You Cannot Sense My PINs: A Side-Channel Attack Deterrent Solution Based on Haptic Feedback on Touch-enabled Devices

Caijin Ling
School of EIE
Heyuan Polytechnic
Heyuan, Guangdong 517000, China
Email: ling8983@gmail.com

Xiali Hei, Kam Kong and Michael Peays
Dept. of CIS
Delaware State University
Dover, DE 19901, USA
Email: {xhei, kkong}@desu.edu,
mpeays15@students.desu.edu

Mohsen Guizani
Dept. of ECE
University of Idaho
Moscow, ID 83844, USA
Email: mguizani@ieee.org

***Abstract*—In this paper, we introduce a novel and secure solution to mitigate side-channel attacks to capture the PINs like touchID and other credentials of touch-enabled devices. Our approach can protect haptic feedback enabled devices from potential direct observation techniques such as cameras and motion sense techniques including such as accelerometers in smart-watch. Both attacks use the concept of shoulder surfing in social engineering and were published recently (CCS'14 and CCS'15). Hand-held devices universally employ small vibration motors as an inexpensive way to provide haptic feedback. The strength of the haptic feedback depends on the brand and the device manufacturer. They are usually strong enough to produce sliding movement and make audible noises if the device is resting on the top of a desk when the vibration motor turns. However, when the device is held in the hand the vibration can only be sensed by the holder; it is usually impossible or uncertain for an observer to know when the vibration motor turns. Our proposed solution uses the haptic feedback to inform the internal state of the keypad to the user and takes advantage of the fact that the effect of haptic feedback can be easily cloaked in such a way that direct observation techniques and indirect sensing techniques will fail. We develop an application on Android cell phones to demonstrate it and invite users to test the code. Moreover, we use real smart-watch to sense the vibration of Android cell phones. Our experimental results show that our approach can mitigate the probability of sensing a 4-digit or 6-digit PINs using smart-watch to below practical value. Our approach also can mitigate the probability of recognizing a 4-digit or 6-digit PINs using a camera within 1 meter to below practical value because the user does not need to move his or her hand during the internal states to input different PINs.**

***Index Terms*—haptic feedback; random keypad generator; vibration sensor; touch-enabled devices; security**

I. Introduction

Touch-enabled devices including mobile devices are ubiquitously utilized in our daily life. However, they are also attracting attention from attackers. Information leakage caused by touched keys from attackers has been a topic of concern for a long time. Currently, almost all systems involve a PIN or password based identity access control before a user can access requested resources. Therefore, protecting the credential during user's typing has become the core of any secure system. Whether the researcher can defend against pattern recognition based side channel attacks or not become a crucial problem.

A typical authentication process on today's handheld devices involves the entering of passwords using a keypad on a touch screen. To enter the password, the user needs to know what keys she needs to press. This almost always means the user would find the location of the keys on the screen and physically position her finger over the targeted key and press down the finger. Thus, any person or camera happens to have a view of the keypad, the finger, and the shoulder movement can potentially see or record the key strokes and thus know the password with high probability [1]. Indeed, attackers adopt various tools such video, smart-phone [2]–[4], smart-watch [1], sunglass, a teapot, and etc. to get user's passwords. It was reported that the user's shirt could be used as a tool by attackers to get passwords [5], [6].

We propose a solution based on "invisible" haptic feedback to prevent the finger movement detection and acoustic based detection by cameras and sensors. When a user presses a key on the haptic feedback-enabled touch screen, the device will produce a sequence of "invisible" haptic feedback to the user. The way the keypad communicate to the user employs its internal status of the keypad. The user would hold down the finger steady and count the number of haptic feedback silently in his or her head until he or she reaches the desired state. He or she would then release the finger, and correct key (which may or may not be the same as the actual key he or she pressed) is registered as input. When the user presses the key, the internal state is "neutral". If the user releases the key now, the keypad will register the exact key the user pressed. There is no need for the user to be concerned with the danger of shoulder surfing and other side-channel attacks. Otherwise, a different key may be registered as input. Thus, if the user holds down the key, then the internal state of the keypad will start changing at a random pace. Furthermore, the keypad will send a single pulse of haptic feedback to the user at every change of the internal state. The number of internal states is typically very small (3 or 4), and thus very easy for the user to keep track in her head. If the user waits long enough, the

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

internal state will be reset to neutral (indicated by a double pulse of haptic feedback), and the same sequence of internal state change will start all over again.

If the user releases the finger now, then the key registered by the keypad may be different from the actual key pressed. The registered key is a function of the pressed key and the internal state. The registered key matches the pressed key if and only if the internal state is neutral. The mapping of the pressed key and internal state to the registered key will be explained later.

Since the haptic feedback can only be sensed by the user, and only the user knows the internal states of the keypad and hence the registered input key. Moreover, the user will not move his or her finger during the change of internal state, through cameras and sensors on smart-watch, an attacker cannot be sure the actual internal state at the time of the release of the finger, and therefore will not be able to know for certain what key the user intended to press. Thus, the solution can defend against the attacks exploiting finger movement detection and acoustic based detection by cameras and sensors. These attacks are based on the knowledge of finger movement and keyboard layout. They include the causal shoulder surfing, sophisticated use of video recording and image analysis to track the movement of a finger, sensing the of carried smart-watch. Our experiment shows that our novel solution is effective against such attacks.

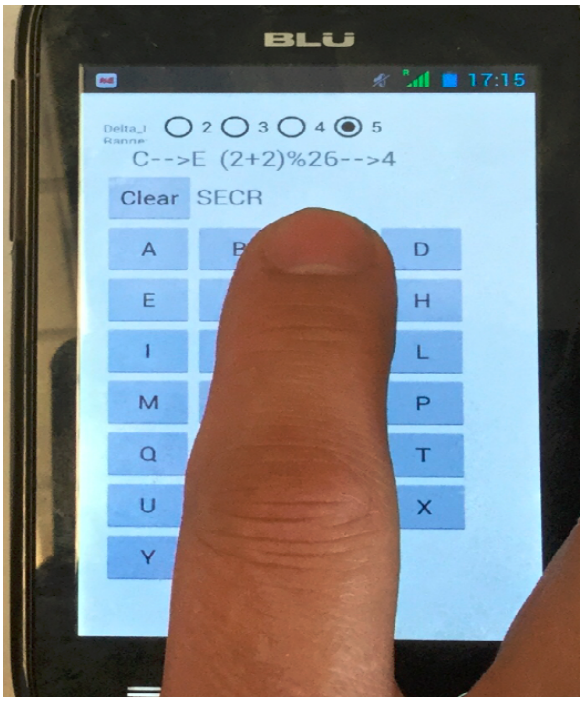

Fig. 1. Demo Application

Our contributions are summarized as follows:

- We present a novel solution to prevent the side-channel attacks to identify the PINs or passwords.
- To the best of our knowledge, we are the first group who proposed to a promising new non-visual interaction paradigm – haptic feedback based motor to solve the problem of security.
- We develop an Android application prototype and test it.

Our scheme is effective and significant: (1) We get nearly 100% success rate in our users test; (2) Our solution is light-weight and less expensive, in which only an already built-in motor in a smart-phone is required; (3) Our solution can be widely applied to protect the PINs and other credentials.

The remainder of this paper is organized as follows: In Section II we describe the background and attack models. We introduce our novel approach in Section III. In Section IV, we analyze and calculate the performance. We show our real experimental results and evaluate the solution in Section V. We present related discussions in Section VI. In Section VII, we discuss the related work, and we conclude the paper in Section VIII.

II. Threat Model

There are a lot of attacks based on touch-enabled devices. According to the mechanism of attacks, we can classify these attacks into two categories:

- Sensors based side-channel attacks. Sensors may be an accelerometer, a microphone, and others. Attacks usually use the sensors to detect a victim without his or her knowledge. For instance, a smart-watch is used to attack victim [7] when the smart-watch was controlled by the attacker. An accelerometer in the smart-watch can offer the exact coordinates of keys. An acoustic sensor can record the sound of user's PIN typing by a microphone as well.
- Vision based side-channel attacks. Some researchers employ footage to record the track of movement. Although attackers cannot see the specific keys, they can detect the key through shadow [8], or the light [9], or the coordinates [10], or other visible features of a key to calculate the real key. According to [9], the success rate of detection is over 97% per character.

  In this paper, we specifically focus on how to defend against above new attacks.

III. Our novel approach

The most innovative part of our solution is the hidden internal state s of the keypad. When the user presses a key `pressedKey`, the keypad registers `targetKey` as input. It is important to note that `pressedKey` may not be the same as `targetKey`. For example, the user may physically press the key `pressedKey=J` when he or she really want to input `targetKey=G`. Figure 1 illustrated our demo application. It shows a keypad and internal states. Above the keypad, the upper line is used for a user to see the internal states. The lower line is used for debugging. In real and usable application, the lower line will not be displayed.

The central theme of our solution is the fact that `targetKey` is a function of `pressedKey` and the internal state s:

`targetKey` = f(`pressedKey`, s).

In this paper, we propose the hidden internal state internal Δi. If we denote the index of a key by idx(`key`), then:

idx(`targetKey`) = idx((`pressedKey` + Δi) mod N), where N is the number of keys.

The range of Δi is $0, 1, 2, ..., L-1$; where L is the number of internal states. A larger value of L provides better security but may be cumbersome for the user. Our usability study and experimental results suggest that $L = 3$ or $L = 4$ are right choices. The interpretation of the number of haptic feedback is explained with an example as follows.

A. Communication of the hidden internal state

Assume that the following keys are aligned in a row: `Q`, `W`, `E`, `R`, `T`, `Y`, ... Each key is associated with an index. For the purpose of this example, we will assume that the indexes are $i = 0, 1, 2, 3, 4, 5, ...$, correspondingly. When the user presses and holds down his or her finger on the key `E`: the index is $i = 2$ and the internal counter is $\Delta i = 0$.

Please refer to Fig.2 3 as we examine the different scenarios.

- If he or she releases key immediately, $\Delta i = 0$. The keypad will register the key $i = 2 + \Delta i = 0$, hence `E`.
- If he or she continues to hold down the key for a duration of T_1, then the device will vibrate for a very brief moment T_2. The keypad now increments its internal counter: $\Delta i = 1$. If he or she releases the finger now, the keypad will register the key $i = 2 + \Delta i = 2 + 1 = 3$, hence `R`.
- On the other hand, if he or she continues to hold down the key for another duration of T_1, then the device will vibrate once for a very brief moment T_2 again. The keypad now increments its internal counter again: $\Delta i = 2$. If he or she releases the finger now, the keypad will register the key $i = 2 + \Delta i = 2 + 2 = 4$, hence `T`.
- On the other hand, if he or she continues to hold down the key for another duration of T_1, then the device will vibrate once for a very brief moment T_2 again. The keypad now increments its internal counter again: $\Delta i = 3$. If he or she releases the finger now, the keypad will register the key $i = 2 + \Delta i = 2 + 3 = 5$, hence `Y`.
- On the other hand, if she continues to hold down the key for another duration of T_1, then the device will vibrate twice for a very brief moment T_3 again. This is a signal that the keypad resets its internal counter: $\Delta i = 0$. If he or she releases the finger now, the keypad will register the key $i = 2 + \Delta i = 2 + 0 = 2$, hence `E`; the actual key she pressed.
- Now the Δi is reset to zero. If he or she continues to hold down the key, then the change of the internal counter Δi repeat in the same manner as before.

B. Timing of Finger Release

By counting the pulses silently in his or her head, a user can "know" the value of the hidden variable Δi throughout the duration of key-press. The variable Δi has a small range (typically 0, 1, 2, and 3) and reset itself with a twin pulse. If a user forgets the count, he or she can start the counting again after a reset. Hence, it does not require much concentration from the user.

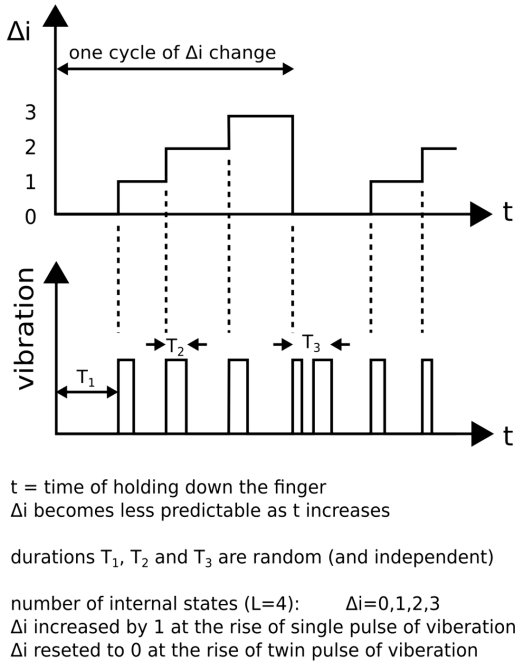

Fig. 2. Graphs of Δi and vibration vs hold-down-time

It is clear that the biggest uncertainty arises if a user releases the finger at almost the same time when a new pulse is about to start. We propose that the user should release the finger as soon as he or she "knows" the Δi. This means if the user knows that a single pulse is coming, he or she should release the finger as soon she sense the rise of the pulse. Otherwise, he or she should wait out until he or she senses a twin pulse has just passed.

C. Further Obfuscation

If the durations T_1, T_2 and T_3 are fixed, it is easy to calculate Δi by noting the duration of a key press. In other words, there is a perfect correlation of the value of Δi and the duration of a key-press. To make such calculations unrealizable, we need to reduce the correlation as much as possible. Therefore, we propose that the durations T_1, T_2 and T_3 should be random.

To further obfuscate the effectiveness of vision based side channel attacks, we suggest users to press the keypad and wait out a random number of cycles before he or she releases the keypads. The randomness of durations T_1, T_2 and T_3, and the randomness of "idling" cycles make the prediction of Δi based on the duration of key-press more difficult.

D. Recommended Usage

If there is no danger of shoulder surfing (no camera or people around) and other side-channel attacks, the user should use the keypad the way she would normally use: just press and release the actual key (corresponding to $\Delta i = 0$). On the

other hand, if he or she feels there is a need to obfuscate the password, he or she should choose a key randomly offset-ed from the target key, and hold down the key and wait out a random number of cycles before he or she release the finger for the right offset. This strategy is particularly useful if he or she is chatting with someone who can see his or her finger and keypad. In this case, he or she would just hold down the key for as short as a few seconds or as long as several minutes while he or she is engaged in a conversation.

It is worth to discuss the significance of the two choices that the user has to make when he or she want to obfuscate her input:

- Offset Δi,
- Number of cycles to wait out.

To be most effective, the user should choose these two value as random as humanly possible. It is also important to note that the user should not purposely avoid $\Delta i = 0$. It is important to note that the user should try to wait out, at least, one cycle if time permits.

E. Random Keyboard

Vision-based side channel attacks utilize the knowledge of a finger movement and keyboard layout. They include the casual shoulder surfing and sophisticated use of video recording and image analysis to track the movement of a finger. Random keyboard layout has been suggested by researchers as a way to deter such attacks. It is clear that our solution can be enhanced by the use of a random keyboard.

F. Other Variation

The central themes of our solution are based on the following observations:

- The keypad maintains a hidden internal state,
- The internal state changes in a random manner,
- The internal state is communicated to the user via a "hidden" channel (haptic feedback in our solution)
- The actual input is a function of the following: what appeared to be input, the internal state

It is clear that our solution can be applied to a variety of input methods. We are currently testing and evaluating various variations.

IV. Performance in theory

A. Assumptions

We will now perform a mathematical analysis of the performance of our solution.

Let L be the number of internal states, and `targetKey` = f(`pressedKey`, s). We assume that, for fixed `pressedKey`, the function f is injective on the variable s. In other words, for a fixed `pressedKey`, each state s is guaranteed to produce a different `targetKey`. We assume that the internal states are hidden from an attacker; and that the user employs the solution in such a way that the internal state becomes unpredictable at the time of his or her finger releases (this will be the case if the user holds down the key long enough).

Finally, we make the assumption on the attacker: the attacker has perfect information of the `pressedKey` and the mechanism of our solution.

B. Predictability of Δi

We will assume that the random variables T_1, T_2 and T_3 are independent. These variables control the duration of the individual pulses. It is important to note that, in practice, these variables must have definite positive minimum and maximum values; otherwise it would be impossible for a human user to sense the different states of the internal variables. We'll denote the minimum, maximum and the difference of T_i as follows: $\min T_i$, $\max T_i$, and $\delta T_i = \max T_i - \min T_i$.

Let t be the duration of holding down a key; then the effectiveness of our solution depends on the variation of Δi versus t. In other words, the effectiveness depends on the predictability of Δi. Since there are L internal states: $\Delta i = 0, 1, 2, \ldots, L-1$; the predictability of Δi is least if all states have the same probability $1/L$.

The probability of each internal state is predictable for a very small t. Indeed if the random variable T_1 has a definite minimum $\min T_1 > 0$; then $\Delta i = 0$ for $0 \le t < \min T_1$.

The duration of the first cycle is $LT_1 + (L-1)T_2$. The duration of one subsequent cycle is $LT_1 + (L-1)T_2 + T_3$. Let $T(N)$ be the total duration of the first $N \ge 1$ cycles, then $T(N)_{min} = NL\min T_1 + N(L-1)\min T_2 + (N-1)\min T_3$, $T(N)_{max} = NL\max T_1 + N(L-1)\max T_2 + (N-1)\max T_3$. Hence $\delta T(N) = T(N)_{max} - T(N)_{min} = NL\delta T_1 + N(L-1)\delta T_2 + (N-1)\delta T_3$.

It is clear that the internal state Δi become more and more unpredictable as $\delta T(N) = NL\delta T_1 + N(L-1)\delta T_2 + (N-1)\delta T_3$ become larger and larger. This can be achieved by increasing any of the following variables: N, L, δT_1, δT_2, and δT_3.

C. Asymptotic Probability of Δi

As the user holds down the keys longer and longer, it becomes increasingly difficult to guess the value Δi. In the limiting case, the asymptotic probability of guessing the correct Δi is $\frac{1}{L}$. If there are w characters in the password, the asymptotic probability of guessing the correct password is $\frac{1}{L^w}$.

Thus, if $L = 3$ and $w = 4$, then the asymptotic probability of guessing the correct password is $\frac{1}{3^4} = 1.23\%$. As another example, if $L = 4$ and $w = 6$, then the asymptotic probability of guessing the correct password is $\frac{1}{4^6} = 0.02\%$. This is indeed a very low probability.

In practice, several factors will affect the probability:

1. The attacker may not have perfect information about the position of the finger and the layout of the keypad. This is the case if the attacker has to rely on a video recording from a distance or indirect detection method such as using the built-in accelerator of a smart watch). Such perfect information will increase the effectiveness of our solution.

2. The human user tends to avoid releasing the key at "neutral" internal state ($\Delta i = 0$), thinking that if all target key is different from a pressed key, the attacker will have a harder

time to guess the correct answer. This argument, of course, is fallacious. The best strategy relies on total randomness. Therefore, the unconscious tendency of avoiding $\Delta i = 0$ will decrease the effectiveness of our solution.

3. As we recall: idx(`targetKey`) = idx((`pressedKey` + Δi) mod N), and $\Delta_i = g(t)$, where t is the duration of holding down the key; and g is a certain random function, see the top part of the graph in page 3. Hence idx(`targetKey`) = g'(idx(`pressedKey`), t), where g' is another random function.

There is a correlation between the `targetKey` and the `pressedKey` and t, the correlation is high: (1) when t is small; and (2) if the variance of the randomness of T_1, T_2, T_3 is small.

Therefore, if the user is impatient when entering the password, he or she may release the finger at the very first or second pulse. In that case, the probability of guessing Δi is significantly higher.

V. Performances Evaluation

In this section, we show the performances of security and summarize the users' experience.

A. Demo Application Development:

The password demo is an Android application developed using Android Studio. The basic API for the vibrator is included in API level 1, which is the very original API published in 2008. It is safe to assume almost all current Android devices come with a built-in vibrator. To obtain an instance of the system vibrator, we call `getSystemService()` with `VIBRATOR_SERVICE` as the argument. There are many API related to the controlling of the vibration. The following ones are of particular importance to our application development.

`public abstract boolean hasVibrator ()` Check whether the hardware has a vibrator.

`public abstract void cancel ()` Turn the vibrator off. This method requires the caller to hold the permission VIBRATE.

`public void vibrate (long[] pattern, int repeat)` Vibrate with a given pattern.

Pass in an array of integers that are the durations for which to turn on or off the vibrator in milliseconds. The first value indicates the number of milliseconds to wait before turning the vibrator on. The next value indicates the number of milliseconds for which to keep the vibrator on before turning it off. Subsequent values alternate between durations in milliseconds to turn the vibrator off or to turn the vibrator on. To cause the pattern to repeat, pass the index into the pattern array at which to start the repeat, or -1 to disable repeating. This method requires the caller to hold the permission VIBRATE.

The API does not provide any method to control the intensity of vibration. We can, however, control the pattern of vibration. Thus by inserting many very briefs of silence within a pulse of vibration, we can control the apparent intensity of the pulse. For example, if the duration of the pulse is 200 milliseconds, we can divide the pulse into twenty cycles of 10 milliseconds each. And within each cycle, we can turn on the vibrator for six milliseconds and turn off the vibrator for four milliseconds. The apparent intensity of the cycle is then 6/10*100% = 60%. Thus, the whole cycle will appear to be 60% of the full intensity.

TABLE I
Average consumption of time of compare

Digits	Average consumption of time (Unit:s)				
	normal	2-state	3-state	4-state	5-state
4	1.8	2.8	3.5	4.8	5.8
5	2.1	3.6	4.6	5.6	6.5
6	2.6	4.3	5.1	6.2	7.3

B. Usability Test

- User is given a list of pre-generated random passwords (each consists of 6 characters).
- User looks at the password secretly (away from the attacker) and enters it openly (no obstruction or hiding the fingers)
- An attacker has a full unobstructed view of the user's finger and keypad.
- After each user's inputting the password, the attacker write down what he or she thinks the password is on a piece of paper.
- At the end of the test, we tally the number of attacker's correct guesses.

We have a consumption of time of compare between the regular keypad and our method. We invited 30 users to test our demo applications. We employed 4-digit, 5-digit and 6-digit pass-codes. And we designed four type deferent internal states applications, 2-state, 3-state, 4-state, and 5-state, respectively. In the Table I, which contains three type length digits. The data displays average cost of normal method and our scheme. We can see that our scheme is one third times slower than the usual method. However, since our method is an "invisible" and having one-way function, it is not likely detected by vicious tricks, included side channel attacks via radio.

C. User Test and Experimental Results

We invited 30 random smartphone users to test our demo applications and then tried our best to guess their 4-digit, 5-digit, and 6-digit PINs twice. As we can see from the TABLE II and TABLE III, we can see that the more internal states, the lower the percentage of successful attacks. Besides, we also tested the usability of the corresponding number of internal states. As we can see in TABLE IV, when the number of internal states is 5, a user failed to type the correct PINs once during 50 tests. When the number of internal states is less than 5, no user failed to type the correct PINs once during 50 tests. Thus, we choose the number of internal states to 3 or 4. We simulated the attacks using an Android phone and a smart-watch with an accelerometer. The tester carries a smart-watch on his or her right wrist and an Android phone on his or her left wrist. The tester typed the PINs and the computer

TABLE II
ACCURACY OF THE SYSTEM WITH 4-DIGIT PINS

Number of internal states	Number of trials	Correct Guesses	Percentage of successful attack
2	30	1	3.3%
3	30	0	0%
4	30	0	0.0%
5	30	0	0.0%

TABLE III
ACCURACY OF THE SYSTEM WITH 6-DIGIT PINS

Number of internal states	Number of trials	Correct Guesses	Percentage of successful attack
2	30	0	0%
3	30	0	0%
4	30	0	0.0%
5	30	0	0.0%

connected the smart-watch recorded and read the information of the accelerometer. During the internal state, as shown in Fig. 3, all the deltas used in paper [1] equals to zero. This means such data are not suitable for the schemes the paper [1] used, and the smart-watch cannot get any information about the tester's input.

For the attack in paper [9], the success guessing rate of one-character (or one-digit) is over 97% while the success rate of recognizing 4-character passcodes is greater than 90%. In [10], the attack breaks an average of over 50% of the PINs on the first attempt and an average of over 85% of the PINs in ten attempts. Thanks to randomized and hidden internal states, our work can render their attacks unsuccessful because fewer motions needed to type a PINs or other credential.

Fig. 3. One testing process

D. Discussion on a robotic robbery

Touch-based authentication is a primary conventional attack these days. Serwadda et al. [11] present two LEGO-driven robotic attacks on a population statistical driven attack and a user-tailored attack. As we are known, population statistical driven attack is based on patterns gleaned from a large population of users, and user-tailored attack is launched based on samples which get from the victim, while our solution is

TABLE IV
USABILITY OF THE DEMO APP WITH DIFFERENT INTERNAL STATES

Number of internal states	Number of success trials	Number of failure of failure
2	50	0
3	50	0
4	50	0
5	49	1

based on "invisible" haptic feedback. A user presses a key on the haptic feedback-enabled touch-screen, and the device will produce a sequence of "hidden" haptic feedback to the user. This is the way the keypad communicate to the user its internal status of the keyboard. Whatever getting the position or pressure by a LEGO robot that is trained on how to swipe on the touch screen is no use for our method.

E. More discussion on the smart-watch based attack

Our experiment shows that smart-watch can not detect any haptic vibration coming from a hand-held device when a user presses his or her finger on the device. We have also demonstrated that even if the API does not support direct manipulation of the intensity of vibration of the haptic feedback, we can still control it programmatically. Therefore, we are confident that we can cloak the haptic vibration from detection by current smart-watch technology. It has been demonstrated [1] that a smart-watch can be used to detect the wearer's finger movement, and hence deduce the key the wearer is pressing. But without additional information on the haptic feedback, the smart-watch has no advantages over any direct observation technique, because it has the same uncertainty about the internal state Δi. Thus, a smart-watch wearer, employing our solution properly, can be certain that the probability of a smart-watch sensing a four-digit PIN is well below a practical value, namely: $100/3^4 = 1.23\%$ for four internal states, or $100/3^5 = 0.41\%$ for five internal states. It is clear that, if necessary, we can further reduce the probability by using a random keyboard, which cannot be sensed with a smart-watch.

Therefore, although our scheme is slower than the usual method, it is significant "invisible" method, and it is a meaningful method because it is only used to privacy password. We can use acoustic mask (such as playing music) to prevent the information leakage of the motor in cell phone.

VI. RELATED WORK

Security is an everlasting topic, especially in the mobile era. Orozco et al. [12] suggested reasonable practicality of implementing haptic-based biometric systems, and that it was an avenue worth pursuing. Kuber et al. [13] carried out a feasibility study of tactile-based authentication. After that, some tactile authentication schemes employed multi-modal interfaces [14], haptic wheel [15], and audio [16]–[18] etc. Fingerprint-based personal identification technology was used in smart-phone for several years. However, some researchers showed that

fingerprint authentication does not have a strong security level in [19] and [20] because experimental results showed that the fake fingerprints fabricated by latex or body doubles are the most difficult to discriminate. Additionally, a smart-phone with fingerprint authentication is much more expensive than a standard one. A fingerprint-based audio authentication scheme using frequency domain statistical characteristic was proposed [21]. However, FRANCESCO et al. [22] declared that they tested their approach on 154 individuals, achieving a false alarm rate of about 4 percent and an impostor pass rate of less than 0.01 percent. Therefore, characteristic authentication based on features analysis has not led to techniques providing an acceptable level of accuracy. Chang et al. [23] proposed a new graphical-based password KDA system for touch screen handheld mobile devices to displace normal keyboard. Rao et al. [24] proposed two authentication schemes that support keyboard as well as graphical mouse-based input that map password characters to other regions of the password space. It has more than 6.9% errors. Our solution is different from all the schemes. Our scheme is effective and significant: (1) We get nearly 100% success rate in our users test; (2) Our solution is light weight and less expensive, in which only an already built-in motor in a smart-phone is required; (3) Our solution can be widely applied to protect the PINs and other credentials.

VII. Conclusion

The attacks targets on smart devices will arise due to the popularity of them. Especially, it is very hard to defend against the attacks in paper [1], [11] and [25]. Our paper shows a potentially widely applied method to protect the PINs or credentials. It is reasonable to assume that when a user presses her finger on a hand-held device, vibrations from the device are transmitted through her muscle and bones to another part of the body. It is therefore theoretically possible for a wearable device to detect such haptic feedback when the user presses her finger on a vibrating hand-held device.

Acknowledgments

This work was supported in part by the State of Delaware Federal Research and Development Matching Grant Program (DEDO start-up grant) and US NSF under grants CNS-1566166.

References

[1] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "When good becomes evil: Keystroke inference with smartwatch," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1273–1285.

[2] D. Balzarotti, M. Cova, and G. Vigna, "Clearshot: Eavesdropping on keyboard input from video," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 170–183.

[3] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, "A fast eavesdropping attack against touchscreens," in *Information Assurance and Security (IAS), 2011 7th International Conference on*. IEEE, 2011, pp. 320–325.

[4] 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*. ACM, 2011, pp. 527–536.

[5] M. Backes, T. Chen, M. Duermuth, H. Lensch, and M. Welk, "Tempest in a teapot: Compromising reflections revisited," in *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 2009, pp. 315–327.

[6] M. G. Kuhn, "Electromagnetic eavesdropping risks of flat-panel displays," in *Privacy Enhancing Technologies*. Springer, 2004, pp. 88–107.

[7] A. Sarkisyan, R. Debbiny, and A. Nahapetian, "Wristsnoop: Smartphone pins prediction using smartwatch motion sensors," in *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.

[8] W. Ma, P. Duan, S. Liu, G. Gu, and J.-C. Liu, "Shadow attacks: automatically evading system-call-behavior based malware detection," *Journal in Computer Virology*, vol. 8, no. 1-2, pp. 1–13, 2012.

[9] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 1403–1414.

[10] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, "Beware, your hands reveal your secrets!" in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 904–917.

[11] A. Serwadda, V. V. Phoha, Z. Wang, R. Kumar, and D. Shukla, "Toward robotic robbery on the touch screen," *Acm Transactions on Information & System Security*, vol. 18, no. 4, 2016.

[12] M. Orozco, Y. Asfaw, S. Shirmohammadi, A. Adler, and A. El Saddik, "Haptic-based biometrics: a feasibility study," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*. IEEE, 2006, pp. 265–271.

[13] R. Kuber and W. Yu, "Feasibility study of tactile-based authentication," *International Journal of Human-Computer Studies*, vol. 68, no. 3, pp. 158–181, 2010.

[14] R. Kuber and S. Sharma, "Toward tactile authentication for blind users," in *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 2010, pp. 289–290.

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

[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*. ACM, 2011, pp. 197–200.

[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*. Springer, 2011, pp. 81–90.

[18] ——, "Counting clicks and beeps: Exploring numerosity based haptic and audio pin entry," *Interacting with computers*, vol. 24, no. 5, pp. 409–422, 2012.

[19] P. Johnson, F. Hua, and S. Schuckers, "Texture modeling for synthetic fingerprint generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 154–159.

[20] Q. Huang, S. Chang, C. Liu, B. Niu, M. Tang, and Z. Zhou, "An evaluation of fake fingerprint databases utilizing svm classification," *Pattern Recognition Letters*, vol. 60, pp. 1–7, 2015.

[21] M.-Q. Fan, H.-X. Wang, and H.-J. Li, "A fingerprint-based audio authentication scheme using frequency domain statistical characteristic," *Multimedia tools and applications*, vol. 70, no. 3, pp. 2255–2270, 2014.

[22] F. Bergadano, D. Gunetti, and C. Picardi, "User authentication through keystroke dynamics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 367–397, 2002.

[23] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, "A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices," *Journal of Systems and Software*, vol. 85, no. 5, pp. 1157–1165, 2012.

[24] K. Rao and S. Yalamanchili, "Novel shoulder-surfing resistant authentication schemes using text-graphical passwords," *International Journal of Information and Network Security*, vol. 1, no. 3, p. 163, 2012.

[25] T. Beltramelli and S. Risi, "Deep-spying: Spying using smartwatch and deep learning," *arXiv preprint arXiv:1512.05616*, 2015.