Hindawi
Mobile Information Systems
Volume 2017, Article ID 6047312, 17 pages
https://doi.org/10.1155/2017/6047312

Research Article

Analysis and Improvement on a Unimodal Haptic PIN-Entry Method

Mun-Kyu Lee,[1] Jin Yoo,[2] and Hyeonjin Nam[3]

[1]*Department of Computer Engineering, Inha University, Incheon 22212, Republic of Korea*
[2]*LG Electronics, Seoul 06267, Republic of Korea*
[3]*Vieworks, Anyang 14055, Republic of Korea*

Correspondence should be addressed to Mun-Kyu Lee; mklee@inha.ac.kr

Received 14 March 2017; Revised 24 July 2017; Accepted 15 August 2017; Published 2 October 2017

Academic Editor: Stefania Sardellitti

User authentication is a process in which a user of a system proves his/her identity to acquire access permission. An effective user authentication method should be both secure and usable. In an attempt to achieve these two objectives, Bianchi et al. recently proposed novel unimodal PIN-entry methods that use either audio or vibration cues. This paper analyzes the security of their method, in particular, the vibration version of one of their proposals, Timelock. A probabilistic analysis and real attack experiment reveal that the security level guaranteed by Timelock is lower than that claimed in Bianchi et al.'s paper. As countermeasures to this problem, three PIN-entry methods are proposed and a usability study is performed. According to the result of this study, a simple modification may improve the security significantly while retaining the design philosophy of unimodal systems. In addition, the proposed methods address the PIN compatibility issue of Timelock and they can be used to enter a legacy numerical PIN without any change in the PIN.

1. Introduction

User authentication is a process in which a user of a system proves his/her identity to acquire access permission for that system. For user authentication, three approaches are typically adopted, that is, knowledge-based (e.g., password), object-based (e.g., ID card), and biometric-based (e.g., fingerprint) authentication models, among which the most prevalent form is the first [1]. Personal Identification Numbers (PINs) are a special form of the first category, and they are used for various purposes such as Automatic Teller Machines (ATMs), digital door locks, and smartphones. However, the traditional PIN-entry mechanism, in which a user directly touches the PIN digits on a numeric keypad, is very vulnerable to observation attacks by a shoulder-surfing attacker [2]. That is, a shoulder-surfing attacker may memorize a victim's PIN after observing the log-on procedure over the victim's shoulder. This weakness of PINs has motivated the development of randomized authentication methods that adopt challenge-response procedures [2–5]. In these methods, a user is provided with a random challenge and enters a response that can only be computed by combining the challenge and secret PIN. The amount of information in a challenge-response pair should be sufficiently large so that an attacker who does not know the PIN cannot deduce any useful information about the PIN from the observed pair. For usability, however, the challenge is designed to be a simple question that a legitimate human user can easily answer. One of the well-known challenge-response methods is the binary method proposed by Roth et al. [2]. The layout of this method is similar to that of the legacy 4×3 PIN pad except that each digit, $0, 1, \ldots, 9$, is displayed with a background color, either black or white. Among the ten numbers, five numbers are colored black and the other five white, where the color assignment is decided randomly by the authenticator. The user recognizes the background color of the number s/he wants to enter and enters that color by touching a button, either "Black" or "White," instead of entering the PIN digit directly. Then, the user's binary input is consistent with five numbers. That is, from the authenticators' point of view, the user's PIN digit could be one of these five numbers. Four rounds of the above procedure enable the authenticator to

uniquely identify the user's target number as $2^4 > 10$. As a result, a user has to perform 16 rounds to enter a four-digit PIN. A PIN-entry method that involves randomness can also be an effective countermeasure against a smudge attack where an attacker analyzes finger smudges on a touchscreen [6, 7].

Recently, it was discovered that the binary method is actually vulnerable to well-trained human attackers, and PIN-entry methods that are more resistant to such attacks have been proposed [8, 9]. The design goal of these methods as well as the binary method is to prevent an observation attack by a human observer who does not use a recording device, and thus such countermeasures are suitable for the few situations in which an attacker cannot use a recording device such as a phone camera [8]. In contrast, the above methods cannot protect a PIN if an attacker can gather challenge-response pairs using a hidden recording device without being detected and s/he can perform an offline analysis on the obtained data. An attacker can further enhance his/her attack capability by using automated analysis tools and nontrivial visual information such as reflections in a victim's sunglasses [10, 11]. Therefore, a PIN-entry method with recording resilience is required. However, it was proved in [8] that the resistance to recording attacks can only be achieved by sacrificing the resistance to random guessing attacks *if the challenge-response pairs are observable*. In other words, if a PIN-entry method is designed to be secure against recording attacks, the attacker's chance of successful log-on by entering a random PIN is increased, which causes another kind of problem. As a result, the only solution for recording resilience without sacrificing random guessing resilience is to design a PIN-entry method whose challenge-response pairs cannot be observed by an attacker. To achieve this goal, many authentication methods using secondary data channels such as audio and haptic cues have been proposed [12–18]. These methods aim to protect either challenges or responses from an attacker's visual observation.

In this paper, we concentrate on the authentication methods using vibration. Recently, Bianchi et al. proposed three authentication methods, Spinlock, Colorlock, and Timelock, which are based on a simple counting mechanism [19]. Although the proposal in [19] provides both variants, that is, audio and haptic versions, the above simple counting mechanism is effective, in particular, for the haptic mode. The methods in [19] adopt a novel unimodal approach. That is, they depend only on vibration cues and do not use any visual cues [19]. They are based on the finding in cognitive science that users engaged in nontrivial cognitive tasks perform worse when asked to split their focus over multiple sensory channels [20]. Roughly speaking, the main concept of [19] is to ask users to simply count the number of times that nonvisual cues (vibrations) occur while keeping his/her finger on a button. When the user releases the button, the number of vibrations accumulated up to that point is recognized as the user's PIN digit. Because an attacker does not have any access to the vibration channel, the challenge is kept secret. On the other hand, a response, that is, the time during which a user's finger stays on the button, may be revealed to the attacker. In [19], it was claimed that a proper randomization would reduce the correlation between a PIN and its entry time, keeping the counting-based approach secure against observation attacks. This claim was supported by an attack experiment involving three human attackers, but no numerical evidence was given about the exact level of security guaranteed by the proposed mechanism.

The motivation for secondary channel-based methods is their higher security which justifies their relatively low usability compared to other methods such as regular PIN-entry using a legacy PIN pad. Therefore, it would be more desirable to either give rigorous proof or quantify the guaranteed security level. The purpose of this study is to quantify and improve the security of the counting-based approach. We show that a much greater bias exists in the PIN-entry time according to the PIN value than that recognized in [19], implying that time randomization is not sufficient to hide the correlation. We verify this fact by both a mathematical analysis and a real attack experiment.

The idea of attacking a PIN-entry method using its time variation was previously used in [21, 22], and this type of attack is called a *timing attack*. Our attack may also be viewed as a timing attack, but the difference of our attack from those in [21, 22] is that we use the variation in the transmission time of a system-generated challenge, and thus the attack is supposed to work equivalently for any victim user. On the contrary, in the target systems of the attacks in [21, 22], the challenge transmission time does not vary, but the feasibility of an attack depends on a victim user's response time due to the variations in human cognitive load.

To resolve the issue of time bias while maintaining the advantages of the original system, we propose three countermeasures such that the PIN-entry time is independent of the PIN digits. Our first, second, and third countermeasures are designed to utilize three intrinsic characteristics of numbers, that is, cardinal (for quantity), ordinal (for order or rank), and nominal (for symbolic use) characteristics, respectively. In the first countermeasure, Addlock, a user counts the number of vibration cues for a fixed time interval and enters the sum of the counter value and the PIN digit. The second countermeasure, Counter Phone Lock, is a novel combination of the counting-based approach with the previous solution, Phone Lock [16]. That is, it uses the order between PIN numerals, $0, 1, 2, \ldots, 9$. This is in contrast to the original counting-based methods in [19] and our first countermeasure, in which the PIN numerals are regarded as cardinal numbers. The third countermeasure, Map lock, regards the PIN digits as symbols and uses only a single vibration cue as a hidden indicator. Addlock and Counter Phone Lock keep the unimodal approach in [19], while Map lock asks a user to perform bimodal tasks involving visual and vibration challenges.

To evaluate and compare the usability of the proposed countermeasures, we conduct a user study involving 18 volunteers. According to the analysis results, Addlock is the fastest among the three methods, and Counter Phone Lock was the slowest. Regarding the required number of trials per PIN-entry session, Addlock is marginally better than the other two methods. An interesting result was that even though the PIN-entry time of Map lock was significantly faster than that of Counter Phone Lock, participants gave

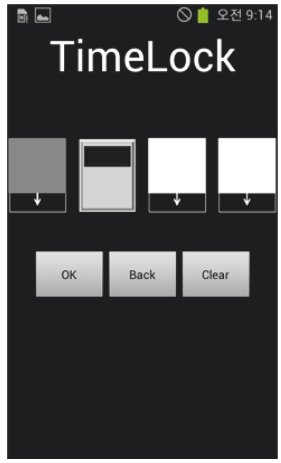


Figure 1: Screenshot of a Timelock implementation when the leftmost two digits have been entered (the same screen layout was used for the experiments in [8, 18]. See Figure 3 in [8] and Figure 7 in [18]).

similar usability scores for these two methods. This implies that users prefer unimodal methods than bimodal ones for vibration-based PIN-entry.

We also conducted attack experiments against the three proposed methods and verified that none of them was vulnerable under the settings similar to those where the method proposed in [19] was effectively attacked.

In addition to the improved security, the proposed countermeasures have another advantage in that they are compatible with the legacy PIN pad. In other words, they do not require any change in the definition of a PIN; they only provide a new interface for a traditional numeric PIN. In contrast, the counting-based methods in [19] define unique PINs involving additional nonnumeric PIN elements such as rotational direction (for Spinlock), color (for Colorlock), and button position (for Timelock). As a result, our countermeasures can be applied to existing systems, for example, banking by smartphone, without any significant change in the existing infrastructure. A user may then choose between a legacy PIN pad (in a secure environment such as a private room) and the new method (in an open place) without changing his/her PIN.

2. Review of Counting-Based Pin-Entry

In [19], Bianchi et al. introduced three variants, Spinlock, Colorlock, and Timelock, for counting-based PIN-entry. Among those three methods, Timelock was the most advanced. Therefore, we review and analyze Timelock here, although our analysis may be applied similarly to the other two variants. Figure 1 is a screenshot of an implementation of Timelock on an Android-based smartphone according to the description in [19]. It has four rectangular virtual buttons arranged in a row. Each button is mapped to each PIN digit, and a user inputs a digit by directly touching the corresponding button and keeping the finger on the button. While the finger remains pressed on the button, unimodal cues are delivered to the finger in appropriate intervals. The delivered cues are 113 ms long beeps for the audio version and 25 ms long vibration buzzes for the haptic version. When the user releases the button, the number of cues accumulated up to that point is entered as the user's PIN digit. A unique characteristic of Timelock is that the order in which the four buttons are touched also matters for authentication. That is, the four buttons must be entered in a predetermined order, which significantly enlarges the PIN space, although each PIN digit is restricted to a number in $\{1, 2, 3, 4, 5\}$. As a result, the size of the PIN space is $5^4 \times 4! = 15{,}000$.

It should be noted that the time the user's finger presses on the button reveals the corresponding PIN digit if the time interval between two consecutive cues is fixed. As a countermeasure to this possibility, Timelock adopts two cue randomization techniques: constant beats and random beats. In the constant beat mode, the intercue interval is randomly selected once per PIN digit, and the same interval is used for that digit. On the other hand, in the random beat mode, the interval is randomized every time a vibration cue is triggered. In both modes, the intercue interval is between 300 and 400 ms. To conceal further the correlation between a PIN and its entry time, an additional randomization, that is, an initial pause, is added. To be precise, the time from the user's button touch to the first cue activation is randomly selected from 0 to 1,500 ms for the constant beat mode and from 0 to 2,000 ms for the random beat mode. Timelock also has four error-correction mechanisms that allow users to change their input. For more details, refer to [19].

We briefly remark that the term *counting* was also used with another meaning in the literature. For example, in [23–25], a counting-based mechanism was defined as a k-out-of-n scheme, where a secret is composed of k objects out of n publicly known objects. During authentication, a user is given a challenge composed of a random subset of secret objects as well as decoy objects. The user then *counts* the number x of secret objects in this challenge and calculates a simple function of x. For example, in the Foxtail protocol [26], a user is asked to compute $\lfloor (x \bmod 4)/2 \rfloor \in \{0, 1\}$. The statistical attacks in [23, 24] are based on the observation that the distributions of secret objects and decoy objects in a challenge are different from each other when the responses are considered together. This enables an attacker to distinguish secret objects by observing many sessions, for example, hundreds of sessions. In [25], these attacks were further improved by adopting linear algebra techniques that transform a problem instance to a system of linear congruence.

3. Security Analysis of Counting-Based PIN-Entry

3.1. Reduction of PIN Space Size. It is easy to see that the order of buttons to be touched is revealed after an attacker observes only one PIN-entry session of Timelock. As a result, if the attacker attempts a log-on after observing a session, the probability of a successful log-on becomes $1/5^4 = 1/625$ as already mentioned in [19]. This figure is not a sufficiently small one when we take into account the fact that most

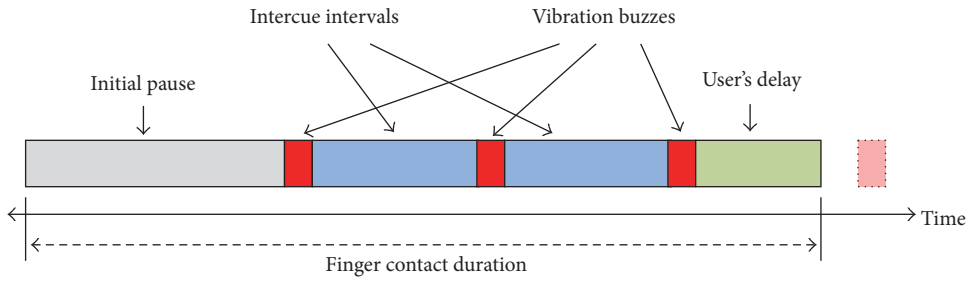
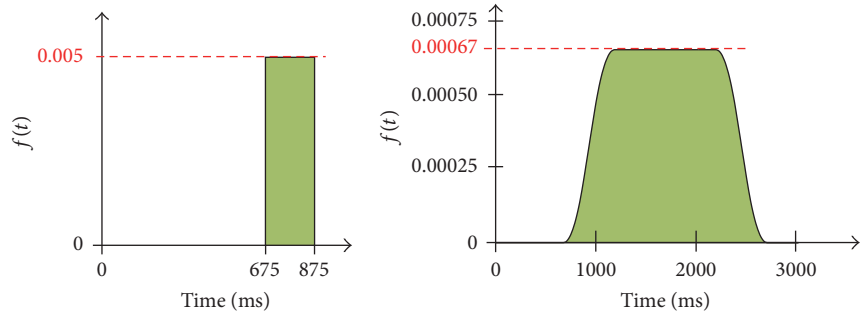


Figure 2: Breakdown of the entry of one PIN digit.



(a) Excluding the initial pause and user's delay

(b) Including the initial pause and user's delay

Figure 3: Distribution of the finger contact duration for PIN numeral "3."

systems allow multiple trials. For example, most Android-based smartphones ask a user to pass authentication within five trials. In this case, the success probability is 5/625 = 0.8%, which is quite high. Moreover, the attacker obtains additional five chances after 30 s has passed. The probability of success then becomes 10/625 = 1.6%.

Another issue with a nonstandard PIN is that it is not compatible with existing systems [8]. If the new PIN-entry method is supposed to be used for limited purposes, for example, to unlock a smartphone, defining a new PIN space such as in Timelock is not a problem. However, let us consider a case in which the new method is used for more general purposes. For example, a numeric PIN is frequently used to approve a financial transaction in both ATMs and smartphones. If a different set of PIN should be defined for banking by smartphone, all the information about the bank accounts and related software should be modified. Furthermore, some users may have difficulties in using the new system. Therefore, it would be desirable to keep the original PIN space and only change the PIN-entry interface. A user may then choose from between a legacy PIN pad and the new method without changing his/her PIN.

3.2. Effect of Cue Randomization. Another concern about Timelock is that the cue randomization (randomization of the intercue intervals and initial pauses) may not completely eliminate the correlation between a PIN digit and its entry time. Although it was already recognized in [19] that the Pearson correlation coefficients were far from zero, that is, 0.7 and 0.62 for the constant and random beat modes, respectively, it was claimed in [19] that Timelock was sufficiently secure because no PIN was successfully recovered in an attack experiment involving three attackers. In the experiment, the attackers were given recorded PIN-entry videos for 40 PIN-entry sessions as well as the *average* time that users needed to insert a PIN digit, but none of them successfully deduced a PIN. In this section, we estimate the security of Timelock against a better-prepared attacker who knows the exact probability distribution of PIN-entry time rather than simply its average value, which is a more reasonable assumption. For simplicity, we only provide the results from the constant beat mode of the vibration-based version.

Figure 2 shows a breakdown of the time required to enter a PIN digit when the numeral is "3." Note that we should consider the user's delay because the user may not release his/her finger immediately after the third buzz occurs. This delay should be smaller than an intercue interval, because otherwise the fourth buzz will be activated. In the constant beat mode, after an intercue interval between 300 and 400 ms is randomly selected, the same value is used for all intervals. Therefore, the time for three buzzes and two intercue intervals has the distribution shown in Figure 3(a), where $f(t)$ represents a probability density function for the *finger contact duration* t, that is, the time a user's finger presses on the button. The probability that $a \le t \le b$ can be computed by $\int_a^b f(t)dt$. By additionally considering the random initial

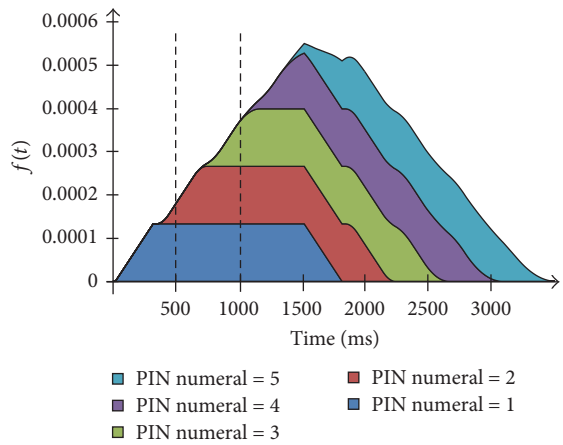


Figure 4: Accumulated distribution of the finger contact duration for Timelock, considering all possible PIN numerals.

pause (0 to 1,500 ms) and the user's random delay (up to the intercue interval), we obtain the distribution shown in Figure 3(b).

Similar distributions can be constructed for the other numbers. Figure 4 shows a merged distribution of the finger contact duration considering all possible PINs, where a PIN digit is selected from $\{1, \ldots, 5\}$. In Figure 4, the contribution of each PIN numeral to $f(t)$ is represented as a distinct color.

To help readers to understand this distribution, we present an example that computes the probability that $500 \leq t \leq 1{,}000$ ms, where t is the finger contact duration. Using the probability density function, we can estimate this probability as $\int_{500}^{1{,}000} f(t)dt \approx 0.136$. Furthermore, we can also compute the conditional probabilities Pr (PIN numeral = "1" | $500 \leq t \leq 1{,}000) \approx 0.489$, Pr (PIN numeral = "2" | $500 \leq t \leq 1{,}000) \approx 0.423$, and Pr (PIN numeral = "3" | $500 \leq t \leq 1{,}000) \approx 0.088$. PIN numerals 4 and 5 are not consistent with this range of contact duration, and thus their probabilities are zero.

Note that, unfortunately, the attacker also knows this distribution. When the attacker observes that the contact duration t satisfies $500 \leq t \leq 1{,}000$ ms, s/he will successfully determine the PIN digit with a quite high probability of 0.489 by only choosing the most probable number for the PIN digit. However, the attacker's success probability reduces for some ranges of t, that is, around 1,500 ms, because in this region, all five possible PINs may be consistent with a given t, and their conditional probabilities are almost uniform except when the PIN digit is 5. Averaging out all cases, we found out the expected success probability of an observation attack for a single PIN digit is approximately 0.4. Because the procedure to enter each PIN digit is independent of each other, the attacker can adopt a divide-and-conquer approach. The expected success probability of an observation attack for a 4-digit PIN is approximately $0.4^4 = 0.0256$, which is significantly greater than the expected success probability of a random guessing attack without any observed information, $0.2^4 = 0.0016$. Note that the above estimation assumed that only one PIN-entry session was observed. If an attacker may observe more than one session performed with a fixed PIN, as in [19], the probability of success will significantly increase.

To verify whether the above attack is practical, we designed an attack experiment similar to that explained in [19]. We implemented Timelock on a Samsung Galaxy S4 smartphone running Android 4.2.2. For the attack experiment, three volunteers were recruited to play the role of user victims. They were 25, 25, and 31 years old, respectively, and one of them was female. They were all engineering students of a local university and smartphone users. Each of these participants was assigned a random Timelock PIN.

When conducting user studies, we took measures to ensure that all possible ethical issues that we could consider were properly handled, although our studies did not go through a formal IRB (institutional review board) review. We attached flyers in our local university to recruit voluntary participants. Before participating in the actual study, the participants were informed of the purpose and procedure of the study. We did not collect personal identification information; we only collected demographic data such as age and gender in an anonymized form. The participants were informed what information would be collected and for what purpose it would be used. We obtained the participants' explicit consent for their participation. We did not use the participants' personal smartphones; we used a device dedicated for our study. We did not ask the participants to use their original PINs; only system-generated PINs were randomly assigned to participants. These measures were also applied to the user studies in Sections 5 and 6.

After an introduction to the PIN-entry mechanism of Timelock, the participant had some time for practice. Each participant then conducted PIN-entry tests until 20 successful PIN-entries were collected for his/her fixed PIN. The above procedure was recorded using a Sony α6000 digital camera (59 fps, 1920 × 1080 pixels). This camera was fixed on a tripod behind the participant and focused on the smartphone. The participants were informed that they were being filmed and asked not to obstruct their input from view. The recording was done in such a way that the participants were not recognizable from the video material. The video

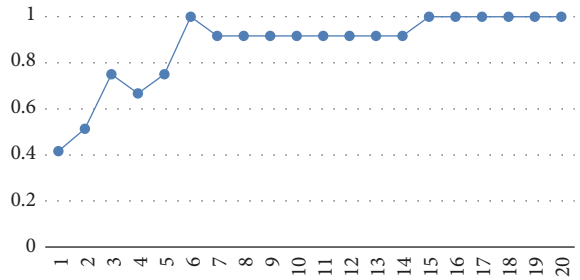


Figure 5: Success rate of observation attacks (for a single PIN digit) according to the number of recorded sessions.

material for each participant was approximately 4 min long. A ten-dollar coffee coupon was given to each participant as an incentive.

An analysis of the recorded sessions was performed by an attacker who well understood the working mechanism of Timelock. She was given the three video files as well as the probability distribution shown in Figure 4. The attacker was able to adopt a divide-and-conquer approach because each PIN digit was independent from the others. Regarding a target PIN digit, she played back the video at slow speed and manually noted the user's touch duration to enter that digit. She combined these touch duration data with the above probability distribution and deduced the value of the target PIN digit. For example, when the time to enter a PIN digit was approximately 1.0 s in the video, the probabilities that this PIN digit was "1," "2," and "3," were estimated to be 0.36, 0.36, and 0.28, respectively. There was no possibility that the PIN digit could be "4" or "5." Her strategy was then to choose the candidate with the maximum probability. That is, she selected "1" or "2." If the correct PIN digit was "1," her success rate would be 0.5. On the other hand, if the real PIN digit was "3," she would never succeed. However, she could refine further the attack accuracy by using the information accumulated over additional sessions. Figure 5 shows the average success rate according to the number of recorded sessions used for the attack. We can verify that a PIN digit was recovered with a probability of almost 100% after the fifth session. Even with only the first session, the success rate was significantly higher than 0.2, the success probability of a random guessing. More details are given in Appendix.

The above result contradicts the experimental results reported in [19]. It should be noted that, in the attack experiment in [19], an attacker was only provided with the *average* time that users needed to insert each of the five possible PIN digit values, not an exact distribution of that time. We remark that it is reasonable to assume that an actual attacker knows this distribution. Although our experiment was a limited one involving only one attacker and three victims, it indicates that Timelock is not as resistant to observation attacks as expected when the attacker was well-prepared.

4. Countermeasures

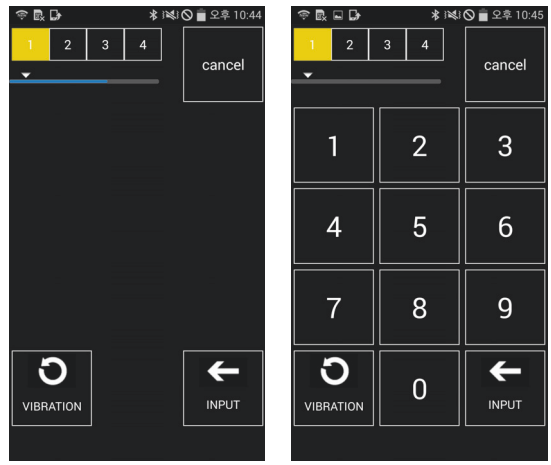
The essential issue of the counting-based approach in [19] is that we cannot prevent the numeral of a PIN digit and its entry time from having a nonnegligible correlation. In this section, we propose three countermeasures to address this issue.

4.1. Fixing the Duration of Vibration. The first countermeasure is to make the duration of vibration look independent of the value of a PIN digit to the attacker. This system leads a user to wait without any action during a predefined and fixed time interval. During this period, up to ten cues may be safely activated. Figure 6(a) shows an implementation of this idea. In the top left part, a progress bar is located. This bar is initially colored gray, but the blue part extends from the left as time goes by. To allow a user to be prepared, no cues occur during the initial 500 ms needed for the blue bar to reach a small white triangle marker. After the blue bar passes over the marker, cues are activated until the entire bar becomes blue. The duration of this cue-activation interval (from the marker to the right end) is fixed as 2,625 ms, which is exactly the time required for nine vibration cues (25 ms each) plus eight intercue intervals (300 ms each). However, the number of vibration cues actually activated in this interval is randomly selected from $\{0, 1, \ldots, 9\}$ by the device. A user recognizes the cues and counts the number of cues while holding the device.

For the user to input his/her PIN digit after the progress bar reaches the end, s/he is requested to do a simple addition, that is, the counter value + the value of the PIN digit, and enters the result using the interface shown in Figure 6(b). If the result is greater than nine, the user enters only the last digit, which is equivalent to addition modulo 10. If the user failed to count the cues, s/he can use the bottom left "vibration" button to ask the device to repeat the cues. In addition, the left arrow on the bottom right corner is an error-correction mechanism to change the user's current input. This procedure is repeated for each PIN digit. The top right "cancel" button cancels all previously entered digits.

This PIN-entry method was named Addlock, to reflect its working mechanism. From the attacker's viewpoint, whatever the user enters may be mapped to one of the PIN numerals in $\{0, 1, \ldots, 9\}$ with the same probability, 1/10, because s/he does not know how many cues occurred. In addition, the timing attack of the previous section is not possible as the progress of the bar is fixed, irrespective of the PIN numeral. As a result, Addlock is secure against observation attacks.

4.2. Using Ordinal Characteristics of PIN Items. It is also possible to design another effective countermeasure if we use the fact that there is a well-defined order of the PIN numerals, $0, 1, 2, \ldots, 9$. Figure 7 shows our second countermeasure that is motivated by this observation. It is essentially a simple modification of Phone Lock [16]. In the original Phone Lock [16], each PIN digit was not a numerical value but was selected from a set of ten tactons. Nine of the tactons were defined via combinations of the number of vibration buzzes (one, two, or three) and their durations (40, 80, or 160 ms). The tenth tacton was "no buzz." However, because recognition of the target pattern and its order from among a large set of haptic cues was challenging, the failure rate was quite high, that is, 10.38% for a 4-digit PIN [16]. However, the advantage of Phone Lock from a security viewpoint is that it



(a) (b)

Figure 6: Addlock: (a) cue activation during a fixed time interval and (b) interface for user's input.

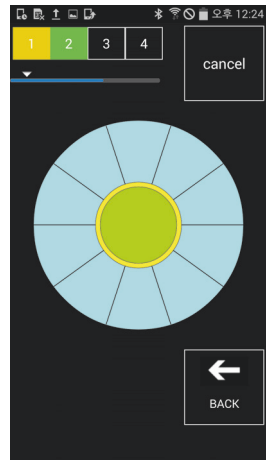


Figure 7: Counter Phone Lock: the second countermeasure that uses the relative order of PIN numerals.

completely randomizes the relationship between a PIN and the user's actions.

By combining the ideas of Timelock and Phone Lock, it is possible to design a PIN-entry method that is both secure and usable. Our second countermeasure, called Counter Phone Lock, is a realization of this idea. It has a ten-sectored wheel as in the original Phone Lock. Each sector will be allocated a distinct number in $\{0, 1, \ldots, 9\}$. However, this allocation is not visible. That is, the numbers are not shown on the screen. A PIN-entry session begins when a user taps anyone of the ten sectors. When the user taps a sector, the device randomly chooses a number $n \in \{0, 1, \ldots, 9\}$ and allocates n to this sector, activating n vibration cues. Ten cues are activated for $n = 0$. The allocation for the remaining nine sectors is then done in a deterministic way. Starting from the initially touched sector, adjacent sectors are allocated $n + 1 \bmod 10$, $n + 2 \bmod 10$, and so on, clockwise, and $n - 1 \bmod 10$, $n - 2 \bmod 10$, and so on, counterclockwise. This is equivalent to assign 0 through 9 in a clockwise direction, randomizing the position of 0 for each digit entry. After tapping a sector and recognizing the number of vibrations for the current sector, the user may find the target sector corresponding to his/her PIN digit by counting the relative distance between the two sectors. For example, if the user recognizes three vibrations by tapping a sector and the target number is 6, the target sector should be the third sector from the initial sector in the clockwise direction. When the user finds the target sector corresponding to his/her PIN digit, s/he enters the digit by dragging and dropping the sector to the small center circle. Before the user finalizes his/her choice, s/he may tap as many sectors as s/he wants and identify the numbers allocated to those sectors for double-check. As in Addlock, the top left progress bar gradually increases while the vibrations for a specific sector are being activated. The time duration for the progress bar to fully grow is fixed to 2,950 ms (time for ten vibration cues plus nine intercue intervals) irrespective of the

actual number of vibrations. This enforced delay provides the resistance to observation attacks. Before the progress bar has fully grown, no drag-and-drop action is recognized by the device. This procedure is repeated for each PIN digit, and the left arrow and "cancel" buttons play the same role as those of Addlock.

Note that the attacker does not know which PIN numeral corresponds to the initially touched sector. Consequently, the attacker does not know which number is the user's final choice. That is, the probability that the attacker may guess a correct 4-digit PIN is exactly 1/10,000 even after the attacker has observed multiple sessions. In other words, Counter Phone Lock is secure against observation attacks.

4.3. Interpreting PIN Digits as Symbols Instead of Numbers. In Timelock and Addlock, both a counter value and target PIN digit were interpreted as cardinal numbers. On the other hand, another characteristic of numbers, that is, order, was taken into account in Counter Phone Lock. For our third countermeasure, we attempt a more radical change: the PIN digits are no longer interpreted as numbers but are seen as symbols. Furthermore, the vibration channel is not used to transmit a multiple-cue counter but is used to transmit a single cue that is used as a hidden indicator.

Figure 8(a) is the initial interface of the third countermeasure, named Map lock (the main idea of this method was published as a patent [27]. It can be viewed as a vibration-based variant of [5], which used only a visual mapping) that is similar to a legacy PIN pad. Immediately after a user touches any region on this screen, ten alphabet characters, "A" through "J," are randomly mapped to PIN numerals, "1" through "0," as shown in Figure 8(b). This mapping is updated periodically and automatically as follows. After a predefined time interval (300 ms) since the initial mapping appeared, each character is replaced by the next character in the alphabet. For example, "H," mapped to "1" in Figure 8(b), is replaced by "I," as shown in Figure 8(c). Furthermore, "I" mapped to "2" in Figure 8(b) is now replaced by "J" in Figure 8(c). The order of characters is circular so that "J" that was mapped to "3" in Figure 8(b) is replaced by "A" instead of "K" in Figure 8(c). This update is repeated every 300 ms, until each number meets the tenth character. For example, the character under "2," which is initially "I," is updated to "J," "A," "B," . . ., "H," in turn, every 300 ms. The final arrangement shown in Figure 8(d) occurs after exactly 2,700 ms has passed since the initial mapping appeared. To provide a user with a secret challenge, the device generates a simple 25 ms long vibration buzz at some point during the above 2,700 ms long period. To be precise, the buzz is generated when exactly 110 ms has passed since the target mapping was shown. The selected values of these specific parameters, such as 110 ms for the buzz initiation time, were obtained from exhaustive trial-and-error processes to maximize perceptibility. At the moment a vibration occurs, the user should remember which character is mapped to his/her target number. For example, if the user's target PIN digit is 3 and a vibration occurs at the second update, shown in Figure 8(c), s/he should remember the character, "A," colocated with the target value, 3, at the moment when the simple vibration buzz is felt. When 1,000 ms has passed since the final mapping of the first stage occurred, a completely new random mapping is generated for the second stage, as in Figure 8(e), and similar updates are performed. A vibration buzz occurs at some moment. Thus, it takes exactly 3.7 s from the user's initial touch to the end of the first stage.

The above procedure is repeated for each PIN digit. After four stages, that is, 40 updates are completed during exactly 14.8 s, the user should remember four characters, that is, one for each PIN digit. S/he enters these four characters via the traditional keypad interface, as shown in Figure 8(f). The small upper four squares with numbers "1" to "4" in Figure 8(b) are stage indicators that also play the role of an error-correction mechanism. A colored square indicates that the current updates and vibration are for the stage numbered by that square. If the user touches any of the four squares, the updates are restarted from that stage. For example, if the user touches the square numbered "2" while the final stage is being performed, stages 2, 3, and 4 will be done again, resulting in new 30 updates. For the user's convenience, every stage is allocated a distinct color, and this color is used for alphabet characters as well as the indicator square. For example, Figures 8(b)–8(d) show that the first stage uses yellow. For the second stage, the corresponding square and characters are colored light green, as shown in Figure 8(e). When the user has entered an incorrect character, s/he can correct it by touching the left arrow button in Figure 8(f).

It is easy to see that Map lock is secure against observation attacks because whatever character the user inputs in Figure 8(f) appears under every number in the corresponding stage. Because the attacker does not know when a vibration has occurred, this character is consistent with any number in $\{0, 1, \ldots, 9\}$ with the same probability, 1/10. Therefore, observation does not give any useful information to an attacker.

We remark that Map lock somewhat deviates from the design philosophy of Timelock. That is, Map lock is not unimodal but bimodal because a user must recognize two kinds of stimuli (visual and haptic) simultaneously, although the haptic stimuli are very simple ones. In the next section, we examine how this change affects the overall authentication performance of users.

5. Usability Study for Proposed Countermeasures

To verify whether the proposed countermeasures are practical, we performed a usability study. To be precise, the goal of this study was to identify the most usable method among the three methods. For the study, we implemented the three PIN-entry methods as well as the traditional regular PIN pad on a Samsung Galaxy S4 smartphone running Android 4.2.2.

5.1. Procedure. The procedure of our study was designed as follows, using repeated measures design. First, we explained our study to each participant and collected demographic data. Each participant was assigned a random 4-digit numeric PIN and helped to remember that PIN by entering it ten times using the regular PIN pad. Then, for each of the three

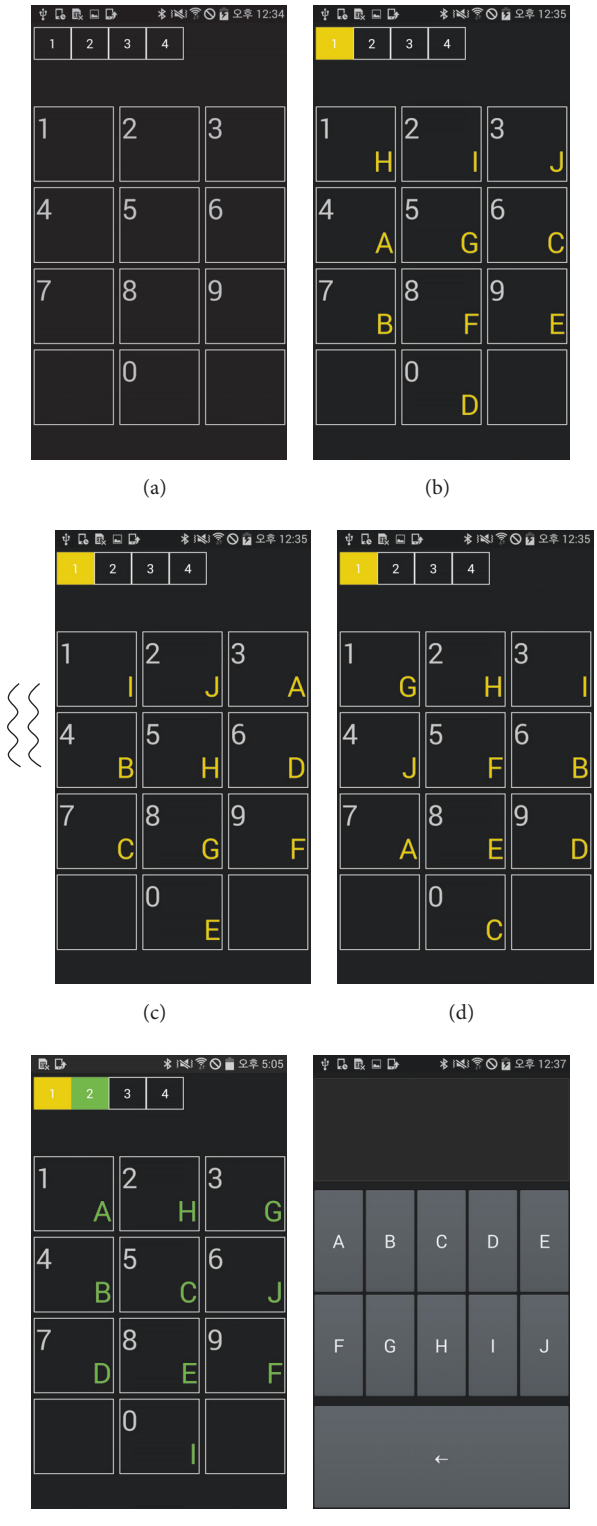


Figure 8: Map lock: (a) initial interface, (b) initial mapping for first PIN digit, (c) updated map, (d) final mapping for first PIN digit, (e) initial mapping for second PIN digit, and (f) user input interface.

Table 1: Results of descriptive statistical analysis (mean and standard deviation).

Analyzed data	Counter Phone Lock	Map lock	Addlock
Task completion time (s)	27.6 (9.4)	20.6 (4.1)	18.0 (3.1)
Trials per session	1.16 (0.47)	1.18 (0.48)	1.08 (0.36)
Usability score	2.67 (1.11)	2.78 (1.08)	3.61 (1.06)

Table 2: Results of inferential statistical analysis (A: Counter Phone Lock, B: Map lock, and C: Addlock, (H): after Holm correction, and (B): after Bonferroni correction).

Analyzed data	ANOVA	t-tests:	A versus B	A versus C	B versus C
Task completion time	$F = 115.597$, $p \ll 0.001$	(H)	$\mathbf{p \ll 0.001}$	$\mathbf{p \ll 0.001}$	$\mathbf{p \ll 0.001}$
		(B)	$\mathbf{p \ll 0.001}$	$\mathbf{p \ll 0.001}$	$\mathbf{p \ll 0.001}$
Trials per session	$F = 2.629$, $p = 0.073$	(H)	$p = 0.629$	$p = 0.061$	$\mathbf{p = 0.040}$
		(B)	$p = 1.000$	$p = 0.091$	$\mathbf{p = 0.040}$
Usability score	$F = 3.859$, $p = 0.028$	(H)	$p = 0.385$	$\mathbf{p = 0.024}$	$\mathbf{p = 0.030}$
		(B)	$p = 1.000$	$\mathbf{p = 0.024}$	$\mathbf{p = 0.045}$

new PIN-entry methods, the participant was instructed to conduct the following procedure. First, a detailed explanation of the method was given to the participant, and s/he repeated 20 PIN-entry sessions. The first ten sessions were for training, and only the last ten sessions were logged for analysis. A session was considered a success if the participant passed the PIN-entry test within three trials, as in a typical ATM. The timing data for each successful trial were stored for statistical analysis. If s/he failed to enter the PIN in three trials, that session was logged as a failure. In that case, we reminded the participant of his/her PIN, and helped him/her to redo the task until s/he entered a correct PIN to complete that PIN-entry session. After completing the 20 sessions including training sessions, the participant proceeded to the next method. To counterbalance learning effects, we generated $3! = 6$ permutations among the three methods and applied the same number of instances of each permutation to participants. As a result, the number of participants should be a multiple of six. After completing the three tests, each participant was asked to answer a final questionnaire. The questionnaire evaluated the user's overall preference for each method in terms of usability. The participant was asked to answer the question, "How convenient is this method?" by choosing a score from a 5-point Likert scale for each method.

5.2. Participants. We recruited 18 volunteers for the study from our local university and general public via a bulletin board. Therefore, each permutation was applied to three participants. As a result, we logged 540 ($= 18 \times 3 \times 10$) sessions in total. The average age of the participants was 28.6 years. Six participants were female and twelve were male. While we were collecting demographic data, we asked the participants how much time per day they usually spent using smartphones. Four of the participants answered that it was under two hours, most participants (eight) selected "between two and five hours," and three and two selected "between five and eight hours" and "between eight and twelve hours," respectively. One participant answered that she used a smartphone more than twelve hours a day. A ten-dollar coffee coupon was given to each participant as an incentive.

5.3. Analysis of Results. Table 1 shows the results of analysis of the PIN-entry time (i.e., task completion time), the number of trials needed to enter a PIN, and the usability score that the participants gave. Figure 9 shows the frequencies of each Likert score for usability, where 5, 4, 3, 2, and 1 represent "convenient," "slightly convenient," "neutral," "slightly inconvenient," and "inconvenient," respectively. For example, the red bar for Addlock with height 6 represents that the number of participants who answered that Addlock was slightly convenient was six, and the green bar represents that six people gave neutral answers.

As for the error rate, if the number of trials for a session is greater than three, the session is recorded as a failure. Only three out of 540 sessions failed (one failure for each method). Therefore, the error rate was 0.56% for each method. This implies that the three countermeasures are not significantly challenging to users.

To determine whether there were significant mean differences between the three methods, ANOVA and t-tests were conducted. The analysis results, shown in Table 2, reveal that Addlock is significantly more efficient than Map lock and Map lock is significantly more efficient than Counter Phone Lock in terms of PIN-entry time. Regarding the required number of trials per session, Addlock is marginally better than the other two methods. In summary, Addlock is the best choice among the three proposed methods in terms of PIN-entry time and number of retrials. This conclusion coincides with the usability scores that the participants gave. However, it is notable that participants gave similar scores to Counter Phone Lock and Map lock even though the PIN-entry time of Map lock was significantly faster than that of Counter Phone

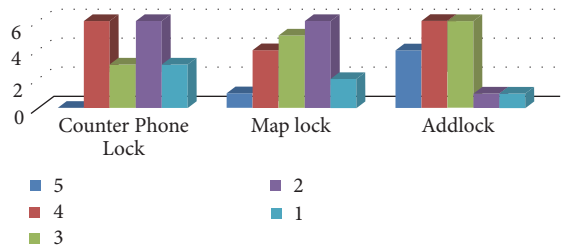
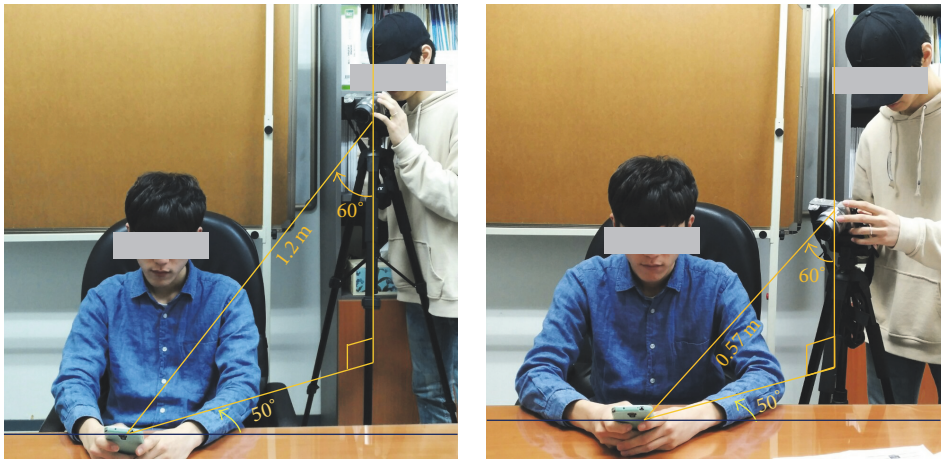


Figure 9: Likert scale data for usability score (frequencies of scores).



(a) First attack experiment (b) Second attack experiment

Figure 10: Setup for attack experiments.

Lock. It seems that the multimodal characteristic of Map lock made this method less comfortable for the participants and canceled out its advantage of faster PIN-entry. This result supports the claim in [19] that unimodal methods are more usable than multimodal ones for haptic cue-based PIN-entry. However, we remark that multimodal methods may be more effective for audio cue-based PIN-entry [18].

6. Security Analysis of Proposed Countermeasures

To verify the security of the proposed countermeasures, we designed an attack experiment similar to that described in Section 3.2. For the attack experiment, three volunteers were recruited as user victims. They were 26, 26, and 22 years old, respectively, and one of them was female. They were all engineering students of a local university and smartphone users. Each participant was assigned a random 4-digit PIN. For each of the three proposed methods, the participant was instructed to follow the following procedure. After a sufficient number of training sessions, s/he repeated PIN-entry sessions until 20 successful PIN-entries were collected for the same PIN. The above procedure was recorded under the same setting as that of the attack explained in Section 3.2. Figure 10(a) shows the setup for the attack experiment. The experiment was conducted in a quiet laboratory whose noise level was between 10 and 22 dB (13 dB on average), where the maximum value of noise was measured when the camera was beginning its operation. After completing the 20 successful sessions, the participant proceeded to the next method. The length of the video materials obtained by recording each participant was approximately 7 to 9 min, 10 to 12 min, and 7 to 8 min for Addlock, Counter Phone Lock, and Map lock, respectively. A ten-dollar coffee coupon was given to each participant as an incentive.

An analysis of the recorded sessions was performed by an attacker who well understood the working mechanisms of the three proposed methods. Because it was impossible to distinguish PIN digits using their entry time, the attacker tried to recognize the presence of vibrations from the recorded sound. The replay of video material including sound information was done on a personal computer with an Intel i7 CPU (3.6 GHz) and 8 GB RAM. The video material was played on a 24-inch monitor with the 1920 × 1080 (full HD) resolution. The attacker used earphones to hear the sound. Apparently, if this acoustic attack was successful, the same attack could be applied to all vibration-based PIN-entry methods including Timelock. However, the attacker could not extract any meaningful information for even a single digit of a PIN. That is, she failed to hear any sound of vibration.

To confirm the security of vibration-based approaches including ours, we conducted another attack experiment whose setting is shown in Figure 10(b). This setting is rather extreme in that the smartphone and the camera were very

close. Because the camera was located in parallel with the shoulder of the victim user, the user would recognize that he was being recorded. However, even with the same amount of recorded material under this setup, the attacker could not hear any vibration sound and thus could not recover any PIN information.

We do not claim that the above experiments show that the proposed methods are perfectly secure against acoustic attacks. If a high-end device such as a directed microphone is used for the attack together with advanced signal processing techniques, the presence of vibrations might be captured. However, taking into account the fact that our experiments were done in a noise-controlled environment, the above experiments provide a good clue that a casual attacker cannot recover a PIN easily in a normal noisy situation.

Next, we evaluate the security of the proposed countermeasures against the statistical attacks proposed in [23–25]. As explained in Section 2, the attacks in [23–25] are only applicable to the cases where a challenge is constructed depending on the secret, and the secret objects are directly visible in a challenge to an attacker. The proposed countermeasures are not k-out-of-n schemes; all challenges are random and independent of a secret PIN, causing no bias in the distribution. Moreover, in Addlock and Counter Phone Lock, there is no visible challenge; only responses are visible. Therefore, the statistical attacks in [23–25] are not applicable to the proposed countermeasures.

Finally, we evaluate the security of the proposed countermeasures against the timing attacks reported in [21, 22] that use the variations in a user's cognitive load to compute a response. The design philosophy of the proposed countermeasures is to force the duration of vibration to be independent of the value of a PIN digit to remove the correlation between a PIN digit and its entry time. However, the PIN-entry time may not be completely independent of a PIN digit due to potential variations in a user's cognitive load. For example, in Counter Phone Lock, assume that a user recognizes two vibrations after tapping a sector. If the target number is 3, the target sector is right next to the tapped one, which is easy to find. On the other hand, if the target number is 7, the user has to mentally locate the fifth sector from the tapped one, which will require nonnegligible time. It is clear that this variation does not harm the security of Counter Phone Lock, because no useful information is leaked due to this time variation. Note that an attacker does not see any number in either a challenge or the corresponding response. However, for Addlock, partial information about the PIN may be revealed if there is a detectable variation in a user's PIN-entry time. For example, assume that a user recognizes three vibration cues while the progress bar is growing in Figure 6(a). If the target number is 1, the user will easily compute the response, $3 + 1 = 4$. On the other hand, if the target number is 9, the user has to compute secretly $3 + 9 \bmod 10 = 2$. This may require slightly more time. This potential variation in the response time was the motivation of the timing attacks reported in [21, 22]. Although it is not clear if an actual timing attack to Addlock is possible or not, we would like to remark that this potential attack to Addlock can be easily prevented if we take care of the order of mental operations. That is, a user may first recall the target PIN, say 9, before vibration cues are activated, that is, before the progress bar reaches the small white triangle marker in Figure 6(a). While the cues are being activated, the user performs "increment" operations on the fly from the target number 9, obtaining $0, 1, 2, \ldots$ in turn. When the vibration ends, the user obtains a response without further computation, making the response time independent of a target PIN digit. However, we have to scrutinize how this will affect the usability of Addlock. We leave this issue for our future research. Finally, it is easy to see that there is no correlation between a PIN digit and its response time in Map lock.

7. Discussion on Applications and Limitations

In this section, we discuss the typical applications and limitations of the proposed PIN-entry methods. In general, the PIN-entry tasks of the randomized and secondary channel-based methods are less convenient than regular PIN-entry. As the experimental results in the previous section show, the PIN-entry task using our methods requires significantly more time than the same task using the legacy PIN pad, which is less than 2 s [8, 18]. Therefore, the proposed methods cannot replace the legacy PIN pad in all applications but can be a good alternative for a security-critical task. Users may also use the legacy PIN pad and the new methods selectively according to context. For example, the user may use the new methods to enter a PIN code associated with a bank account for a financial transaction in a public place such as a lounge. When s/he is in a private place, for example, at home, a more intuitive but insecure method such as regular PIN-entry can be used. Because the PIN space for the new methods is compatible with the existing ones, the new methods will not cause any significant change in the existing infrastructure, which is not the case for the methods with incompatible PIN spaces such as [4, 19]. It is also possible that a specific PIN-entry method for authentication could be selected automatically with the help of context-aware technologies [29]. The compatibility with the legacy PIN has another merit for memorability. Because a user may use his/her PIN without any change, s/he does not need to memorize a new PIN for the new PIN-entry system.

The limit of vibration-based methods including ours is that they require a secure channel for the transmission of vibration cues. For example, these methods cannot be directly used for current ATMs, where a vibration interface is not available. In this case, PIN-entry methods with audio cues might be more appropriate because many of the recently deployed ATMs are equipped with an audio jack [18]. We also note that the proposed methods can be easily modified to audio versions by changing the vibration cues to beeps, as in [19]. Nevertheless, such audio versions will not be able to outperform the previous methods customized for audio interfaces such as [18] because these customized methods fully utilize the higher bandwidth of an audio channel. For example, instead of plain beeps, they can easily transmit more information-intensive audio cues such as an alphabet letter "A," which would be significantly more complex for

Table 3: Comparison of usability and security of various PIN-entry methods (TCT (task completion time): PIN-entry time, Error rate: rate of erroneous input in three trials, Comp.: compatibility with legacy 4-digit PIN, P_{RG}: success probability of attacker's log-on (in three trials) by randomly guessing a PIN, and P_O: success probability of attacker's log-on (in three trials) after observing one PIN-entry session).

Method	TCT (s)	Error rate	Comp.	P_{RG}	P_O
Regular	1.4*	NA	◯	0.0003	≈1.0
Undercover [12]	32–45	>0.315	×	0.00015	≪0.00015**
Vibrapass [14]	3.9–8.2	>0.148	◯	0.0003	0.04–0.6
Haptic Wheel [15]	23.0–23.5	0.16–0.18	×	<0.0002	<0.0002
Spinlock [19]	13.9–20.1	0.07–0.08 (0.62–0.68)†	×	≤0.0003	NA
Colorlock [19]	10.0–10.1	0.05–0.09 (0.14–0.18)†	×	0.0002	0.0768‡
Timelock [19]	8.0–10.8	0.02–0.04 (0.02–0.04)†	×	0.0002	0.0768
TictocPIN [28]	15.8	0.0	◯	0.0003	0.0048
Addlock	18.0	0.0056	◯	0.0003	0.0003
Map lock	20.6	0.0056	◯	0.0003	0.0003
Counter Phone Lock	27.6	0.0056	◯	0.0003	0.0003

*Data from [8], **0.00015 is the claimed value in [12], which is significantly smaller in practice. †Values in parentheses represent reset rates (rates of canceled trials). ‡Expected value assuming that the randomization parameters for Colorlock and Timelock are the same.

haptic channels to convey. However, audio channel-based methods cannot be used when a secure audio interface such as a pair of earphones is not available. Even if earphones are available, wearing earphones for authentication might be inconvenient or take significant time. Therefore, if a device can only generate vibration cues, vibration-based methods will be a more convenient solution in many cases. The typical platforms for vibration-based methods will be smartphones, smart pads, and wearable devices such as smart watches.

8. Comparison with Previous Works

In this section, we compare the usability and security of the proposed methods with those of the previous methods, in particular, previous haptic approaches. Table 3 summarizes the features of the previous haptic methods as well as our proposal. For reference, we also included the data for the regular PIN-entry method.

Of the three counting-based variants in [19], Colorlock adopts an input mechanism similar to that of Timelock. In other words, it has four buttons, and a user inputs a PIN digit by touching a button and keeping the finger on the button until the number of delivered vibration buzzes reaches a target number. Another variant, Spinlock, shows a dial interface and a user inputs a PIN digit by rotating the dial until the number of delivered vibration buzzes equals the digit. Although some randomization techniques are adopted to reduce the correlation of the finger contact duration or the angle of rotation with a PIN digit, it is anticipated that they will be vulnerable to the correlation attack that we presented in Section 3.2. Regarding the TCT, the PIN-entry times of Colorlock and Timelock are significantly smaller than that of Addlock. However, we remark that this speedup has been obtained by reducing the PIN-digit space to $\{1, 2, 3, 4, 5\}$, which seriously sacrificed the resistance to observation attacks and the compatibility with regular PINs as explained in Section 3.1. If the PIN-digit space is redefined as $\{0, 1, 2, \ldots, 9\}$ for compatibility, the PIN-entry time will be almost doubled. It should be noted that using this standard digit set only prevents the leakage of button order in Colorlock and Timelock, but it does not prevent our correlation attack presented in Section 3.2. (See Appendix.) Another minor problem in the case of Colorlock is its reset rate. The timing data for Colorlock (as well as Spinlock and Timelock) include only those for successful sessions but do not include those for canceled sessions. However, because the reset rate is nonnegligible, it is fair to amortize the time required for resets to the whole PIN-entry time. On the other hand, our timing data already include the time for reset and correction.

In addition to the counting-based methods we considered in this paper, there has been extensive research on haptic channel-based PIN-entry methods in the literature. Undercover [12] transmits a tactile cue to a user through a customized trackball interface. By combining this information and a separate visual challenge, the user is asked to input a response. While Undercover provides a novel integration of visual and tactile challenges, it has a security issue because a significant amount of statistical information about a PIN may be leaked to an attacker via the visual challenges [23]. In addition, its PIN-entry time is very long and its error rate is too high to be used in practice. Although the fast PIN-entry time of Vibrapass [14] and its compatibility are very attractive, it is too vulnerable to be used as a countermeasure to observation attacks. An attacker of Vibrapass may obtain information about the PIN by observing only a user's responses, because a response always contains the correct PIN as a subsequence. As a result, an attacker may pass the PIN-entry test with a probability of up to 60% by observing only a single PIN-entry session. The above attacks against Undercover and Vibrapass do not require the attacker to have access to the secret haptic channel. Haptic Wheel [15] can be regarded as a preliminary version of Phone Lock [16], but it requires a customized haptic device and shows a relatively

high error rate. Recently, a novel improvement on the binary method [2] using vibration was proposed in [28], where a vibration cue is used as a hidden indicator, similar to our third countermeasure. However, it does not aim at providing full recording resilience. For example, after observing a PIN-entry session, a recording attacker can narrow the search space for a PIN down to 625 instead of 10,000. In summary, the proposed method, in particular Addlock, is slightly slower than recent regular-PIN-compatible vibration-based methods such as TictocPIN [28] but provides higher security. Therefore, Addlock is a good alternative for a security-critical task. However, if a user may be satisfied with a moderate level of security, a faster solution might be preferred.

There are also many authentication methods that use secure channels other than haptic or audio channels. For example, Thorpe et al. proposed a brain-computer interface-based method [30], where a user's brain signals are transmitted for authentication. In [31], a user's input is recognized by an eye tracking device. In [32, 33], random challenges are safely delivered using the differences of three-dimensional (3D) depth to a user through a 3D visual channel. This difference can be recognized only at a specific spot in front of the glasses-free 3D display, where the user's eyes are located. Therefore, the 3D challenges are unobservable by an attacker. However, such methods are only applicable to a device equipped with a three-dimensional display. Another kind of secure visual channel can be realized if an additional wearable computing device with a private display, for example, Google Glass, is available [34, 35]. Finally, there are visual obfuscation methods using hand-shielding effects [36, 37] and fake cursors [38]. However, it has not been proven or quantified how much security such obfuscation methods guarantee.

9. Conclusion

In this paper, we analyzed the security of the counting-based PIN-entry proposed by Bianchi et al. [19], focusing on the vibration version of Timelock in particular. Both a probabilistic analysis and real attack experiment revealed that the security level guaranteed by Timelock is lower than that claimed. As countermeasures to this problem, we proposed three PIN-entry methods and performed a study to analyze the usability and security of these methods. According to our analysis, Addlock was the most efficient solution, confirming that the advantage of a unimodal system as conjectured by Bianchi et al. is still valid.

Appendix

In this section, we provide the details of the attack experiment which was briefly explained in Section 3.2. We begin by showing a section of the analysis sheet submitted by the attacker. Table 4 is a section in the analysis sheet for a single PIN digit. Because there are twelve PIN digits (three 4-digit PINs) in total, the attacker had eleven more sections of similar form. According to the table, the time to enter the PIN digit was approximately 1.0 s in the first session. The probabilities that this PIN digit was "1," "2," and "3," were estimated to be 0.36, 0.36, and 0.28, respectively, according to the probability distribution given in Figure 4. The attacker's strategy was then to choose the candidate with the maximum probability. Because in this case the probability was the same for "1" and "2," she randomly selected one of the two candidates. Therefore, if the correct PIN digit was actually either "1" or "2," her success rate would be 0.5. After the second session was observed, the averages of P_i over the first two sessions were computed. For example, the average of P_1 was $(0.36 + 0.36)/2 = 0.36$. The same held for P_2. Therefore, up to the second session, "1" and "2" were the equally probable candidates. However, when she observed that the entry time for the third session was 0.3 s, the average of P_1 was computed as $(0.36 + 0.36 + 1.00)/3 \approx 0.57$, and the average of P_2 through P_5 was 0.24, 0.19, 0, and 0, respectively. Because the maximum among these five values was 0.57, she concluded that the PIN digit was "1." This procedure (computing the average of P_i over sessions and choosing the candidate with maximum average) was repeated up to session 20. The eighth column in the table shows the maximum average probability up to the corresponding session and the ninth column shows the candidate(s) that had that maximum value. The digits in the parentheses are the candidates that coincided with the measured time for that session but did not have the maximum probability. The final column shows the success rate of the attack using the above strategy. (The actual PIN digit was "1.") We remark that the above attack could be refined further by pruning the candidates with probability 0. That is, the attacker might exclude "4" and "5" after the first session because $P_4 = P_5 = 0$ and also exclude "2" and "3" after the third session.

Now, we analyze the overall performance of the attacker for the whole twelve digits. Note that the sheet shown in Table 4 can be constructed independently for each of the twelve PIN digits. To evaluate the attack performance, we may consider two distinct metrics. The first one is the digit-wise average success rate, which we define as the average of the success rates for the four digits. For example, if the attacker could recover the four digits in a PIN with the probability of 0.5, 0, 0.4, 0.7, respectively, the digit-wise average success rate is computed as $(0.5+0+0.4+0.7)/4 = 0.4$. On the other hand, another metric is the success rate for a whole PIN, which we define as the probability that the attacker successfully recovers a whole PIN. In the above example, this rate is $0.5 \times 0 \times 0.4 \times 0.7 = 0$. Figure 11 shows the change of these two kinds of success rates according to the number of recorded sessions used for the attack. The blue line stands for the digit-wise average success rate, that is, average among the twelve maximum probabilities. This was shown in Figure 5. The red line is the success rate for a whole PIN averaged over three 4-digit PINs. This figure shows that an attacker may obtain a significant amount of information by observing only a few PIN-entry sessions. Note that the expected success rate of a single-digit guessing is 0.2 and that of a whole PIN is 0.0016, which are significantly lower than the values in the figure.

In Section 3.1, we pointed out the PIN space reduction issue of Timelock which resulted from limiting a PIN digit to $\{1, \ldots, 5\}$. To resolve this issue, we designed a modified version of Timelock where a PIN digit is selected from the traditional set $\{0, 1, \ldots, 9\}$, instead of $\{1, \ldots, 5\}$. A PIN numeral "0" is entered by counting ten buzzes. To see whether

Table 4: Deduction of a PIN digit using the measured contact duration (P_i = Pr (PIN numeral = i); $A(P_i)$: average of P_i over sessions).

Session	Time (s)	P_1	P_2	P_3	P_4	P_5	max($A(P_i)$)	PIN candidate(s)	Success rate
1	1.0	0.36	0.36	0.28	0.00	0.00	0.36	1, 2 (, 3)	0.5
2	1.0	0.36	0.36	0.28	0.00	0.00	0.36	1, 2 (, 3)	0.5
3	0.3	1.00	0.00	0.00	0.00	0.00	0.57	1	1.0
4	0.3	1.00	0.00	0.00	0.00	0.00	0.68	1	1.0
5	0.8	0.47	0.47	0.06	0.00	0.00	0.63	1 (, 2, 3)	1.0
6	0.8	0.47	0.47	0.06	0.00	0.00	0.61	1 (, 2, 3)	1.0
7	0.7	0.51	0.49	0.00	0.00	0.00	0.60	1 (, 2)	1.0
8	1.1	0.33	0.33	0.31	0.02	0.00	0.56	1 (, 2, 3, 4)	1.0
9	0.8	0.47	0.47	0.06	0.00	0.00	0.55	1 (, 2, 3)	1.0
10	0.4	0.95	0.05	0.00	0.00	0.00	0.59	1 (, 2)	1.0
11	1.6	0.18	0.25	0.25	0.25	0.08	0.55	1 (, 2, 3, 4, 5)	1.0
12	0.9	0.41	0.41	0.18	0.00	0.00	0.54	1 (, 2, 3)	1.0
13	1.4	0.26	0.26	0.26	0.21	0.01	0.52	1 (, 2, 3, 4, 5)	1.0
14	1.6	0.18	0.25	0.25	0.25	0.08	0.50	1 (, 2, 3, 4, 5)	1.0
15	1.4	0.26	0.26	0.26	0.21	0.01	0.48	1 (, 2, 3, 4, 5)	1.0
16	0.6	0.59	0.41	0.00	0.00	0.00	0.49	1 (, 2)	1.0
17	0.9	0.41	0.41	0.18	0.00	0.00	0.48	1 (, 2, 3)	1.0
18	1.6	0.18	0.25	0.25	0.25	0.08	0.47	1 (, 2, 3, 4, 5)	1.0
19	0.7	0.51	0.49	0.00	0.00	0.00	0.47	1 (, 2)	1.0
20	0.5	0.74	0.26	0.00	0.00	0.00	0.48	1 (, 2)	1.0

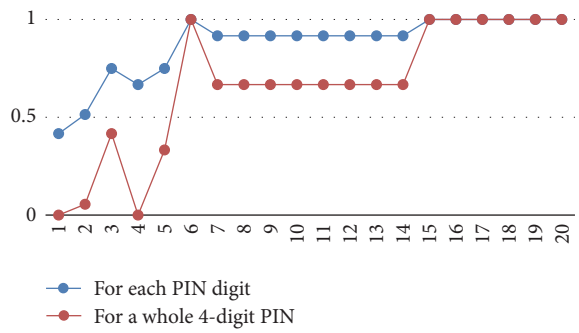


Figure 11: Success rate of observation attacks according to the number of recorded sessions.

this change may also prevent an observation attack effectively, we computed a merged distribution of the finger contact duration t for this modification and obtained the probability density function shown in Figure 12. It is easy to see that there are also biases in this distribution similar to those in the original Timelock. For example, the probability that $600 \leq t \leq 1{,}000$ ms is $\int_{600}^{1{,}000} f(t)dt \approx 0.058$, and there are only three possible numerals that are consistent with this range of t. Their conditional probabilities are Pr (PIN numeral = "1" | $600 \leq t \leq 1{,}000$) ≈ 0.459, Pr (PIN numeral = "2" | $600 \leq t \leq 1{,}000$) ≈ 0.438, and Pr (PIN numeral = "3" | $600 \leq t \leq 1{,}000$) ≈ 0.103. Therefore, the attacker will be able to figure out the correct PIN digit with probability 0.459 for this range. The average success probability of an observation attack on a single PIN digit for the total range of time is approximately 0.325, which is slightly smaller than that of the original Timelock, but still quite high.

Let us consider other variants of Timelock. Using another modality, that is, audio instead of vibration, changes the probability distributions shown in Figures 4 and 12. In addition, the use of a random beat mode may slightly reduce the correlation between a PIN digit and its entry time but significantly increases the PIN-entry time [19]. We could also increase the ranges of intercue intervals and initial pauses, but these changes will also increase the PIN-entry time and reduce its usability. Therefore, none of the above variants can be an essential solution to prevent an observation attack.

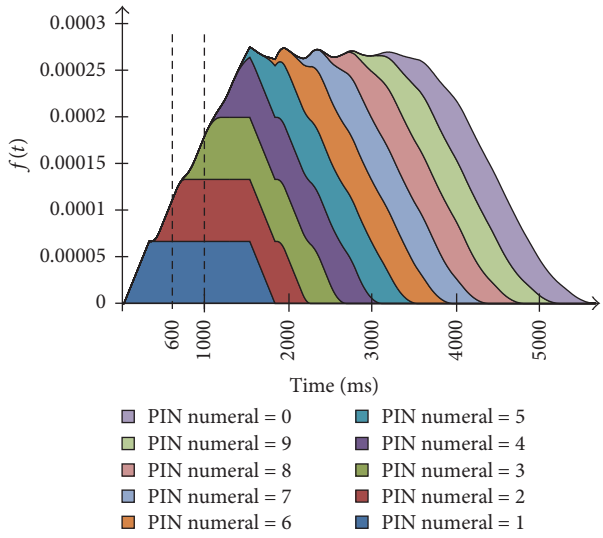


Figure 12: Accumulated distribution of finger contact duration for modified Timelock, where the PIN digit is selected from {0, 1, . . . , 9}.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant no. 2017R1D1A1A09000915) and in part by the MSIT, Korea, under the ITRC Support Program (IITP-2017-2012-0-00646) supervised by the IITP. The authors would like to thank Jong-Hyuk Im, Hee-Yong Kwon, Won-Il Pyo, Yejin Yoon, and Yebyoul Son for their help in the analysis, and all voluntary participants of our experiments.

References

[1] C. S. Weir, G. Douglas, T. Richardson, and M. Jack, "Usable security: User preferences for authentication methods in eBanking and the effects of experience," *Interacting with Computers*, vol. 22, no. 3, pp. 153–164, 2010.

[2] 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 '04)*, pp. 236–245, Washington, DC, USA, October 2004.

[3] D. S. Tan, P. Keyani, and M. Czerwinski, "Spy-resistant keyboard: more secure password entry on public touch screen displays," in *Proceedings of the 17th Australia Conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future (OZCHI '05)*, ACM, Canberra, Australia, November 2005.

[4] A. De Luca, K. Hertzschuch, and H. Hussmann, "ColorPIN - securing PIN entry through indirect input," in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems (CHI '10)*, pp. 1103–1106, Atlanta, Ga, USA, April 2010.

[5] M.-K. Lee and H. Nam, "Secure and usable PIN-entry method with shoulder-surfing resistance," in *HCI International 2013*, vol. 374 of *CCIS*, pp. 745–748, 2013.

[6] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies (WOOT '10)*, Berkeley, CA, USA, 2010.

[7] E. von Zezschwitz, A. Koslow, A. De Luca, and H. Hussmann, "Making graphic-based authentication secure against smudge attacks," in *Proceedings of the 18th International Conference on Intelligent User Interfaces (IUI '13)*, pp. 277–286, ACM, Santa Monica, Calif, USA, March 2013.

[8] M.-K. Lee, "Security notions and advanced method for human shoulder-surfing resistant PIN-entry," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 695–708, 2014.

[9] T. Kwon, S. Shin, and S. Na, "Covert attentional shoulder surfing: Human adversaries are more powerful than expected," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 716–727, 2014.

[10] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, "iSpy: Automatic reconstruction of typed input from compromising reflections," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS '11)*, pp. 527–536, Chicago, Ill, USA, October 2011.

[11] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, "A fast eavesdropping attack against touchscreens," in *Proceedings of the 2011 7th International Conference on Information Assurance and Security, IAS 2011*, pp. 320–325, Melaka, Malaysia, December 2011.

[12] H. Sasamoto, N. Christin, and E. Hayashi, "Undercover: authentication usable in front of prying eyes," in *Proceedings of the 26th Annual CHI Conference on Human Factors in Computing Systems (CHI '08)*, pp. 183–192, ACM, New York, NY, USA, April 2008.

[13] T. Perković, M. Čagalj, and N. Rakić, "SSSL: shoulder surfing safe login," in *Proceedings of the International Conference on Software, Telecommunication and Computer Networks 2009*, pp. 270–275, 2009.

[14] A. De Luca, E. Von Zezschwitz, and H. Hußmann, "Vibrapass - secure authentication based on shared lies," in *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems (CHI '2009)*, pp. 913–916, New York, NY, USA, April 2009.

[15] A. Bianchi, I. Oakley, J. K. Lee, and D.-S. Kwon, "The haptic wheel: design and evaluation of a tactile password system," in *Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI '10)*, pp. 3625–3630, ACM, Atlanta, Ga, USA, April 2010.

[16] 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 (TEI '11)*, pp. 197–200, ACM, Funchal, Portugal, 2011.

[17] A. Bianchi, I. Oakley, and D. S. Kwon, "Spinlock: A single-cue haptic and audio PIN input technique for authentication," in *Haptic and Audio Interaction Design*, E. W. Cooper, V. V. Kryssanov, H. Ogawa, and S. Brewster, Eds., vol. 6851 of *Lecture Notes in Computer Science*, pp. 81–90, Springer, Berlin, Heidelberg, 2011.

[18] M.-K. Lee, H. Nam, and D. K. Kim, "Secure bimodal PIN-entry method using audio signals," *Computers and Security*, vol. 56, pp. 140–150, 2016.

[19] A. Bianchi, I. Oakley, and D. S. Kwon, "Counting clicks and beeps: Exploring numerosity based haptic and audio PIN entry," *Interacting with Computers*, vol. 24, no. 5, pp. 409–422, 2012.

[20] C. Spence, M. E. R. Nicholls, and J. Driver, "The cost of expecting events in the wrong sensory modality," *Perception and Psychophysics*, vol. 63, no. 2, pp. 330–336, 2001.

[21] T. Perković, S. Li, A. Mumtaz, S. A. Khayam, Y. Javed, and M. Čagalj, "Breaking undercover: Exploiting design flaws and nonuniform human behavior," in *Proceedings of the 7th Symposium on Usable Privacy and Security (SOUPS '2011)*, Pittsburgh, Pennsylvania, July 2011.

[22] M. Cagalj, T. Perkovic, and M. Bugaric, "Timing attacks on cognitive authentication schemes," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 584–596, 2015.

[23] 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 in 19th Annual Network and Distributed System Security Symposium (NDSS, 2012)*, 2012.

[24] H. J. Asghar, S. Li, R. Steinfeld, and J. Pieprzyk, "Does counting still count? revisiting the security of counting based user authentication protocols against statistical attacks," in *Proceedings of the 20th Annual Network and Distributed System Security Symposium, (NDSS 2013)*, 2013.

[25] H. J. Asghar, R. Steinfeld, S. Li, M. A. Kaafar, and J. Pieprzyk, "On the linearization of human identification protocols: attacks based on linear algebra, coding theory, and lattices," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1643–1655, 2015.

[26] S. Li and H. Shum, "Secure human-computer identification (interface) systems against peeping attacks: SecHCI," *IACR Cryptology ePrint Archive*, Article ID 268, 2005.

[27] M.-K. Lee, User authentication method with parameterized security and usability, February 2014, Korea Patent 10-1368518.

[28] T. Kwon and J. Hong, "Analysis and improvement of a PIN-Entry method resilient to shoulder-surfing and recording attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, article no. A6, pp. 278–292, 2015.

[29] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley, "CASA: context-aware scalable authentication," in *Proceedings of the 9th Symposium on Usable Privacy and Security (SOUPS '13)*, pp. 3:1–3:10, ACM, Newcastle, UK, July 2013.

[30] J. Thorpe, P. C. van Oorschot, and A. Somayaji, "Pass-thoughts: authenticating with our minds," in *Proceedings of the 2005 workshop on New security paradigms (NSPW '05)*, pp. 45–56, Lake Arrowhead, Calif, USA, September 2005.

[31] 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 (SOUPS '07)*, pp. 13–19, ACM, Pittsburgh, Pa, USA, July 2007.

[32] M.-K. Lee and H. Nam, "Secure and fast PIN-entry method for 3D display," SECURWARE 2013, pp. 26–29, 2013.

[33] M.-K. Lee, J. B. Kim, and M. K. Franklin, "Enhancing the security of personal identification numbers with three-dimensional displays," *Mobile Information Systems*, vol. 2016, Article ID 8019830, 9 pages, 2016.

[34] D. K. Yadav, B. Ionascu, S. V. Krishna Ongole, A. Roy, and N. Memon, "Design and analysis of shoulder surfing resistant PIN based authentication mechanisms on Google Glass," in 1st Workshop on Wearable Security and Privacy (In Association with Financial Crypto 2015), 2015, paper 8.

[35] P. Lantz, B. Johansson, M. Hell, and B. Smeets, "Visual cryptography and obfuscation: a use-case for decrypting and deobfuscating information using augmented reality," in *Financial cryptography and data security*, vol. 8976 of *Lecture Notes in Computer Science*, pp. 261–273, Springer, Berlin, Heidelberg, 2015.

[36] D. Kim, P. Dunphy, P. Briggs et al., "Multi-touch authentication on tabletops," in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems (CHI '2010)*, pp. 1093–1102, Atlanta, Georgia, USA, April 2010.

[37] Q. Yan, J. Han, Y. Li, J. Zhou, and R. H. Deng, "Designing leakage-resilient password entry on touchscreen mobile devices," in *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS '13)*, pp. 37–48, ACM, Hangzhou, China, May 2013.

[38] A. De Luca, E. Von Zezschwitz, L. Pichler, and H. Hussmann, "Using fake cursors to secure on-screen password entry," in *Proceedings of the 31st Annual CHI Conference on Human Factors in Computing Systems: Changing Perspectives (CHI '2013)*, pp. 2399–2402, Paris, France, May 2013.