Timing Attacks on PIN Input Devices

Denis Foo Kune
CSE Department
University of Minnesota
foo@cs.umn.edu

Yongdae Kim
CSE Department
University of Minnesota
kyd@cs.umn.edu

ABSTRACT

Keypads are commonly used to enter personal identification numbers (PIN) which are intended to authenticate a user based on what they know. A number of those keypads such as ATM inputs and door keypads provide an audio feedback to the user for each button pressed. Such audio feedback are observable from a modest distance. We are looking at quantifying the information leaking from delays between acoustic feedback pulses. Preliminary experiments suggest that by using a Hidden Markov Model, it might be possible to substantially narrow the search space. A subsequent brute force search on the reduced search space could be possible without triggering alerts, lockouts or other mechanisms design to thwart plain brute force attempts.

Categories and Subject Descriptors

Computer Systems [**Computer Communication Network**]: Security and Protection

General Terms

Security

Keywords

Keystroke timing, Passcode, PIN

1. INTRODUCTION

Personal Identification Numbers (PIN) and access codes are common secrets used to authenticate users. Examples include passcode locks on touch screen smartphones, access codes for restricted doors (such as those in medical facilities) and PINs for ATMs. A number of PIN input devices have acoustic feedback mechanisms, especially if the buttons are virtual and drawn on a touch screen. An attacker at a modest distance can overhear those acoustic signals and use them to reduce the passcode combination search space. This attack becomes more important for stand-alone door access codes which are not coupled with a device that the attacker must possess such as an ATM card. To compound the problem, those keypads tend to have a loud acoustic feedback to maximize the chances of the user hearing it in a noisy environment. We apply the techniques developed by Song *et al.* [4] for inter-keystroke timing attacks including a gaussian model for the inter-keystroke timings followed by a Hidden Markov Model [3]. The original attack was modeled for a full keyboard and used timings observed by the lower layers of the network communication protocol stack. We adapt those techniques to numeric keypads and use acoustic signals of keystrokes on those numeric keypad. We are investigating the practicality of such attacks in the real world against deployed systems that use such numeric keypads. We aim at quantifying the amount of information leaking through those timings and thus modeling bounds on the reduction in search space for PINs and access codes.

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

2. CONTRIBUTION

Timing attacks using inter-key delays have been studied and shown to be successful by observing network traffic caused by those keystrokes [4]. Analyzing acoustic emanation from keyboard based on the unique sounds produced by each keys has been used to recover text being typed [1, 5]. Using those studies as a starting point, we are using the audio feedback beeps that are easier to obtain and analyze in noisy environments than characteristic acoustic emissions from individual physical keys. This becomes more important for devices that do not have a readable screen such as door keypads.

2.1 User Interface Audio Feedback

We recorded the audio feedback samples from three types of keypads; an iPhone 3G passcode lock, an office door keypad and an ATM. The numbers recorded were arbitrary and do not contain any valid secret pass codes. Figure 1 shows sample traces of the three input devices where the onset of the signal is apparent. The acoustic recordings were done in a quiet environment except for the ATM digits at the bottom of the figure. For that particular noisy sample, it was relatively easy for the human ear to separate the beeps, but it was hard to identify the key strokes on the plotted trace. To clean the trace and filter out all unwanted frequencies, we ran the acoustic sample through a Fast Fourier Transform and clipped anything outside of the 4KHz to 5KHZ range. The resulting trace is shown at the bottom of Figure 1. If we were faced with an even noisier environment, additional techniques such as source separation could be used.

2.2 iPhone Distance and Timing Measurements

As a first step, we looked at the iPhone keypad and the passcode lock mechanism. The physical layout is a standard phone keypad with the numbers "1, 2, 3" on the top row. The individual keys drawn on the touch screen are 17mm

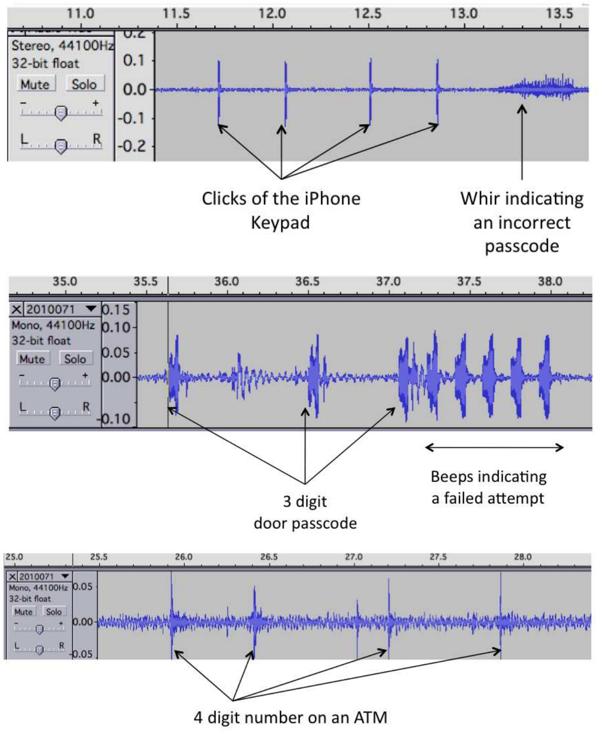

Figure 1: Audio Feedback from three sources: (from top to bottom) iPhone 3G, door and ATM.

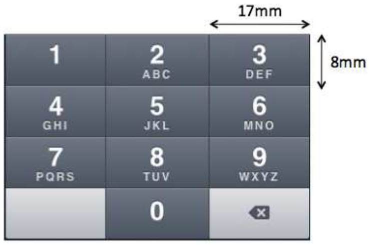

Figure 2: Physical layout of the iPhone keypad.

wide by 8mm tall. The layout is illustrated in Figure 2. An application was written asking users to enter random two-digit numbers and it measured the inter-key timings. Those timings were very close to those observed from an audio recording of the acoustic feedback. Figure 3 shows the distribution of the measured timings for the key pair $\{6, 7\}$ for a single user.

2.3 Analysis

From a previous work [4] we adopt the method to study keystroke timings but apply it to a 10 digit keypad instead of a full QWERTY keyboard. We are building a Gaussian model for the inter-key delays to produce a statistical model that would work across many users. It has been reported that users tend to have unique keystroke rhythms [2] which have been proposed for inclusion in current authentication mechanisms. By producing a distinct Gaussian distribution for equidistant key pairs, we take into account the variations by user while still providing enough information to reduce the search space. We reuse the assumptions in [4] and use only key pairs to build longer passcodes.

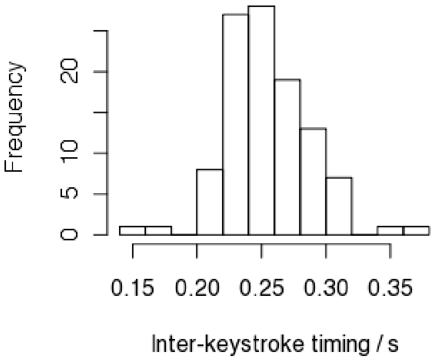

Figure 3: Distribution of inter-keystroke timing for the key pair {6,7}.

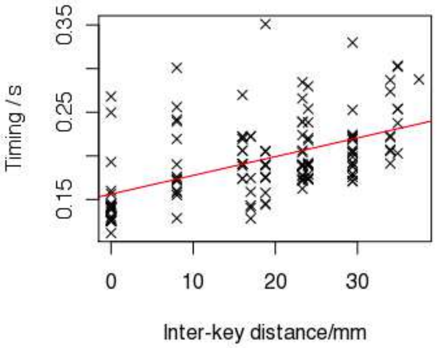

Figure 4: Linear model of timing versus distance on the iPhone keypad

2.4 Inter-Keystroke Timings and Distance

Intuitively, a user having to cover longer distances would result in longer delay in inter-keystroke timings. Measurements using the iPhone keypad show a linear relationship between inter-key distance and the inter-keystroke delay timing measurements. Figure 4 shows the result of a linear regression from a test user with the inter-keystroke timings as the exploratory variable and the distance between keys as the explanatory variable. For this particular experiment, the p-value was 1.387×10^{-10} for the user indicating that a linear correlation is highly likely.

The linear relationship between timing and distance seems consistent between users, but appears to vary in magnitude. In our analysis, the gradient of the linear models between users are not the same, although the p-values for each linear model are very small and on the same magnitude as the one illustrated above. Those results seem to suggest that different users move at different paces on the keypad and thus need to be accounted for in our Gaussian model.

3. INFERRING DIGIT SEQUENCES

With a method of observing inter-keystroke timings, we then use it to infer key sequences in PINs and passcodes.

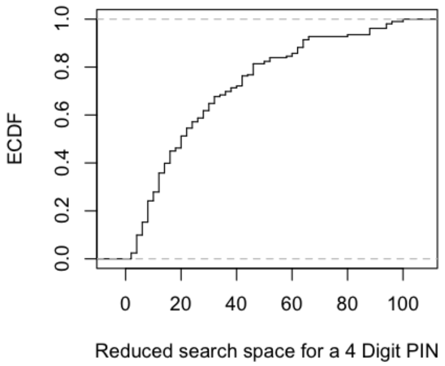

Figure 5: Search space considering only inter-key distance under ideal conditions

3.1 Hidden Markov Model

Given the correlation of key pair timings and distance covered, we use a Hidden Markov Model (HMM) with the timings as the observable outputs as the system moves to a different state. With a sequence of n timings $\vec{y} = (y_1, y_2, ..., y_n)$, we are trying to determine the hidden states $\vec{q} = (q_1, q_2, ..., q_n)$, where q_i represents a given key pair. For each observed timing y_i, where $1 \leq i \leq n$, we compute a set of most likely states $\{q_a, q_b, ..\}$. We can then use those key pair sets and look for possible combinations of keys.

3.2 Reducing the Search Space

We extract possible sequences from the list of key pairs $\{q_a, q_b, ..\}$ for each timing by aligning possible pairs with overlapping keys. For example the pairs $\{\{0, 5\}, \{5, 6\}, \{6, 2\}\}$ is a possible combination giving the sequence $\{0, 5, 6, 2\}$, while the pairs $\{\{0, 5\}, \{7, 6\}, \{4, 2\}\}$ do not yield a possible sequence although both sets of pairs have the same timing sequence.

Figure 5 shows the search space size under ideal conditions where the inter-key delay has an exact mapping to the distance covered. The median case produces 20 possible PINs for a given timing sequence. In this model, we only considered distance covered and we don't account for delay differences between keys that are different but equidistant. For example, we assume that the two pairs {1, 0} and {3,0} have the same timing since they cover the same distance. As a further refinement, timing differences for equidistant pairs can be taken into account. Thus the two pairs above may have different timings due to keys being obscured by the tapping hand.

3.3 Limitations

There is a many-to-one relationship between the set of possible sequences and the set of observed timings. In other words, it is possible to have multiple key sequences result in about the same distance traveled between keys, and therefore multiple state vectors $\vec{q}$ matching the observed timings vector $\vec{y}$. Our experiments seem to indicate that under ideal conditions, the search space is reduced by at least 2 orders of magnitude, but the search space might still be large enough to activate secondary protection mechanisms such as permanent lock outs. However, if we observe at least one key press, we can significantly shrink the already reduced search space.

4. ONGOING WORK

Our experiments suggest that the distribution of the inter-keystroke timing is not symmetrical and has a long tail. Intuitively, it is harder for a user to move quicker than to move slower. The current Gaussian model is symmetrical. We are in the process of moving our model to use empirical measurements for the distribution instead.

We are investigating the difference between tapping separate key pairs instead of full passcodes, with four digits being a very common passcode length. This will either validate the assumptions we made earlier, or give us a new model for numeric keypads. In addition, we intend to take into account the familiarity of the users with the keypads. Currently, we are considering all timings in our experiments. However, as the users get more familiar with the keypad, the timing patterns might change.

In our current experiments, users were given a random key pair to enter. In future experiments, we intend to give the users some practice on longer sequences as proposed in previous work [4] before doing the actual measurement. This will simulate users who are familiar with their PINs or passcodes.

In contrast with previous studies [4, 5], there may not be a common habit between users on their use of keypads. Therefore, we need to factor the user's tapping habits. On a smartphone's touchscreen, users can use a thumb, both thumbs, indexes or multiple fingers. On door keypads, users can use their thumbs or index fingers. On ATMs, users might not use their thumbs, but other combinations of fingers are possible.

Possible countermeasures include introducing random delays in the audio feedback that are long enough to prevent the attack described here, but short enough to prevent evaluation gaps from the user's perspective. Another countermeasure could be a system that dictates the pace at which digits need to be entered, thus avoiding timing attacks.

5. REFERENCES

[1] D. Asonov and R. Agrawal. Keyboard acoustic emanations. In *2004 IEEE Symposium on Security and Privacy, 2004. Proceedings*, pages 3–11, 2004.

[2] J. Ilonen. Keystroke dynamics. *Advanced Topics in Information Processing–Lecture*, 2003.

[3] S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice hall, 2009.

[4] D. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of the 10th conference on USENIX Security Symposium-Volume 10*, page 25. USENIX Association, 2001.

[5] L. Zhuang, F. Zhou, and J. Tygar. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):1–26, 2009.