Study on Possibility of Estimating Smartphone Inputs from Tap Sounds

Yumo Ouchi*, Ryosuke Okudera*, Yuya Shiomi*, Kota Uehara*,
Ayaka Sugimoto*, Tetsushi Ohki* and Masakatsu Nishigaki*
* Shizuoka University, Shizuoka, Japan
E-mail: nisigaki@inf.shizuoka.ac.jp Tel: + 81-53-478-1467

Abstract— **Side-channel attacks occur on smartphone keystrokes, where the input can be intercepted by a tapping sound. Ilia et al. reported that keystrokes can be predicted with 61% accuracy from tapping sounds listened to by the built-in microphone of a legitimate user's device. Li et al. reported that by emitting sonar sounds from an attacker smartphone's built-in speaker and analyzing the reflected waves from a legitimate user's finger at the time of tap input, keystrokes can be estimated with 90% accuracy. However, the method proposed by Ilia et al. requires prior penetration of the target smartphone and the attack scenario lacks plausibility; if the attacker's smartphone can be penetrated, the keylogger can directly acquire the keystrokes of a legitimate user. In addition, the method proposed by Li et al. is a side-channel attack in which the attacker actively interferes with the terminals of legitimate users and can be described as an active attack scenario. Herein, we analyze the extent to which a user's keystrokes are leaked to the attacker in a passive attack scenario, where the attacker wiretaps the sounds of the legitimate user's keystrokes using an external microphone. First, we limited the keystrokes to the personal identification number input. Subsequently, mel-frequency cepstrum coefficients of tapping sound data were represented as image data. Consequently, we found that the input is discriminated with high accuracy using a convolutional neural network to estimate the key input.**

I. INTRODUCTION

Recently, owing to the widespread use of smartphones and cashless payment services, the input of sensitive information such as personal information and passwords on smartphones has become more prevalent. One technique to steal personal information is side-channel attack, in which a cryptographic module is observed from the outside and cryptographic analysis is performed based on the secondary information obtained. As side-channel attacks are not recorded in a log, evidences of the attacks are difficult to obtain. One of the side-channel attacks is an attack called the telecommunications electronics material protected from emanating spurious transmissions (TEMPEST) attack [1], which detects weak electromagnetic waves and sounds leaking from a display or a cable and acquires the displayed information or inputted text. Methods of TEMPEST attacks, which use the sound generated by the input to a smartphone or tablet device to infer the input content, have been proposed [2][3][4]. However, existing methods use attack scenarios that require active interference with the devices of legitimate users and therefore cannot pose a realistic threat. In this study, we created a passive attack that used the sound of smartphone tapping to estimate the input contents. We evaluated the severity of the threats and considered the defense measures.

II. RELATED STUDIES

Side-channel attacks occur on smartphone keystrokes, where the input can be intercepted by a tapping sound. Ilia et al. proposed a method for inferring input content using a built-in microphone and the tapping sounds of a legitimate user's smartphone or tablet [2]. It was demonstrated that when the terminal of a regular user had multiple built-in microphones, the sound generated by the tap was received differently by the upper and lower microphones. From the difference in the sound arrival time, we calculated the origin of the sound when the user tapped on the screen. The results indicated that the keystrokes were estimated with 61% accuracy. However, this attack scenario lacks validity because the device that wiretaps the tapping sound is a microphone built into a legitimate user's terminal, which requires prior intrusion. If the attacker has access to the target smartphone, then the malicious person can directly obtain the keystrokes of the legitimate user using a keylogger. Li et al. reported that keystrokes can be estimated with 90% accuracy by emitting sonar sounds from an attacker smartphone's built-in speaker and analyzing the reflected waves from a legitimate user's finger during tap input [3]. However, this attack is a side-channel attack, in which the attacker actively interferes with the legitimate user's terminal and can be described as an active attack scenario. Li et al. proposed an attack to infer input from the sound of typing on a physical PC keyboard [4]. The authors reported that they estimated the keystrokes with 96% accuracy using feature extraction and clustering based on a cepstrum analysis of audio data obtained by listening to the sound of a regular user's typing from a nearby microphone. This implies that even in smartphones, it is possible to infer the input of a legitimate user to a smartphone by listening to the sound of the tap input with an external microphone. In this study, we demonstrated that an attacker can hear the sound of a tap when a legitimate user inputs a key on a smartphone. Additionally, we will consider defensive measures after assessing the severity of threats in a passive attack scenario, in which an external microphone is used to eavesdrop.

III. Attack Method

In a passive attack scenario where an attacker wiretaps the sound of a legitimate user's keystrokes with an external microphone, we verified the extent to which information regarding the legitimate user's keystrokes is leaked to the attacker. First, we limited the keystrokes to the personal identification number (PIN) input. The attacker listens to the target's smartphone taps by placing a listening device in the vicinity. Features were then extracted from the collected voice data and identified using machine learning. The discriminator used was a convolutional neural network (CNN). The CNN input was a two-dimensional image of the audio data that describes, as a heat map of time transitions, the mel-frequency cepstrum coefficients (MFCC). The output of the CNN was the key information entered by a legitimate user; MFCC is typically used in speech recognition. The CNN is a deep learning method that is widely used in image recognition. Studies have been conducted where the feature vectors of an MFCC were imaged and then classified using a CNN [5]. Fig. 1 shows the flow for verifying the estimation of smartphone inputs from tapping sounds. The verification was repeated by varying the distance between the target smartphone and the microphone for wiretapping.

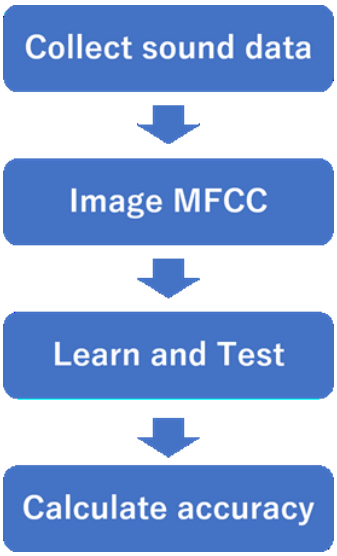

Fig. 1. Flow for estimation verification

IV. Experiment

A. Experimental Environment

Table 1 lists the specifications of the equipment used in the experiments. A soundproof room was used during the recording, and the background noise was measured. A normal sound level meter (NL-42 manufactured by Rion Corporation [8]) was installed at the location where the target smartphone was placed. When frequency weighting was used for the A characteristic and time weighting was used for the fast characteristic, the background noise level was 30-35 dB.

Table 1. Experimental equipment

Equipment	Name
Smartphone	iPhone 6
Tablet	iPad Pro
CPU for analysis	Intel® Core™ i7-6500U@2.50GHz
OS (PC)	Windows 10 Pro
Audio editing software	Audacity
Programming language	Python 3.7
Speech processing library	librosa 0.7.0
Deep learning library	Keras 2.3.1
	TensorFlow 1.14.0

B. Sound Dataset

The experimental environment is shown in Figure 2. In the soundproof room, the PIN input interface of a Japanese software keyboard was displayed on the screen of the target smartphone. A participant wore earphones and tapped each of 10 numbers from 0 to 9 consecutively for 100 times such that the fingernails touched the screen for 1,000 taps. To simplify the analysis of voice data, a metronome voice of 100 bpm was played through the earphones during the tapping, and the experimenter tapped to the rhythm of the metronome. In a passive attack scenario, we believe that recording microphones are better suited to mobile devices used by attackers than precision microphones or listening devices. In this study, due to the size of the tablet devices and the limited size of the soundproof room, four tablet devices (recording microphones) were placed at 10, 30, 50, and 70 cm from the target smartphones and tapping sounds were recorded simultaneously. The format of the audio data was m4a.

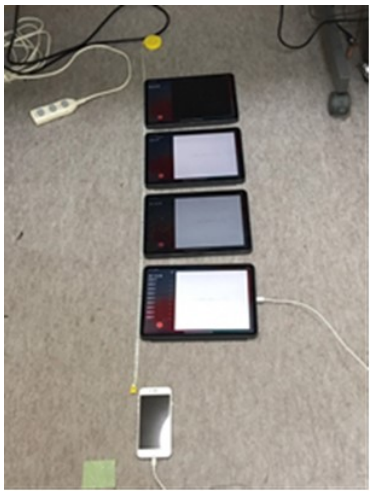

Fig. 2. Experimental condition

C. *Sound Processing*

After the recording, the audio data for each number was converted from an m4a format to a wav format. Using the audio editing software Audacity [6], the entire audio data was time split every 0.35 s so that each tap became its own audio file. Fig. 3 shows an example of audio data division. We used librosa, a speech processing library in Python, to create two-dimensional images ("MFCC images") from the MFCC of each tap in the audio data. An example of the image is shown in Fig. 4; it is a heat map of the MFCC coefficients in 20 dimensions, with time on the horizontal axis and 20 dimensions on the vertical axis. In the actual identification, the color bars, axes, and labels were deleted and output as a 640 x 480 pixels image. Because the MFCC images were color images, the CNN received three channels of RGB image input.

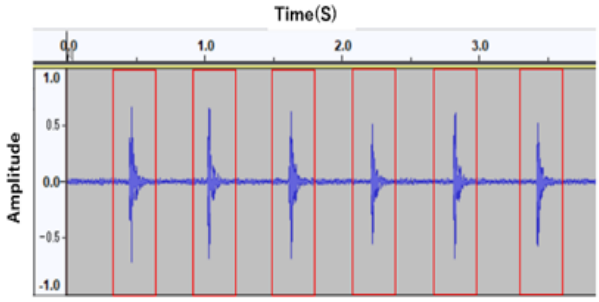

Fig. 3. Time partitioning of audio data

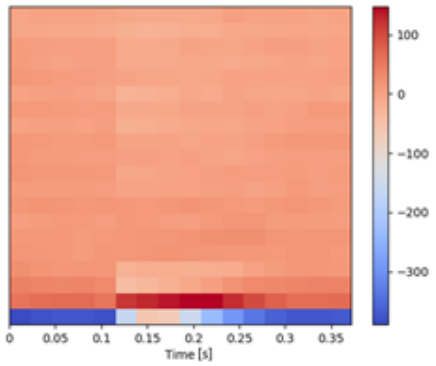

Fig. 4. MFCC illustration

D. *Machine Learning*

In this study, we used Keras and TensorFlow, which are deep learning libraries in Python, to train and discriminate using the CNN. We referred to [7] for the network configuration of the CNN. The CNN model used in this study is shown in Fig. 5. The MFCC images (three channels of RGB images) described in Section 4.3 were used as input to the CNN. The CNN first compressed them to 50 × 50 pixels and then transformed them in two consecutive steps using 3 × 3 filters to obtain 32 feature maps. Next, max pooling (a pooling operation to extract the maximum element for a small region) was performed, and the image size was reduced by half to 2 × 2. After this, we performed two consecutive convolutions and max pooling. The resulting three-dimensional arrays were smoothed in one dimension and connected to all of the bonded layers. The activation function was a softmax function in the output layer and a ramp function in the other layers. In the training and evaluation of the discriminator, all MFCC images created (as described in Section 4.3) were partitioned into testing and training data with a ratio of 8:2. Furthermore, the training data was partitioned into validation and learning data with a ratio of 8:2.

V. RESULT

Fig. 6 and Fig. 7 show the learning curve. In Fig. 6, the vertical axis is the accuracy, and the horizontal axis is the epoch, where "acc" represents the accuracy of the training data, and "val_acc" the percentage of correct responses for the validation data. In Fig. 6, the vertical axis is the loss, and the horizontal axis is the epoch transition, where "loss" is the loss of the training data and "val_loss" is the loss of the validation data. Fig. 7 shows that the training data was trained with almost 100% correct answers. Because the percentage of correct responses for data verification was high, we confirmed that we were able to create the model without overlearning.

We evaluated the accuracy of the trained CNNs by inputting the data for evaluation. A five-fold cross-validation was performed to calculate the discrimination rate and the average of the percentage of correct responses, obtained from the five evaluations, was used. The results are shown in Table 2, confirming that our attack can identify the PIN input with high accuracy.

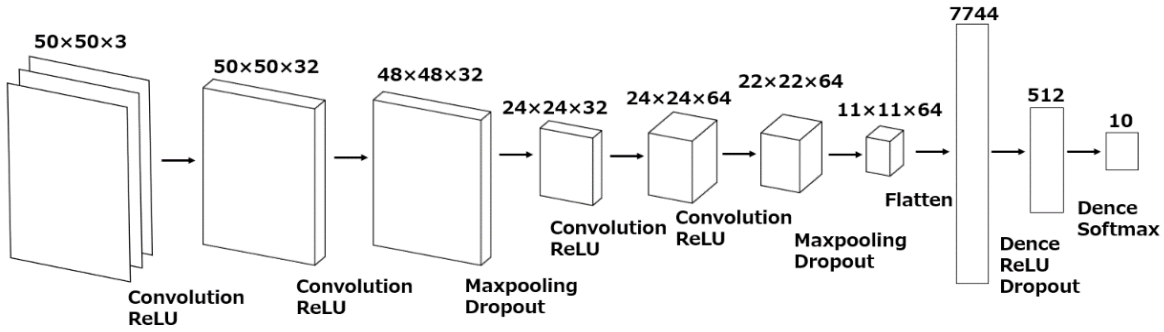

Fig. 5. CNN model

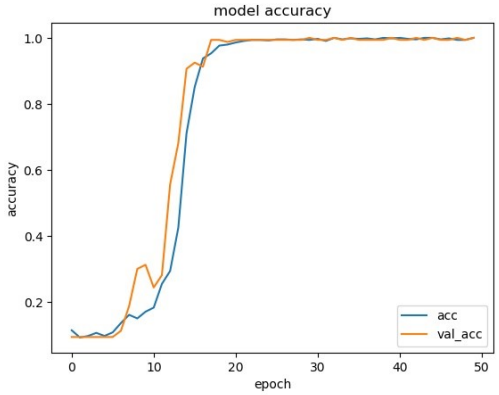

Fig. 6. Learning curve of accuracy

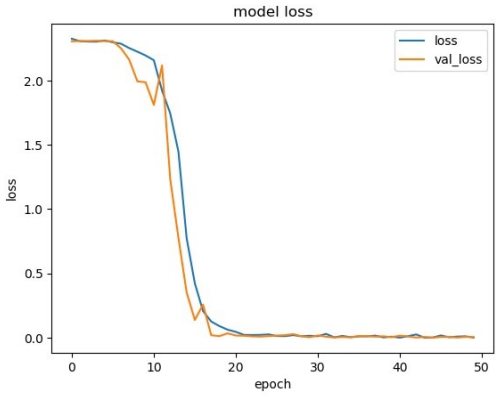

Fig. 7. Learning curve of loss

Table 2. Accuracy by distance

Distance (cm)	Accuracy (%)
10	98.9
30	99.0
50	98.4
70	96.3

VI. DISCUSSION

A. *Limitations*

As shown in Table 2, we can expect that CNN-based input prediction attacks are feasible with high probability. However, there are many limitations to this evaluation. The first limitation is the tapping method. In our experiment, we collected the sound of fingernail tapping on a smartphone, which is limited to behavioral scenarios. Therefore, it is necessary to verify the estimation of tapping with a finger pad only. Furthermore, in the experiment, for each number from 0 to 9, 100 taps were tapped continuously and equally at fixed time intervals. Therefore, it is necessary to collect audio data of arbitrary taps and learn them. The second limitation is the attack environment. In this experiment, the data was recorded in a soundproofed room and evaluated in a noise-free environment. In the future, experiments should be conducted in various environments with various noise levels for evaluation. In our experiment, smartphones for tapping and tablets for recording were set up on the floor of the soundproof room. However, various situations exist in the actual attack environment, like when the smartphone is placed on a table or held in the hand. In the future, experiments and evaluations should be conducted in situations where users are using a smartphone. The third limitation is terminal dependence. In this experiment, we evaluated the combination of a specific smartphone device and a recording device. However, the characteristics of the tapping sound may depend on the smartphone used by a legitimate user and the recording device used by an attacker. In the future, we will evaluate various smart phones and recording devices. The fourth limitation is the number of participants. Because this study was only a basic study, it was conducted on only one participant. In the future, we will increase the number of participants for evaluation. The fifth limitation is the attack target. In our study, we limited the key input to PINs. In the future, we will consider both a flick-type 50-note keyboard and a QWERTY keyboard. The sixth limitation is the attack method. In this study, we assumed that the attacker had access to the correct answer of a legitimate user's keystrokes during CNN training. In the future, we will perform an experiment involving an attacker learning the CNN using his own keystrokes and then estimating the user's keystrokes using this CNN during the attack.

B. *Defensive measures*

In our experiment, we did not observe a significant change in the percentage of correct keystroke estimates when the distance between the target smartphone and the attacker's tablet device (recording device) differed. Therefore, merely “keeping a distance from the attacker” may not be an effective defense measure. Using adversarial and other examples, we are of the opinion that it is necessary to examine the method of synthesizing environmental sounds to effectively disturb CNNs. Li et al. considered two defenses against keystroke attack methods that infer keystrokes based on key-by-key characteristics of keyboard strikes: the input environment and input content [5]. From the point of view of the input environment, it is important to confirm that there is no wiretapping device in the room and that no sound can be intercepted from outside the room. Furthermore, the defensive measure is not only a simple password but also a combination with one-time password and biometric authentication. These countermeasures are also effective as a defense measure against attacks.

VII. CONCLUSION

In this study, we analyzed the extent to which a user’s keystroke information is leaked to an attacker in a passive attack scenario, where the attacker wiretaps the sound of the user's keystrokes using an external microphone. First, we

limited the key input to the PIN input only. The MFCC of tapping sound data were represented as image data and a CNN was used to estimate the key input. We discovered that the input can be discriminated with high accuracy. Although the experiments were conducted in a soundproof room, no difference was observed in the estimation accuracy between the attacker and the target in the range of 10-70 cm. In the future, we will perform a more comprehensive analysis of the defense measures.

REFERENCES

[1] National Security Agency: TEMPEST fundamentals, NACSIM 5000, Feb 1982.

[2] Ilia Shumailov, Laurent Simon, Jeff Yan and Ross Anderson: Hearing your touch: A new acoustic side channel on smartphones, arXiv : 1903.11137, 2019.

[3] Li Lu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Xiangyu Xu, Guangto Xue and Minglu: KeyListener: Inferring Keystrokes on QWERTY Keyboard of Touch Screen through Acoustic Signals, IEEE INFOCOM 2019.

[4] Li Zhuang, Feng Zhou and J. D. Tygar: Keyboard Acoustic Emanations Revisited, ACM Conference on Computer and Communications Security, November 2005, pp. 373-382.

[5] Leon Mak An Sheng and Mok Wei Xiong Edmund: Deep Learning Approach to Accent Classification, CS229 2017.

[6] Audacity: The Free, Cross-Platform Sound Editor, available from〈http://audacity.sourceforge.net〉(accessed 2020-08-06)

[7] Keras Documentation: Train a simple deep CNN on the CIFAR10 small images dataset (online), available from 〈 https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn.py 〉 (accessed 2020-08-06)

[8] RION, Sound and Vibration; Web-support Support Room: Sound Level Meters, available from < https://rion-sv.com/products/10005/NL420009> (accessed 2020-08-03)