deployability-security* metrics proposed in [10], where a metric is not shown if neither our schemes nor legacy passwords offer corresponding benefit. We have the following observations in comparison. 1) Our schemes are rated as not *mature* since they are just proposed and have not been widely deployed. 2) Our schemes are not *server-compatible*, as most current servers support only static and replayable passwords, which could be changed in the near future. 3) Our schemes are *quasi-resilient-to-internal-observation* in a sense that any key logger or malware which fails to capture the hidden transformation causes no password leakage. Overall, this table shows that our schemes significantly improve the security strength while retaining most benefits of legacy passwords.

	Nothing-to-Carry	Easy-to-Learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Internal-Observation	Resilient-to-Theft	No-Trusted-Third-Party	Requiring-Explicit-Consent	Unlinkable
CoverPad Schemes	●	●	○	○	●	●	●		●		●	●	○	○	●	●	●	●
Legacy Passwords	●	●	●	○	●	●	●	●	●	●	●		○		●	●	●	●

Table 2: Comparison between CoverPad and legacy passwords using usability-deployability-security metrics [10]. ● = offer the benefit, ○ = almost offer the benefit, *no circle* = does not offer the benefit

6. DISCUSSION

6.1 Eavesdropping Attacks

Eavesdropping attacks such as vision-based eavesdropping may require the physical presence of an adversary, which limits the scale of their threat. However, the scale of attacks is not the only factor that determines the impact of attacks, which is also decided by the severity of potential losses. If a victim is an important person in a company, password leakage may lead to disclosing sensitive corporate data, which would provide sufficient incentives to an adversary. While internal attacks such as malware and logic key logger could be prevented by properly updating and configuring the computing system [30, 3, 6, 37] used by the victim, it is difficult to effectively mitigate the threat of external eavesdropping attacks due to inevitable exposure of human-computer interaction during the entry of legacy passwords. This threat becomes more serious in scenarios when a mobile device is used in public places.

Nevertheless, the threat of external eavesdropping attacks can be effectively mitigated with CoverPad. Besides enhanced security features, our scheme retains most benefits of legacy passwords and can be implemented on *commodity* devices. Our scheme is not only applicable to mobile devices but also other devices equipped with touch screen. For example, many ATM machines have been deployed with touch screen. Our scheme can be deployed on these machines to mitigate the threat of the ATM skimming attack [25].

6.2 Device Screen Size

Although we implement our scheme on Apple iPad, it could be easily adapted to other screen sizes, as illustrated in Figure 14. For a mobile phone with a small touch screen like Apple iPhone, a user can use a hand A to perform the hand-shielding gesture, and use the other hand B to hold the phone. The thumb on hand B can be used to press the response keys. For a mobile phone with a larger touch screen like Samsumg Galaxy Note II, a user may not be able to click all the keys with the thumb of hand B that holds the device. To deal with this situation, he only needs to use one hand A to perform the hand-shielding gesture and key pressing sequentially. Once the user raises his hand before pressing a key, the hidden transformation immediately disappears because the gesture is no longer detected by the touch screen. Meantime, the user does not need to worry about whether the actual keys pressed or the finger movements during key pressing may be observed by an adversary, as the sequence of pressed keys alone does not leak any information about the underlying password as analyzed in Section 4.

6.3 Limitations

Ecological validity is a challenging issue in any user study. Like most prior research [19, 27, 26, 15, 24], our experiments engage only university students. These participants are younger and more educated compared to the general population. Therefore, usability evaluation may vary with other populations. Our experiments are

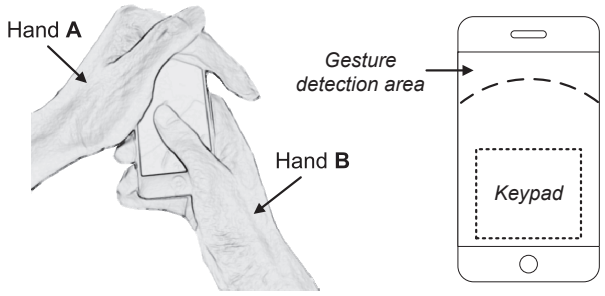


Figure 14: Conceptual demonstration on a small screen device

also restricted by the sample size, which may affect the results of statistical tests. Typical examples are the insignificant results on the login accuracy of our schemes. Moreover, our user study does not include experiments on memory effects (e.g. forgetting). Since our scheme uses the same alphabet and password composition as legacy passwords, the users may use the same coping strategies to help themselves to memorize the passwords in our scheme. The impact of memory effects on the user performance would be similar to legacy passwords as shown in the prior literature [14, 34].

7. RELATED WORK

In this section, we summarize closely related work on achieving leakage resilience of password entry in three different aspects.

Although the problem of achieving leakage resilience of password entry was proposed two decades ago [29], it is still a challenge to design a practical solution till now. Early work in this direction [19, 27, 38, 39, 5] focused on designing schemes solely rely on the cognitive capability of human beings. Unfortunately, all such schemes with acceptable usability have been broken [27, 38, 39, 5]. Recent investigations [11, 40] provided strong evidence for the necessity to construct a protected environment to hide certain user interaction during password entry in order to achieve both security and usability. The establishment of such protected environment may require the features only available from new user interface technologies. A few schemes [26, 33, 13, 15, 24, 8, 7] were designed in this strategy. Among them, our scheme design was mostly inspired by the concept of physical metaphor introduced in [24]. Our scheme distinguishes itself from prior work in the sense that it not only achieves leakage resilience but also retains most benefits of legacy passwords, while some of prior schemes [33, 15] are flawed in terms of security, and the others incur extra usability costs due to various reasons including: 1) using an uncommon device such as gaze tracker [26, 13], haptic motor [8], and large pressure-sensitive screen [24], 2) requiring an extra accessory device [7], and 3) inoperable in a non-stationary environment [8].

On the other hand, the procedure of applying random transformations on a fixed password used in our scheme design is a classic idea to prevent password leakage, but it is not easy to be realized in a *human-friendly* manner without the new user interface technologies, which are only available on modern computing devices. These new technologies give our scheme advantages when compared to recently patented schemes. Take GridCode [17] as an example, which asks users to memorize extra secrets (besides the passwords) in order to perform the transformations specified in its scheme design, while our scheme does not have such requirement. Another advantage of our scheme is that each character of the password uses a different hidden transformation during an authentication attempt, while GridCode uses the same transformation for all the characters in the password. If a hidden transformation in GridCode is disclosed, the entire password will be exposed. However, if a hidden transformation in our scheme is disclosed, only the single character associated with the transformation will be exposed. These two fundamental differences show both security and usability advantages of our scheme.

In terms of design principles, Roth et al. [32] proposed to use a cognitive trapdoor game to transform the knowledge of the underlying password into obfuscated responses. Li and Shum [27] later suggested three other principles including time-variant responses, randomness in challenges and responses, and indistinguishability against statistical analysis. Yan et al. [40] further extended the coverage by including the design principles against brute force attacks, and provided concrete guidelines against generic statistical attacks. Our proposed scheme follows all these design principles to avoid corresponding security flaws.

Bonneau et al. [10] recently proposed a generic framework for evaluating user authentication proposals and emphasized the importance of retaining the benefits of legacy passwords. Their framework introduced twenty-five benefits covering usability, deployability and security. This framework is used in our study to guide the scheme design in retaining the benefits of legacy passwords. Other research on password-based user authentication can be found in a recent survey paper [9], which summarized the development of new password schemes in the past decade.

8. CONCLUSION

In this paper, we proposed a leakage-resilient password entry scheme leveraging on the touch screen feature of mobile devices. It improves leakage resilience while preserving most benefits of legacy passwords. Three variants of this scheme were implemented. The practicability of our scheme was evaluated in an extended user study that incorporates new experiments to examine the influence of additional test conditions related to time pressure, distraction, and mental workload. These conditions were tested for the first time in the evaluation of user authentication schemes. Among these conditions, time pressure and mental workload were shown to have significant impacts on user performance. Therefore, we suggest including these conditions in the evaluation of user authentication schemes in the future research.

9. ACKNOWLEDGEMENTS

Yingjiu Li's work is supported in part by SMU Research Office under number 12-C220-SMU-006.

10. REFERENCES

[1] Ceiling effect. http://en.wikipedia.org/wiki/Ceiling_effect.

[2] Androidcommunity. Samsung galaxy siii display specs. http://androidcommunity.com/samsung-galaxy-siii-display-specs-edge-out-iphone-5-20121002/.

[3] Apple. Mac os x. http://www.apple.com/osx/.

[4] A. D. Baddeley and G. Hitch. Working memory. *The psychology of learning and motivation*, 8:47–89, 1974.

[5] 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 2008 Annual Computer Security Applications Conference*, pages 433–442, 2008.

[6] O. Begemann. Remote view controllers in ios 6. http://oleb.net/blog/2012/10/remote-view-controllers-in-ios-6.

[7] 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 fifth international conference on Tangible, embedded, and embodied interaction*, pages 197–200, 2011.

[8] A. Bianchi, I. Oakley, and D.-S. Kwon. Obfuscating authentication through haptics, sound and light. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 1105–1110, 2011.

[9] R. Biddle, S. Chiasson, and P. C. van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys*, 44(4), 2012.

[10] J. Bonneau, C. Herley, P. van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings of IEEE Symposium on Security and Privacy*, 2012.

[11] B. Coskun and C. Herley. Can "something you know" be saved? In *Proceedings of the 11th international conference on Information Security*, pages 421–440, 2008.

[12] F. I. Craik and J. M. McDowd. Age differences in recall and recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(3):474–479, 1987.

[13] A. De Luca, M. Denzel, and H. Hussmann. Look into my eyes!: can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 7:1–7:12, 2009.

[14] A. De Luca, M. Langheinrich, and H. Hussmann. Towards understanding atm security: a field study of real world atm use. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, 2010.

[15] A. De Luca, E. von Zezschwitz, and H. Husmann. Vibrapass: secure authentication based on shared lies. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 913–916, 2009.

[16] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov. The most dangerous code in the world: validating ssl certificates in non-browser software. In *Proceedings of the 19th ACM Conference on Computer and Communications Security*, pages 38–49, 2012.

[17] L. Ginzburg, P. Sitar, and G. K. Flanagin. User authentication system and method. US Patent 7,725,712, SyferLock Technology Corporation, 2010.

[18] Google. Google glass. http://plus.google.com/+projectglass.

[19] N. J. Hopper and M. Blum. Secure human identification protocols. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pages 52–66, 2001.

[20] H. B. Hotel. ipad - free for every hotel guest. http://www.hollmann-beletage.at/en/ipad.

[21] I. Imbo and A. Vandierendonck. The role of phonological and executive working memory resources in simple arithmetic strategies. *European Journal Of Cognitive Psychology*, 19(6):910–933, 2007.

[22] A. Imran. ipads can now be used as public kiosks. http://www.redmondpie.com/ipad-public-kiosks-video/.

[23] A. R. Jensen. Process differences and individual differences in some cognitive tasks. *Intelligence*, 11(2):107–136, 1987.

[24] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. W. Nicholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1093–1102, 2010.

[25] Krebs. Would you have spotted the fraud? http://krebsonsecurity.com/2010/01/would-you-have-spotted-the-fraud.

[26] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 13–19, 2007.

[27] S. Li and H. yeung Shum. Secure human-computer identification (interface) systems against peeping attacks: SecHCI. In *Cryptology ePrint Archive, Report 2005/268*, 2005.

[28] J. Long and J. Wiles. *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*. Syngress, 2008.

[29] T. Matsumoto and H. Imai. Human identification through insecure channel. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, pages 409–421, 1991.

[30] Microsoft. Windows 8. http://windows.microsoft.com.

[31] F. Miller. *Telegraphic code to insure privacy and secrecy in the transmission of telegrams*. C.M. Cornwell, 1882.

[32] 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*, pages 236–245, 2004.

[33] H. Sasamoto, N. Christin, and E. Hayashi. Undercover: authentication usable in front of prying eyes. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 183–192, 2008.

[34] R. Shay, P. G. Kelley, S. Komanduri, M. L. Mazurek, B. Ur, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. Correct horse battery staple: exploring the usability of system-assigned passphrases. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 2012.

[35] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th USENIX Security Symposium*, 2001.

[36] Spycop. Hardware keylogger detection. http://spycop.com/keyloggerremoval.htm.

[37] TCG. Trusted computing group. http://www.trustedcomputinggroup.org.

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

[39] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the working conference on Advanced visual interfaces*, pages 177–184, 2006.

[40] Q. Yan, J. Han, Y. Li, and R. H. Deng. On limitations of designing leakage-resilient password systems: Attacks, principles and usability. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium*, 2012.

[41] ZDNet. More ipad love: Now hotels offer ipad to customers. http://www.zdnet.com/blog/apple/more-ipad-love-now-hotels-offer-ipad-to-customers/6850.

APPENDIX

A. STATISTICAL TEST RESULTS

In this section, we provide the detailed results of statistical tests. Table 3 shows the results for login time, which indicates that the same test condition may have difference impact on the login time of different schemes.

Average login time of NumPad-Add - *omnibus* KW χ^2_5=32.423, p<.001

normal (10.4)	timed (9.2)	U=551, p=.017 ★
	distraction (11.2)	U=679, p=.184
	distraction+timed (10.3)	U=878, p=.989
	mental workload (11.8)	U=515, p=.003 ★
	mental workload+timed (10.7)	U=696, p=.319
distraction (11.2)	distraction+timed (10.3)	U=718, p=.107
mental workload (11.8)	mental workload+timed (10.7)	U=558, p=.009 ★

Average login time of NumPad-Shift - *omnibus* KW χ^2_5=11.965, p=.034

normal (11.7)	timed (11.2)	U=666, p=.199
	distraction (13.5)	U=645, p=.137
	distraction+timed (11.7)	U=727, p=.485
	mental workload (13.3)	U=655, p=.164
	mental workload+timed (11.4)	U=644, p=.135
distraction (13.5)	distraction+timed (11.7)	U=565, p=.024 ★
mental workload (13.3)	mental workload+timed (11.4)	U=555, p=.019 ★

Average login time of LetterPad-Shift - *omnibus* KW χ^2_5=49.252, p<.001

normal (13.2)	timed (10.1)	U=294, p<.001 ★
	distraction (13.6)	U=774, p=.667
	distraction+timed (11.0)	U=413, p<.001 ★
	mental workload (13.4)	U=653, p=.116
	mental workload+timed (11.5)	U=472, p=.002 ★
distraction (13.6)	distraction+timed (11.0)	U=422, p<.001 ★
mental workload (13.4)	mental workload+timed (11.5)	U=631, p=.075

Table 3: The results of statistical tests for login time (sec). All pairwise tests are Mann-Whitney U. The statistically significant results are marked with ★.

The results of statistical tests on login accuracy are not shown as none of them indicate significance. This is caused by the ceiling effect, which can be observed from the data shown in Table 4. Even in the worst case, 50.0% participants did not make any mistakes during all tests in the test condition, which implies our tests are not sufficiently difficult to distinguish these test conditions regarding their influence on the login accuracy of our schemes. This could be caused by the simple design of our schemes such that they are easy to use even in the presence of time pressure, distraction, and mental workload. However, it does not necessarily imply that these factors will not significantly influence the login accuracy of other user authentication schemes. Since the average results of login accuracy are observed to be worse due to the presence of these factors in our tests, we expect they would have a more significant influence on other schemes with higher complexity.

	NumPad-Add	NumPad-Shift	LetterPad-Shift
normal	82.9%	67.5%	75.6%
timed	78.0%	62.5%	53.7%
distraction	80.5%	70.0%	63.4%
distraction+timed	70.7%	55.0%	58.5%
mental workload	75.6%	57.5%	65.9%
mental workload+timed	65.9%	50.0%	51.2%

Table 4: Evidence for the ceiling effect in statistical tests on login accuracy. Each cell in this table shows the percentages of the participants who did not make any mistakes in a test condition.