POSTER: WiPING: Wi-Fi signal-based PIN Guessing attack

Seunghun Cha[1], Jaewoo Park[1], Geumhwan Cho[1], Jun Ho Huh[2], Hyoungshick Kim[1]
[1]Sungkyunkwan University, Suwon, Republic of Korea
[2]Honeywell ACS Labs, Golden Valley, MN USA
{sh.cha, bluereaper, geumhwan, hyoung}@skku.edu
junho.huh@honeywell.com

ABSTRACT

This paper presents a new type of online password guessing attack called "WiPING" (**Wi**-Fi signal-based **PIN G**uessing attack) to guess a victim's PIN (Personal Identification Number) within a small number of unlock attempts. WiPING uses wireless signal patterns identified from observing sequential finger movements involved in typing a PIN to unlock a mobile device. A list of possible PIN candidates is generated from the wireless signal patterns, and is used to improve performance of PIN guessing attacks. We implemented a proof-of-concept attack to demonstrate the feasibility of WiPING. Our results showed that WiPING could be practically effective: while pure guessing attacks failed to guess all 20 PINs, WiPING successfully guessed two PINs.

CCS Concepts

•Security and privacy → Side-channel analysis and countermeasures;

Keywords

Authentication; Screen Lock; Side-Channel Attacks

1. INTRODUCTION

A personal identification number (PIN) is a numeric password which is popularly used for various applications (e.g., mobile devices, automated teller machines, and point of sale terminals) that require a quick and easy way for users to prove their identity.

However, PIN based authentication systems are inherently vulnerable to brute-force attacks that try to sequentially type all possible PIN combinations because the space of possible PINs is too small (e.g., there are 10,000 possible combinations for 4-digit PIN). What is worse, people typically choose weak, memorable PINs (e.g., "0000", "1234") that are also easy to guess [2, 3].

To mitigate PIN guessing attacks, the most practical defense method is to use a security policy to limit the number of failed unlock attempts. For example, Android only allows up to 20 consecutive failed unlock attempts—if a user fails to type the correct PIN within 5 attempts, the device is temporally locked for 30 seconds; after 20 consecutive failed unlock attempts, Android displays the "Too many PIN attempts" error message, and asks the user to log in with a Google account to unlock the device.

To improve the performance of PIN guessing attacks against such a security policy, we propose a novel attack called "**Wi**-Fi signal-based **PIN G**uessing" (WiPING) that greatly reduce the size of the PIN space to be searched (during a guessing attack) by analyzing wireless signal reflections observed from the physical movements of a victim's finger while unlocking a mobile device by typing the correct PIN. Our approach was motivated by the recent advances in wireless-based motion detection and tracking [1, 5, 6]. In particular, Ali et al. [1] showed the feasibility of a keystroke recognition system that uses Wi-Fi signals generated from the hand and finger movements of a user while pressing keys on a laptop. We extend their work to mobile devices, developing a more advanced technique to detect micro-movements involved in typing a PIN on smaller touchscreen keyboards. We implemented a proof-of-concept attack tool to show the feasibility of the WiPING attack. Using a real-world PIN dataset [3], we demonstrated that WiPING can successfully guess 2 out of 20 test PINs within 20 attempts while guessing attack alone did not guess any PIN. Although at first glance this result might not seem significant, the performance of the WiPING attack can be improved considerably if we optimize the signal processing operations (e.g., Doppler shifts).

Our key contributions are summarized as follows. (1) We proposed a wireless signal based PIN guessing attack called "WiPING" to reduce the size of the PIN space that needs to be searched while performing a guessing attack. (2) We implemented a prototype and evaluated the effectiveness of the WiPING attack by comparing its performance against the performance of pure guessing attacks; a real-world PIN dictionary [3] was used to demonstrate the feasibility of the proposed attack.

2. METHODOLOGY

The proposed attack combines two techniques: (1) analyzing wireless signals to narrow the space of possible PIN candidates down to a small number of PINs with the wireless signal patterns observed, and (2) from those pre-downsized set of possible PINs, selecting just 20 most popularly used PINs based on a real-world PIN dictionary.

Through several pilot tests, we found that it is infeasible to exactly identify the location of the buttons pressed. Instead, we observed that some input behaviors between two button pressures could be clearly identified by using the Wi-Fi signals observed. For classification, we simply categorized the input behaviors for typing a PIN into the six classes as follows: X (Prod again), H (Horizontal), Vd (Vertical-down), Vu (Vertical-up), Dd (Diagonal-down),

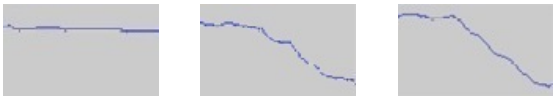
CCS'16 October 24-28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10.

DOI: **http://dx.doi.org/10.1145/2976749.2989067**

and Du (Diagonal-up) (see Figure 1). Therefore, a PIN can be represented as three consecutive Wi-Fi signal patterns. For all 4-digit PINs, there are 216 ($6 \times 6 \times 6$) possible combinations.

(a) X (e.g., 5–5) (b) Vd (e.g., 3–9) (c) Dd (e.g., 1–9)



(d) H (e.g., 1–3) (e) Vu (e.g., 9–3) (f) Du (e.g., 9–1)

Figure 1: Wi-Fi signals generated by six PIN input behaviors.

When one of 216 possible combinations is analyzed, the WiPING attack first generates a list of PIN candidates with those Wi-Fi signal patterns. It then uses a publicly known PIN dictionary to sort that list based on their occurrence likelihood in a descending order.

Our proof-of-concept implementation consists of four steps. The next sections describe those four steps.

2.1 Extracting the target signals

For the analysis of Wi-Fi signals, we assume that there is a transmitter and a receiver near the victim. In theory, any access point can be used as a transmitter. An attacker can locate the receiver (implemented using a software defined radio) at any arbitrary position to capture Wi-Fi signals (including ones related to the victim's PIN input). However, the strongest Wi-Fi signal strength is typically captured when the transmitter and receiver are collinear.

The first step of the WiPING attack is to extract signals related to the victim's PIN input from a Wi-Fi transmitter. To obtain those signals, we need to carefully choose the receiver's center frequency to selectively capture just those signals that are related to the PIN input from the transmitter while suppressing noise. We found that those parameter values could be determined experimentally with a small number of test samples.

2.2 Segmenting and smoothing

When a user types a PIN, irrelevant behaviors (e.g., initially putting fingers on a device or taking a hand off a device after typing a PIN) may also be included in the captured signals. For a proof-of-concept implementation, we simplified this process using the positions identified while being in touch with the smartphone. This process will be implemented (using machine learning techniques) in further work.

After removing irrelevant parts, signal noises also need to be eliminated. To achieve this goal, we applied a smoothing technique to extract just the necessary data. For this naive implementation, we simply ignored data below the pre-determined threshold value, and smoothened the data to, on average, adjacent five values. We also assumed that a user enters a PIN with almost equal time intervals between button presses. Thus, the recorded signals can be divided into three parts as shown in Figure 2.

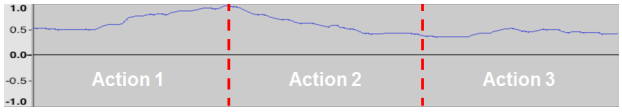


Figure 2: Signal after segmentation and smoothing.

2.3 Identifying PIN entering behaviors

To match the victim's PIN input behavior, we use the Dynamic Time Warping (DTW) algorithm [4]. The WiPING attack tool respectively compares the segmented three signals to the pre-collected data for each of the six input behaviors. In the DTW algorithm, the nearest behavior is selected. The suggested PIN pattern is "Dd Vu H" for the segmented signals in Figure 2.

2.4 Generating possible PIN candidates

Once we obtain a pattern such as "Dd Vu H", we need to enumerate a list of 20 PIN candidates for guessing attack. To implement the pattern-to-PIN guessing process, we used a 3×4 grid with valid range of dial buttons on the touchscreen. Every single finger movement should be within the grid, while the length of each movement cannot be limited. The candidate codes should start from and stay on one of the ten points (i.e. dialpad No. 1, 2, 3, ..., 0) of the 3×4 grid. We acquire a set of PIN candidates from mapping the possible movements from the given pattern. For example, with the given input "Dd Vu H", we can only deduce the following PIN candidates: `1931`, `1521`, `2631`, `1932`, and `4965` as depicted in Figure 3.

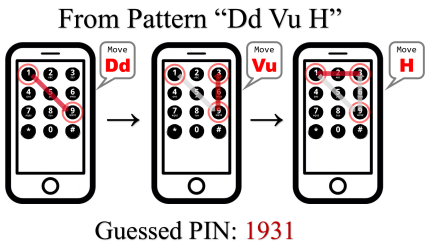


Figure 3: Finding possible PINs from the given input "Dd Vu H".

Further, when the given pattern finds PINs only less than 20, the implemented guessing system builds another patterns to analyze. For instance, if the given finger movement sequence returns unrealizable pattern such as "Du Du Du", no possible PIN candidate can be deduced from it. In such cases, our guessing system looks for the most similar pattern like "Du Du Vu", and derives more matching candidates based on those similar patterns; e.g., `0952` or others as such. Thus, every given pattern can elicit sufficient PIN candidates to perform a guessing attack—this ensures that the total number of PIN candidates is always 10,000.

3. IMPLEMENTATION AND EVALUATION

This section explains the optimal setup for performing the WiPING attack, and presents the experimental results. A video demo is available at **`https://youtu.be/bHPDeoSB03U`**.

3.1 Experiment setup and Data collection

To record the Wi-Fi signals, we used Galaxy Nexus that has a 4.65-inch touchscreen, ipTime N150UA access point (about USD 20) as a transmitter, HackRF One (about USD 300) as a receiver, and VERT2450 (about USD 36) omnidirectional antenna with a handmade directional reflecting board. To improve the attack performance, we tried several distances between the receiver and the device, and between the device and the transmitter as shown in Figure 4. Also, we set the transmitter as channel 1 (2,412 MHZ of center frequency). We experimentally determined the receiver's center frequency as 2,403.5 MHZ through a small number of tests to obtain strong wireless signals while suppressing noise.

We randomly selected 20 PINs from a real-world PIN dataset,

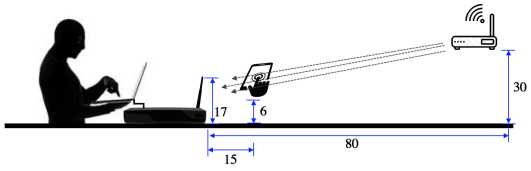


Figure 4: Experiment setup (unit : cm).

and fetched their occurrence probabilities that were also available in the dataset.

3.2 PIN behavior identification accuracy

First, we analyzed the accuracy of identifying user input behaviors. As shown in Table 1, the accuracy was mostly high except for comparisons against Dd and Vd (showing 8 incorrect inferences). This implies that the six classes considered may provide a relatively low accuracy. Instead, if we use four classes by merging classes with similar wireless signal patterns (i.e., "Dd and Vd" and "Du and Vu"), the accuracy will be significantly improved.

Table 1: Analyzing accuracy of each behavior.

	X	Vd	Vu	H	Dd	Du
X	**1**	0	0	0	0	0
Vd	0	**0**	0	0	2	0
Vu	0	0	**0**	0	0	0
H	3	2	0	**12**	3	0
Dd	1	8	0	1	**4**	0
Du	2	1	7	6	0	**7**

Table 2 shows the comparison results, which show that the accuracy can be significantly improved by using the four classes (68%) instead of the six classes (40%). Later we will study the optimal classes for identifying Wi-Fi signals related to PIN input behaviors.

Table 2: Accuracy results (6 classes vs. 4 classes).

	6 classes	4 classes
# of Trials	60	60
# of Success	24	41
Accuracy	40%	68%

3.3 Attack performance

We also compared the proportion of cracked PINs between the WiPING attack and the pure dictionary-based guessing attack. Those results are presented in Table 3. The average # of guessing attempts significantly increased from 3,431.75 to 3,074.90. While the pure dictionary-based guessing attack did not crack a PIN, the WiPING attack managed to crack 2 out of 20 PINs under 20 attempts. As an extension to this paper, we need to consider performing our tests on a larger number of samples to generalize our observation.

4. DISCUSSION

Our experimental result demonstrates the feasibility and potential effectiveness of WiPING attack. However, we observed low performance in distinguishing between diagonal movement and vertical movements (through behavior analysis) since the receiver, device, and transmitter were collinear in our experiment environment.

Table 3: Comparison WiPING attack with simple dictionary-based guessing attack.

	WiPING	Guessing
Average # of guessing attempts	**3074.90**	3431.75
Standard deviation of guessing attempts	2857.66	2991.11
Average # of guessing attempts ≤ 20	**3.50**	-
# of cracked PINs ≤ 20	2	0

As part of future work, we will consider using multiple receivers to fix this problem.

Even when we use multiple receivers, it might not be easy to distinguish some input behaviors such as "Dd-Vd", "Du-Vu", "Vd-Dd", and "Vu-Du". As part of future work, we will consider different classification techniques to improve the accuracy of identifying the input behaviors. Such improvements could significantly increase the number of cracked PINs (within 20 attempts).

5. CONCLUSION AND FUTURE WORK

This study explores the possibility of a new type of PIN guessing attacks called *WiPING* using the wireless signals generated by the victim's finger movements to enter the victim's PIN on a mobile device. To show the feasibility of WiPING, we implemented a prototype and demonstrated that the prototype can successfully guess two PINs out of 20 tested PINs within 20 attempts whereas pure guessing attacks failed to correctly guess all those PINs.

The experiments we conducted so far presented promising preliminary results. However, there is still room for improvement in recognizing fine-grained finger motions for entering a PIN. For example, for ease of implementation, our prototype implementation only used the changes in received signal strength (RSS) values of Wi-Fi signals. According to previous study [1], channel state information (CSI) values might be more suitable to recognize the micro-movements such as those of fingers and hands than RSS values. Therefore, we will also consider the information about the time-series of CSI values while entering a PIN on mobile devices.

Acknowledgment

This work was supported by the NRFK (No. 2014R1A1A1003707), ITRC (IITP-2016-R0992-16-1006), and ICT R&D program (No. B0717-16-0116).

6. REFERENCES

[1] ALI, K., LIU, A. X., WANG, W., AND SHAHZAD, M. Keystroke recognition using Wi-Fi signals. In *Proceedings of Conference on Mobile Computing and Networking* (2015).

[2] BONNEAU, J., PREIBUSCH, S., AND ANDERSON, R. A birthday present every eleven wallets? The security of customer-chosen banking PINs. In *Proceedings of Financial Cryptography and Data Security* (2012).

[3] KIM, H., AND HUH, J. H. PIN selection policies: Are they really effective? *Computers & Security 31*, 4 (2012), 484 – 496.

[4] KINGSTON, A. Speech recognition by machine, 1992.

[5] PU, Q., GUPTA, S., GOLLAKOTA, S., AND PATEL, S. Whole-home Gesture Recognition Using Wireless Signals. In *Proceedings of Conference on Mobile Computing & Networking* (2013).

[6] WANG, G., ZOU, Y., ZHOU, Z., WU, K., AND NI, L. M. We can hear you with Wi-Fi! In *Proceedings of Conference on Mobile Computing and Networking* (2014).