I Know Your Keyboard Input: A Robust Keystroke Eavesdropper Based-on Acoustic Signals

Jia-Xuan Bai, Bin Liu*, Luchuan Song
CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei, China
bjx@mail.ustc.edu.cn,flowice@ustc.edu.cn,slc0826@mail.ustc.edu.cn

ABSTRACT

Recently, smart devices equipped with microphones have become increasingly popular in people's lives. However, when users type on a keyboard near devices with microphones, the acoustic signals generated by different keystrokes may leak the user's privacy. This paper proposes a robust side-channel attack scheme to infer keystrokes on the surrounding keyboard, leveraging the smart devices' microphones. To address the challenge of non-cooperative attacking environments, we propose an efficient scheme to estimate the relative position between the microphones and the keyboard, and extract two robust features from the acoustic signals to alleviate the impact of various victims and keyboards. As a result, we can realize the side-channel attack through acoustic signals, regardless of the exact location of microphones, the victims, and the type of keyboards. We implement the proposed scheme on the commercial smartphone and conduct extensive experiments to evaluate its performance. Experimental results show that the proposed scheme could achieve good performance in predicting keyboard input under various conditions. Overall, we can correctly identify 91.2% of keystrokes with 10-fold cross-validation. When predicting keystrokes from unknown victims, the attack can obtain a Top-5 accuracy of 91.52%. Furthermore, the Top-5 accuracy of predicting keystrokes can reach 72.25% when the victims and keyboards are both unknown. When predicting meaningful contents, we can obtain a Top-5 accuracy of 96.67% for the words entered by the victim.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Information systems** → **Mobile information processing systems**.

KEYWORDS

Acoustic sensor; Keyboard snooping; Signal processing; Robustness

ACM Reference Format:
Jia-Xuan Bai, Bin Liu*, Luchuan Song. 2021. I Know Your Keyboard Input: A Robust Keystroke Eavesdropper Based-on Acoustic Signals. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475539

1 INTRODUCTION

With the popularity of smart devices, attackers can obtain data sources to carry out side-channel attacks from user's inadvertent privacy leaks. Nowadays, companies like Apple and Google have invested tremendous costs in the research and development of technologies like intelligent voice assistants. Due to the expansion of the intelligent voice market, acoustic signals are more likely to be the breakthrough of side-channel attacks.

Recently, researchers have done some works about side-channel attacks on keystrokes using acoustic signals. The work in [10] proposed to use the neural network to extract the special features of different keystrokes. The premise of this algorithm is that a large number of datasets are required to support the training of the neural network, which may be difficult to satisfy in the actual situation. Some other works aimed to predict keystrokes by manually extracting Mel-Frequency Cepstral Coefficients (MFCC) and Time Difference of Arrival (TDoA) from acoustic signals, assuming the microphone and keyboard position are fixed[15]. However, these positions may be varied in actual situations. The work in [18] recognized the victim's keystrokes by emitting ultrasonic waves and receiving reflected signals. However, this method is more susceptible to multipath reflections and keyboard position changes, which affects the robustness in reality.

Three main challenges lie in the scheme of implementing a robust attack system against keystrokes. The first challenge issue of acoustic-based side-channel attack is the lack of prior knowledge. A robust system should not require much pre-training to conduct the attack. On the other hand, the relative position between microphones and the keyboard may affect the collected acoustic signals. A mature attack system should be able to adapt to different relative positions. Secondly, the identification of keystrokes is a fine-grained task because of the subtle difference between different keystrokes. Moreover, keystroke's acoustic signals can vary significantly with victims and keyboards. The key to distinguishing keystrokes is designing features that are accurate, stable, and impervious to these factors. The last challenge is the number of devices. In real situations, it is easier to prevent keystrokes from being eavesdropped on by a large microphone array. We hope that the attack can be conducted using only one smart device without the victim's attention.

To address the above challenges, we propose a scheme using only a single smartphone with dual microphones to conduct a side-channel attack. Facing the unknown attacking environment, we propose a position estimation scheme to obtain the relative position between the microphones and the keyboard. The microphone coordinates are achieved through a parameter optimization algorithm

* Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475539

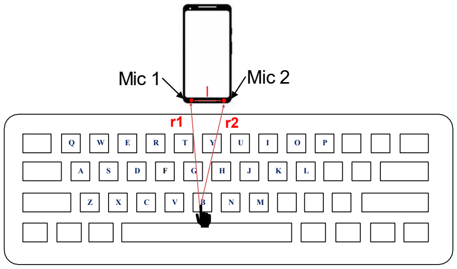


Figure 1: The Attack scenario of the system.

based on TDoA between the generated acoustic signals measured by two microphones. Therefore, the proposed scheme allows us to learn the initially unknown input environment and build a small training set in the restored environment to predict unlabeled signals. Two robust and stable features, i.e., TDoA and Power Spectral Density (PSD), are then extracted to distinguish keystrokes in different positions. To ensure the robustness of the proposed scheme, the features we choose mainly vary with the keystroke position and are less affected by factors such as keystroke tone and sound intensity.

Our contributions can be summarized as follows,

- We propose an efficient scheme for environment estimation to address the challenges from the position variation of microphones and the lack of training data.
- We design a robust algorithm including efficient features extraction and model training to realize the fine-grained identification of keystrokes from unknown victims and keyboards.
- We implement the proposed scheme on a single commercial smartphone and conduct extensive evaluations. Experimental results show that the robustness and accuracy of the proposed scheme surpass the state-of-the-art works.
- To the best of our knowledge, this paper is the first attempt to identify the keystrokes from unfamiliar environments without victims’ ground truth data.

The rest of this paper is organized as follows. We first show the system design of the proposed scheme in Section 2. Then The evaluation of the system is presented in Section 3. Finally, we review several related works in Section 4 and conclude in Section 5.

2 SYSTEM DESIGN

In this section, we describe the design of the proposed scheme, which performs side-channel attacks on keyboard input leveraging the microphones on a single smartphone.

2.1 Attack Scenario

The attack scenario can be considered as follows. We assume that the attacker inadvertently leaves a smart device near the victim’s keyboard and utilizes the two microphones on the device to receive acoustic signals. To facilitate processing, we assume that keyboard layouts of the same type are roughly the same. However, since the input environment is controlled and adjusted by the victim, we lack knowledge about the type and location of the keyboard. Here we focus on the task of recognizing 26 letters and 6 large-size keys, including Enter, Space, the left Shift, the right Shift, the left Control, and Caps Lock, and the extension to other keys is straightforward. Fig. 1 illustrates the attack scenario.

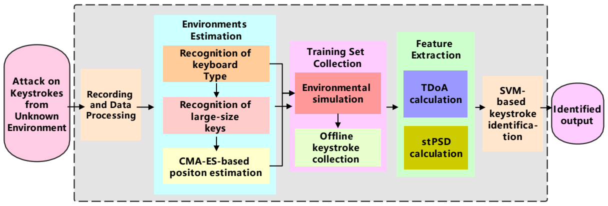

Attack on Keystrokes from Unknown Environment
Recording and Data Processing
Environments Estimation
Recognition of keyboard Type
Recognition of large-size keys
CMA-ES-based positon estimation
Training Set Collection
Environmental simulation
Offline keystroke collection
Feature Extraction
TDoA calculation
stPSD calculation
SVM-based keystroke identifica-tion
Identified output

Figure 2: Illustration of the working flow to predict the victim’s keystrokes with acoustic signals.

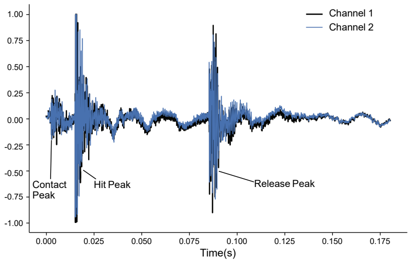


Figure 3: The Waveform of keystroke acoustic signals.

2.2 System Overview

The goal of the proposed scheme is to ensure that the side-channel attack adapts to various unfamiliar environments and still maintains high accuracy. In the proposed system, we assume that the relative position of the microphone and keyboard, the type of keyboard, and the victim’s information are all unknown. Furthermore, we do not need the labeled ground truth data from victims before conducting an accurate attack. Side-channel attacks on keystrokes can be summarized in the following steps.

Recording and data processing Firstly, the attacker record the acoustic signals of keystrokes with two microphones placed around the keyboard. Then they perform segmentation processing on the collected acoustic signals to convert the entire signal to a series of segments, each of which containing only one keystroke.

Environment estimation Next, attackers extract two aspects of information from unlabeled acoustic signals. On one hand, tones of the keystrokes are used to identify the type of keyboard the victim employed when typing. On the other hand, attackers distinguish the parts containing the large-size keys from the other keystrokes based on their tones. On the basis of the above information, a parameter optimization algorithm is utilized to estimate the relative position of the microphones.

Training-set collection Through the previous step, the type of keyboard and the relative position of microphones have been learned. Therefore, attackers can simulate the input environment by placing a keyboard of the same type as the one used by the victim and setting the microphones according to the estimated relative position. In the simulated scenario, a small training set, for example,

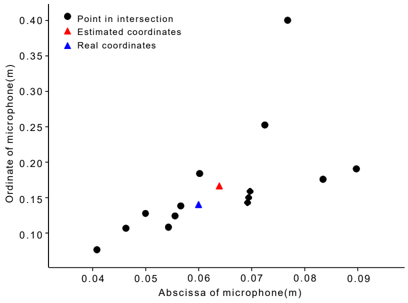


Figure 4: Intersection set and estimated position

including acoustic signals of each keystroke from only one user, can be built to serve the model training.

Feature extraction and model training To identify keystrokes, we extract two robust features from the acoustic signals to identify keystrokes. The selected features are extracted from the training set collected in the simulated scenario and used to train the Support Vector Machine (SVM) model for predicting.

Keystroke prediction Based on the trained model, attackers can accurately predict the initially collected unlabeled keystrokes of victims.

In summary, we conduct the prediction of keystrokes through an offline method. However, since the purpose of the side-channel attack is to decipher the victim's input, there is no obvious difference between online and offline approaches.

2.3 Data Processing

In this part, we mainly describe the pre-processing of the collected signals. Firstly, we normalize the values of the collected signals to the range of $[-1, 1]$. After the normalization process, we divide collected signals into segments of keystrokes. A typical keystroke signal can be divided into three parts: contact peak, hit peak, and release peak. Fig. 3 illustrates these three peaks in the acoustic signal of a single keystroke. In order to segment signals, we use empirical thresholds to detect the presence of contact peaks. Once either of the two normalized values of the channels exceeds 0.2, we consider that the keystroke signal enters the part of the contact peak, and take this time point as the starting point. Next, we start from the determined starting point and divide the next $180ms$ signal into individual keystroke segments according to experience. After data pre-processing, we can get a series of normalized acoustic segments of keystrokes for feature extraction.

We then perform framing and windowing operations on these segments. The characteristics of the acoustic signal and its fundamental parameters all change with time, so it is a non-stationary process. Nevertheless, its features remain unchanged in a short time range, which makes the acoustic signal can be regarded as a quasi-steady-state process. Therefore, we process the signal into frames to obtain features that better reflect the time-domain characteristics. To better reflect the temporal variation of the signal, we set the length of each frame as $10ms$, shifting $2.5ms$ each step. After that, we apply a Hanning window to the signal of each frame to reduce spectrum leakage. With the result of framing and windowing, we can extract features on each frame to better reflect the time-domain characteristics of the signal.

2.4 Environment Estimation

Most researchers have not considered the influence of the relative position between the microphones and the keyboard in the prediction of keystrokes. When the relative position changes, most characteristics of the keystroke signal may be affected.

In this part, we propose a method to address the challenge of the input environment's possible change. Without prior knowledge and labeled datasets, attackers can extract environmental information from the collected acoustic signals, including the type of keyboard and the microphone's relative position.

Firstly, we use a series of collected samples to determine the type of keyboard without any environmental information. Three main types of keyboards are frequently used for typing in daily life, including mechanical keyboards, membrane keyboards, and laptop keyboards. For the same type of keyboard, the mechanical structure, position of each key, and tones of keystrokes are similar, which are greatly diverse among different categories. In order to reflect the difference in tones, we adopt MFCC as the features to classify the type of keyboards. MFCC uses the amplitude of the Fourier transform of the time domain speech frame to analyze the acoustic signal. Specifically speaking, we extract 16 MFCC from the framed and windowed data. Then SVM [9] is used as the classification algorithm in the task of utilizing MFCC to determine the keyboard type, which can reach an accuracy of close to 100%.

After learning the type of keyboard, attackers still do not know any information about the microphone's relative position. Next, we try to identify a series of large-size keys from acoustic samples and estimate the microphone's position based on their coordinates. On the keyboard, the tones of large-size keys such as Space are always different from other keys, so we can use MFCC features to distinguish the larger-size keys. For each type of keyboard, we collect acoustic signals of the commonly used large-size keys in advance, as the training set of the SVM to identify these keys from unlabeled acoustic signals.

Next, we calculate the TDoA corresponding to these recognized large-size keys from the acoustic signal. TDoA reflects the relative position between the keystrokes and the microphones and is hardly affected by other factors. Thus, we consider it as a robust feature. We utilize cross-correlation to calculate TDoA of each recognized large-size key K, represented by t_K, and get its coordinates from its identification results when we consider 'Q' on the keyboard as the origin of the coordinates. Cross-correlation function R can be expressed as follows,

$$R_{x_1,x_2}[\tau] = \sum_{-\infty}^{+\infty} x_1^*[t]x_2[\tau + t] \qquad (1)$$

where x_1 and x_2 represent the signals of the two channels.

In summary, we have identified a series of large-size keys without training data from the victim and obtained their corresponding TDoA and coordinates. Then, the position of the microphone can be estimated from only this information. For each recognized large-size key K, their theoretical formula of TDoA can be obtained

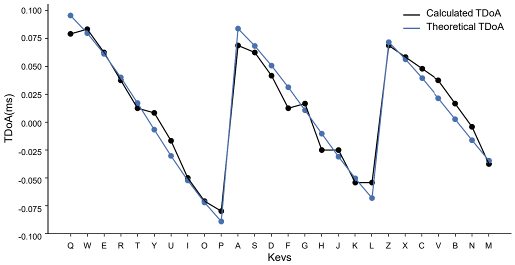


Figure 5: Illustration of the difference between 26 letters’ calculated TDoA and theoretical TDoA

according to Eq. (2), represented by Δt.

$$\Delta t = \frac{\sqrt{(u-u_1)^2 + (v-v_1)^2} - \sqrt{(u-u_2)^2 + (v-v_2)^2}}{V_s} \qquad (2)$$

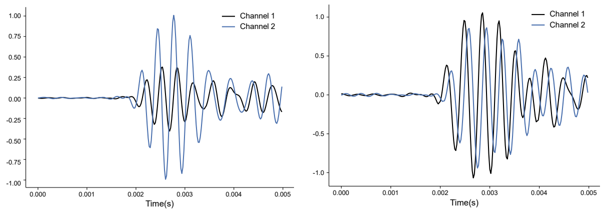
Here we assume that ’Q’ on the keyboard is the origin of the coordinates, and the position coordinates of the two microphones are (u_1, v_1) and (u_2, v_2), respectively. The coordinate of the pressed key is represented as (u, v). V_s represents the speed of sound in the air.

From the perspective of attackers, the microphones are always in positions where Δt and t_K are equal since they symbolize the TDoA values calculated in theory and measured in practice, respectively. Therefore, we take the microphones’ coordinates as parameters and utilize Covariance-Matrix Adaptation Evolution Strategy (CMA-ES) to optimize their corresponding Δt, making it close to t_K. The distance between the two microphones on the smartphone can be used as a constraint in the optimization. The CMA-ES algorithm is a kind of evolutionary algorithm used to solve parameter optimization problems [11]. Compared with other algorithms, CMA-ES is more efficient when optimizing fewer parameters and can approach the target faster when the parameter range is unknown. After the optimization is finished, Δt’s parameters (u_1, v_1) and (u_2, v_2) represent the possible positions of the microphones.

For each recognized large-size key, we run CMA-ES multiple times to get a set of possible positions, where the specific number of runs is selected to 100 based on experience. Thus, each recognized key can obtain a set of possible microphone positions corresponding to itself. Ideally, the intersection of all sets obtained based on the recognized large-size keys should be a certain point, representing the position of the microphone. However, due to the existence of errors, we usually get an intersection set, and the centroid of the set is considered as the final estimate of the microphone position. To avoid the influence of singular values, we ignore the intersection point where the abscissa or ordinate is too different from the mean value. Taking the coordinates of the microphone on the left as an example, we conduct experiments based on the mechanical keyboard and the scene in Fig. 1. Fig. 4 reveals the intersection set and the estimated position.

2.5 Feature Extraction & Model Training

According to the relative position of the microphone and the type of keyboard, we can simulate the victim’s input environment by arranging a scene similar to the input environment and thus collect some samples to train the keystroke prediction model. Specifically,



(a) (b)

Figure 6: (a) The acoustic waveform of key ’Q’ (b) The acoustic waveform of key ’P’

attackers can choose the location to place the same type of keyboard as the victim used, and set the microphones according to the estimated relative position. The work of environmental simulation and offline training-set collection enables the proposed scheme independent of prior knowledge and adapting to unknown environments.

To predict keystrokes in unknown environments, attackers need to find robust features in the keystroke’s acoustic fragment. The principle of selecting features is to avoid influencing factors such as the sound intensity and tone of the keystrokes as much as possible. Therefore, we try to ignore the features related to the sound of keystrokes themselves and pay more attention to revealing the key’s positions. Based on the robustness of TDoA introduced in Section 2.4, it is chosen as one of the features to predict keystrokes. Fig. 5 illustrates the difference between the calculated TDoA and the ground truth value of 26 letters on the keyboard, proving the validity of the proposed algorithm. The sampling rate of the microphones is set to $48kHz$. We conduct this experiment on mechanical keyboards in the scenario illustrated in Fig. 1.

However, there may be multiple keys on the keyboard with close values of TDoA. To ensure the robustness of the proposed scheme, we hope to find other robust features from acoustic signals to realize fine-grained keystroke recognition. Since the energy of the acoustic signals would be dispersed in the air and absorbed by the propagation medium, the acoustic signals will attenuate with distance during the propagation process. Assuming that the energy of signals at point a is I_a and the energy at point b is I_b, the attenuation between the two signals can be expressed as[1, 22]:

$$I_a = I_b \frac{k}{d} e^{\alpha d} \qquad (3)$$

where k is a normalization coefficient, α is the attenuation coefficient, d represents the distance between point a and b. Since the initial energy and the propagation medium of the two collected signals are the same, differences between the energy of the two signals reflect the distance gap. To calculate the difference of the two channels in sound attenuation, we choose PSD as one of the features, which reflects the change of signal power with frequency. Because the signals of the keystrokes are concentrated in several frequencies, we can reduce some of the noise interference by removing some useless low-frequency and high-frequency parts. According to the Wiener–Khinchin theorem [26], PSD $P(\omega)$ can be expressed by the following formula,

$$P(\omega) = \int_{-\infty}^{\infty} R_x(\tau)\mathrm{e}^{-\mathrm{j}\omega\tau}\mathrm{d}\tau \tag{4}$$

where $R_x(\tau)$ represents for the auto-correlation of signal $x(t)$.

Due to the robustness to victim users and keyboards, TDoA and PSD are selected as features. When extracting features, we do not target the entire keystroke segment. In addition to the contact peak, hit peak, and release peak described above, there are still many invalid periods in the entire keystroke segment. We choose the first $20ms$ segment to calculate TDoA because the sound signal that arrives first travels along a straight line from the sound source to the microphone and is minimally interfered with by multipath reflections and noise. Next, we select the frames included in the 3 peaks and calculate the PSD on each of them.

Fig. 6 shows the first $5ms$ waveform of the acoustic signal generated by the keys 'Q' and 'P' collected from the mechanical keyboard in the scene of Fig. 1. In Fig. 6(a), the phase of channel 1 lags behind that of channel 2, and the amplitude of the signal is also smaller. In contrast, the situation is just the opposite in Figure 6(b). It further proves that TDoA and PSD can reflect the location of keystrokes. Moreover, the signal waveform of the two acoustic channels is similar, which ensures that we can calculate TDoA through cross-correlation.

After extracting the features, we choose SVM to classify keystrokes, since it usually has better results than neural networks and other machine learning methods when only leveraging a small number of sample sets.

2.6 Word-Level Precision

When predicting meaningful contents, the recognition accuracy of keystrokes can be improved through the arrangement of letters in each word, and the corresponding word prediction could be given at the same time. In the process of English input, there are always Space and Enter keys between words. We identify the locations of these keys in the signal and consider the part between them as a word's signal to get the corresponding sequence of each word.

For each keystroke the algorithm predicts, we can get a pair of parameters containing the predicted results and their confidence. After disposing of all keystrokes included in a word, we can get a sequence of such pairs of parameters. To increase the accuracy and fault tolerance, we select the top-5 results sort by the confidence of each keystroke prediction as the input for word-level prediction.

To make predictions at the word level, we establish a dictionary containing the most frequently used 1500 words. Every time a sequence of results is obtained after predicting keystrokes in a word, the algorithm will select possible words matching its length from the dictionary. The confidence of each word is the sum of the confidence of its letters. From this, we can get the confidence of all words. Based on the proposed algorithm, the dictionary can be easily expanded and deleted.

3 EVALUATION

3.1 Implementation & Methodology

In this part, we comprehensively evaluate the performance of the proposed scheme. We implement the system on HUAWEI Mate20 Pro as the device. The two microphones are equipped on the bottom of the phone. An Android application that is developed with Java codes is utilized to realize the side-channel attack. When the attack process begins, the phone collects acoustic signals at a sampling rate of $48kHz$.

In order to ensure robustness, we select multiple keyboards with different tones and sound intensities to implement the system. 9 different keyboards are utilized to evaluate, including 5 mechanical keyboards, 2 membrane keyboards, and keyboards for Lenovo and Alienware laptops. In particular, the sound intensity of a mechanical keyboard is stronger, while keystroke sounds on the laptop are the weakest. On the other hand, the design and dimension of different types of keyboards are different, which is why we have to divide the types of keyboards.

We focus on the evaluation of 32 keys and proceed to collect data. We invite 9 volunteers to be the victims, including 2 women and 7 men. To evaluate the performance of the system from multiple aspects, we set up different experimental environments to collect data. Each experimental environment is determined by 3 factors, including the microphone's relative position, the victim, and the keyboard. We arrange 37 experimental environments for data collection, covering all keyboards and victims, and cover 4 different microphone positions, including the top, bottom, left, and right of the keyboard. In each experimental environment, the victim presses each of the 32 keys 10 times. Thus, we establish a database for evaluation, containing 11840 keystroke samples.

3.2 Metrics

In this paper, We leverage the following metrics to evaluate the performance of the proposed attack:

Recall. Given a series of keystrokes of a key K, the recall of inferring the key K is defined as

$$Recall_K = \frac{TP}{TP + FN} \tag{5}$$

where TP, TN, FP, FN represent True Positive, True Negative, False Positive, and False Negative, respectively.

Precision. Given a series of keystrokes of a key K, the precision of inferring the key K is defined as

$$Precision_K = \frac{TP}{TP + FP} \tag{6}$$

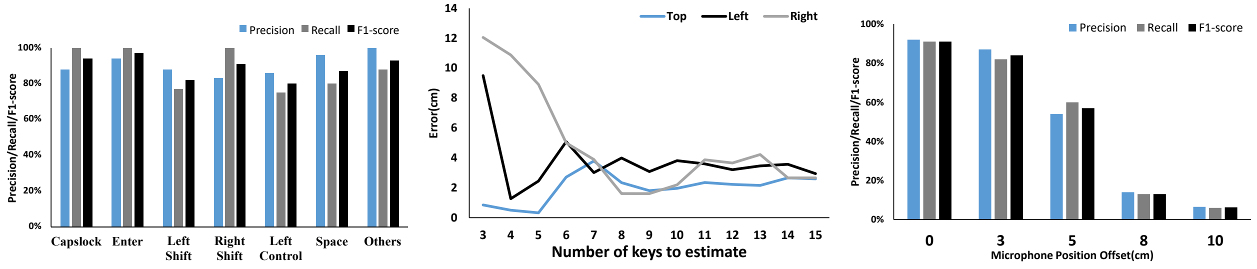
F1-score. The F1-score of inferring the key K is defined as

$$F1 - score_K = \frac{2 * Precision_K * Recall_K}{Precision_K + Recall_K} \tag{7}$$

Top-k Accuracy. Top-k Accuracy represents the probability that the correct result is included in the top k sequences ranked by confidence.

3.3 Evaluation of Microphones' Position Estimation

In this section, evaluations concentrate on environmental estimation performance, including the identification of the large-size keys and microphone position estimation. We utilize the data from the mechanical keyboard as an example to conduct experiments and calculate the average of the results under all experimental settings.



(a) Performance of recognizing large keys

(b) Error in position estimation

(c) Influence of position shift

Figure 7: Performance of CMA-ES based estimating of the microphone’s position

The first step of estimation is recognizing the large-size keys included in the collected samples. Fig. 7(a) shows the performance of identifying six large-size keys commonly used by users when typing. In general, the average precision is 91.2%, and the average recall is 89.3%. Experiments show that the algorithm can identify large-size keys with high accuracy from a series of acoustic signals. Although there are occasional cases of identification errors, these samples could be easily removed due to their excessively offset in position estimation.

Fig. 7(b) illustrates how the error between the real position and estimated position obtained by the CMA-ES varies with the number of keys used for estimation. We select three different microphone positions to conduct experiments and calculate the average error at each position. To ensure the generality of the evaluation, we select three positions, i.e., the left, right, and top of the keyboard, then perform five estimates at each position. Experiments show that the error will gradually shrink to around $3cm$ as the number of keys used for estimation increases to more than 6. Results demonstrate that we can realize the estimation of the relative position using only a single smartphone and a small number of keystroke signals, while no prior knowledge about the keystroke labels is required.

We further conduct experiments to evaluate the influence of the offset in the position estimation. We select 5 microphone placement positions at a distance of $0cm$, $3cm$, $5cm$, $8cm$, and $10cm$ from the original position where we collect the training set. Then an evaluation is performed by predicting unknown keystrokes collected from the four positions we select. The final average result is shown in Fig. 7(c). Compared to the result of no offset in the microphone position, the F1-score drop is less than 8% when offset equals $3cm$. This indicates that the estimation error of about 3cm is acceptable in the microphone position estimation and keystroke prediction.

3.4 Evaluation in Unknown Environment

One of our most important contributions is that the proposed scheme can adapt to various environments and make predictions about the collected keystrokes without prior knowledge. In practical situations, we usually do not have information about the victims and the keyboard before the attack. Therefore, we hope the proposed scheme can predict keystrokes from unfamiliar victims and keyboards.

In order to verify the robustness of the proposed scheme, we choose 6 users to perform evaluations. None of the data of each user participating in the experiment is added to the training set. To prove the effectiveness of the small training set, we predict the keystrokes of a user using the data of one different user each time and average the results. In experiments, the participants use the same keyboard. Fig. 8 displays the performance of the system in predicting keystrokes from different unknown victims. Experiments show that the average accuracy of the system’s Top-1, Top-3, and Top-5 can reach 71.24%, 84.05%, and 91.52%, respectively.

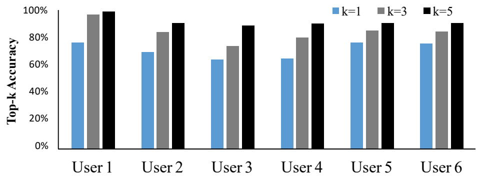


Figure 8: Performance on predicting keystrokes from unknown victims

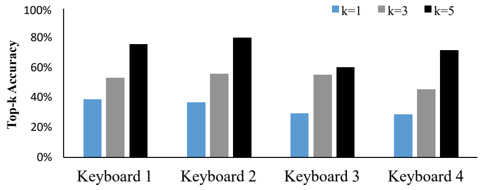
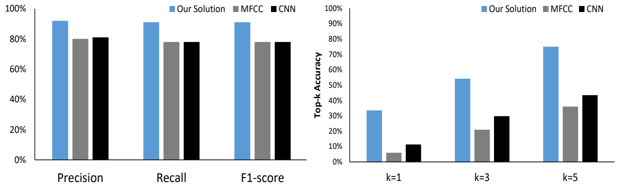


Figure 9: Performance on predicting keystrokes from unknown keyboards

We then evaluate the precision of keystroke signals from unfamiliar keyboards. We select data from 4 keyboards for evaluation. Keyboard 1 and 2 are mechanical keyboards, keyboard 3 is a membrane keyboard, and keyboard 4 is a laptop keyboard. In experiments, the keystrokes used for testing are always from unknown victims and keyboards, while a small training set, including keystroke data of one user, is collected in the simulated environments. For each keyboard, we repeat experiments with different users and average the results. Fig. 9 illustrates the performance of the proposed scheme in predicting keystrokes from different unknown keyboards. The average accuracy of Top-1, Top-3, and Top-5 is 33.74%, 52.53%, and 72.25%. Experiments demonstrate that

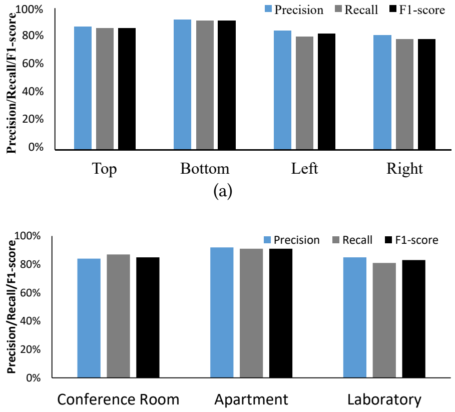
Table 1: Prediction of Words

Words	degree	doctor	medicine	university	London	that	year	promotion	campaign	could
Length	6	6	8	10	6	4	4	9	8	5
Position in results (User1)	1	5	1	1	1	1	3	1	1	5
Position in results (User2)	1	3	1	1	1	1	1	1	1	1
Position in results (User3)	1	5	1	1	1	4	1	1	2	1
Average position	1.00	4.33	1.00	1.00	1.00	2.00	1.67	1.00	1.33	2.33
Words	assistant	succeed	bring	misfortune	across	country	through	join	nothing	proceed
Length	9	7	5	10	6	7	7	4	7	7
Position in results (User1)	1	4	1	1	2	1	6	2	1	2
Position in results (User2)	1	4	1	1	1	4	4	6	1	1
Position in results (User3)	1	3	1	1	1	5	3	2	1	1
Average position	1.00	3.67	1.00	1.00	1.33	3.33	4.33	3.33	1.00	1.33



(a) (b)

Figure 10: (a) Performance of different schemes on inferring keystrokes (b) Performance of different methods on predicting keystrokes from unknown environments



(a)

(b)

Figure 11: (a) Performance in position variation (b) Performance in environment variation

we can make effective predictions for different kinds of keyboards in unfamiliar environments.

3.5 Comparison with the State-of-the-arts

In this section, we compare our method with two related works. In the work of [10], the authors utilize Convolutional Neural Networks (CNN) to recognize keystrokes with more than one mobile phone as recording devices. In [7], the authors extract MFCC of the acoustic signals to classify keystrokes. We compare the performance of the proposed scheme and these two methods on keystroke recognition through experiments in two scenarios.

Firstly, we evaluate the performance in identifying keystrokes when the type of keyboards and the microphones' positions are known, which is the situation usually considered in previous work. We conduct experiments on all the collected data sets and give the results by 10-fold cross-validation. Fig. 10(a) illustrates the overall performance of the proposed scheme and the other two systems, which is the average result in identifying all keystrokes we can recognize. Experiments show that the average precision, recall and F1-score can reach 92.0%, 91.4% and 91.2%, respectively, which are better than that of the work in [7] and [10]. Results demonstrate that the proposed scheme can obtain more accurate prediction in the presence of a training set and is more suitable to fulfill the fine-grained task of keystroke recognition.

We also compare the performance on predicting keystrokes in unknown environments. Four different keyboards are employed to input and only one smartphone is utilized to record keystrokes at an unknown location. As shown in Fig. 10(b), the proposed scheme largely outperforms the other two schemes, thanks to the effective environment estimation scheme and robust feature selection. Although Top-1 Accuracy decreases when facing unfamiliar victims and keyboards, we can still get a high Top-5 Accuracy, which is productive for predicting words typed in an unfamiliar environment.

3.6 Impact of Experimental Components

Position variation Fig. 11(a) illustrates the performance of the system when the microphones are in 4 different positions, including the top, bottom, left, and right of the keyboard, respectively. At each location, we collect keystrokes on the mechanical keyboard from three different users. The prediction of keystrokes can reach 86.3%, 91.5%, 81.9%, and 79.5% of F1-score, respectively. It can be seen that when the microphones are located on the left or right sides, the F1-score is lower than that on the upper or lower sides, because there are usually more keys on the keyboard that have the same or close TDoA when the microphones are on the left or right sides of the keyboard.

Environment variation To evaluate the performance of the scheme in different scenarios, we collect unlabeled data to be predicted from three different environments. In the rooms with different noises, the F1-score of keystroke prediction can reach 85.3%, 91.7%, and 83.5% respectively. As shown in Fig. 11(b), changes in environments have a slight impact on keystroke prediction, but the proposed scheme still shows stability and accuracy in various circumstances. This benefits from the robust features we choose, revealing the differences between the two channels.

Since the proposed scheme extracts two robust features, minimizes the impact of tone and frequency on the classification results, and considers solutions to unfamiliar environments, we can get accurate results under different positions and environments.

3.7 Performance of Word Prediction

To show the accuracy in predicting words, we select 20 words of various lengths and letters from the dictionary and predict the words in 3 different unknown environments. The prediction results are shown in Table 1, in which the position value of 1 indicates that the word has the highest confidence.

For the words used for testing, the average Top-k accuracy can reach 66.67%, 81.67%, and 96.67% for $k = 1, 3, 5$. Experiments show that the top-5 sequence obtained in the prediction is highly effective as input to predict words under unfamiliar conditions. These experiments are conducted without prior knowledge and the results accurately restore the victim's input, demonstrating the higher practical significance of the proposed scheme. Consequently, such side-channel attacks pose more practical and significant threats to user privacy than imagined, while smart devices are highly popular nowadays.

4 RELATED WORK

4.1 Keystroke Prediction Based on Sensors and Networks

Some early achievements of side-channel attacks on keystrokes were mainly realized through motion sensors[3, 6, 13, 14, 16, 17, 20, 21, 23, 28]. Researchers collected motion signals from the victim's smartwatch during the input to obtain the movements, thereby judging keystrokes made by the victim. However, such methods may be affected by unstable human actions, causing difficulties to correspond one-to-one between body movements and keystrokes. Simultaneously, most of these attacks required victims to carry or wear corresponding detection equipment, which may make attacks easier to prevent. Some works utilized WiFi or cellular network signals to carry out side-channel attacks on keystrokes[2, 8, 19, 24, 25, 29], utilizing Channel State Information (CSI) to detect the victim's input. However, they could only apply to situations where wireless signals and transceivers exist. Compared with the above methods, the proposed scheme does not rely on complicated or expensive equipment and can also adapt to various input scenarios.

4.2 Keystroke Prediction Based on Acoustic Signal

Currently, there have been some works that conducted side-channel attacks on the keyboard of computers or mobile phones through acoustic signals[4, 5, 12, 15, 27, 30]. The work in [15] calculated TDoA of the collected two acoustic channels to reduce the label range of each keystroke to 1-3 candidates. They then utilized the k-means algorithm to cluster the keystrokes according to their tones and determined their labels by the average TDoA of each category. The works in [5, 27] utilized TDoA to attack the keyboards of mobile phones and computers, respectively. However, in practice, these works did not consider the impact of environmental changes and required some prior knowledge to make predictions. The works in [4, 12, 30] extracted frequency-domain features for classification. Nevertheless, the performance of these methods may be affected by changes in keyboard types and victims. The works in [7] considered eavesdropping on the keystrokes of an unknown keyboard, but their scheme could only record the keystrokes during the phone call and assumed that the position of the recording device was constant relative to the keyboard. On the other hand, they only chose the MFCC to identify keystrokes, reducing the robustness of the system.

Some other works trained neural networks for keystroke eavesdropping [10]. In the work of [10], researchers predicted keystrokes from 2-16 smartphone arrays, each of which were equipped with microphones. They used CNN to extract features to distinguishing keystrokes and Long Short-Term Memory (LSTM) to improve accuracy by considering the relationship between different keystrokes. However, before the attack, the neural network needs sufficient data for training to distinguish keystrokes and build a dictionary.

The work in [18] identified the user's keystrokes by emitting ultrasonic waves and receiving reflection signals. When the victim's finger moved, the reflection signal received by the microphone generated a special waveform due to the Doppler effect, representing the direction of movement. Then, they locked specific keys by comparing the energy attenuation between the reflected signal and the original one. However, the victim's finger movement was not always stable, which may impact the prediction in the real environment.

Unlike existing works, the proposed scheme predicts keystrokes based on the keystroke location, so it is rarely affected by the victim users and keyboards. Meanwhile, we consider ways to cope with microphone position and keyboard type variation, ensuring the proposed scheme fits unfamiliar environments. Furthermore, the robustness and adaptability ensured by features and the algorithm make the attack accomplish without training with large amounts of data.

5 CONCLUSION

This paper proposes a side-channel attack scheme on keyboard input using acoustic signals from microphones in a smart device. An efficient keyboard type estimation and microphone position estimation scheme are designed to adapt the side-channel attack to unfamiliar environments. Furthermore, two robust features are extracted to alleviate the impact of various environmental changes while fulfilling the fine-grained keystroke classification. The experimental results prove that the proposed scheme can obtain more robust and accurate effects than the existing works in terms of keystroke prediction. Even if the attacker has no prior knowledge, the side-channel attack scheme can still greatly threaten the victim's input and is challenging to prevent. Moving forward, we are interested in further optimizing the performance of keystroke predictions between different types of keyboards so that attackers can eavesdrop on keyboards with different sizes, layouts, and mechanical structures without prior knowledge.

REFERENCES

[1] ISO 9613. 1993. Acoustics-Attenuation of sound during propagation outdoors – Part 1: Calculation of the absorption of sound by the atmosphere.

[2] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using wifi signals. In *Proceedings of the 21st annual international conference on mobile computing and networking*. 90–102.

[3] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2017. Recognizing keystrokes using WiFi devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1175–1190.

[4] Dmitri Asonov and Rakesh Agrawal. 2004. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*. IEEE, 3–11.

[5] Yigael Berger, Avishai Wool, and Arie Yeredor. 2006. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the 13th ACM conference on Computer and communications security*. 245–254.

[6] Liang Cai and Hao Chen. 2011. TouchLogger: inferring keystrokes on touch screen from smartphone motion. In *Proceedings of the 6th USENIX conference on Hot topics in security*. USENIX Association, 9–9.

[7] Stefano Cecconello, Alberto Compagno, Mauro Conti, Daniele Lain, and Gene Tsudik. 2019. Skype & Type: Keyboard Eavesdropping in Voice-over-IP. *ACM Transactions on Privacy and Security (TOPS)* 22, 4 (2019), 1–34.

[8] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. Tracking keystrokes using wireless signals. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 31–44.

[9] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[10] Tyler Giallanza, Travis Siems, Elena Smith, Erik Gabrielsen, Ian Johnson, Mitchell A Thornton, and Eric C Larson. 2019. Keyboard Snooping from Mobile Phone Arrays with Mixed Convolutional and Recurrent Neural Networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–22.

[11] Nikolaus Hansen, Nikolaus Hansen, Andreas Ostermeier, and Andreas Ostermeier. 1997. Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The ...-CMA-ES.

[12] Andrew Kelly. 2010. Cracking passwords using keyboard acoustics and language modeling. *University of Edinburgh, http://citeseerx. ist. psu. edu/viewdoc/download* (2010), 54.

[13] Mengyuan Li, Yan Meng, Junyi Liu, Haojin Zhu, Xiaohui Liang, Yao Liu, and Na Ruan. 2016. When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1068–1079.

[14] Kang Ling, Yuntang Liu, Ke Sun, Wei Wang, Lei Xie, and Qing Gu. 2020. SpiderMon: Towards Using Cell Towers as Illuminating Sources for Keystroke Monitoring. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 666–675.

[15] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. 2015. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 142–154.

[16] Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li, and Kehuan Zhang. 2015. When good becomes evil: Keystroke inference with smartwatch. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1273–1285.

[17] Yang Liu and Zhenjiang Li. 2018. aleak: Privacy leakage through context-free wearable side-channel. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1232–1240.

[18] Li Lu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Xiangyu Xu, Guangtao Xue, and Minglu Li. 2019. KeyLiSterber: Inferring keystrokes on QWERTY keyboard of touch screen through acoustic signals. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 775–783.

[19] Anindya Maiti, Murtuza Jadliwala, Jibo He, and Igor Bilogrevic. 2015. (Smart) watch your taps: side-channel keystroke inference attacks using smartwatches. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*. 27–30.

[20] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM conference on Computer and communications security*. 551–562.

[21] Maryam Mehrnezhad, Ehsan Toreini, Siamak F Shahandashti, and Feng Hao. 2018. Stealing PINs via mobile sensors: actual risk versus user perception. *International Journal of Information Security* 17, 3 (2018), 291–313.

[22] A Norman Mortensen and GL Johnson. 1988. A power system digital harmonic analyzer. *IEEE transactions on instrumentation and measurement* 37, 4 (1988), 537–540.

[23] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. Accessory: password inference using accelerometers on smartphones. In *proceedings of the twelfth workshop on mobile computing systems & applications*. 1–6.

[24] Sougata Sen, Karan Grover, Vigneshwaran Subbaraju, and Archan Misra. 2017. Inferring smartphone keypress via smartwatch inertial sensing. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 685–690.

[25] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. 2014. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 14–27.

[26] Norbert Wiener et al. 1930. Generalized harmonic analysis. *Acta mathematica* 55 (1930), 117–258.

[27] Zhen Xiao, Tao Chen, Yang Liu, and Zhenjiang Li. 2020. Mobile Phones Know Your Keystrokes through the Sounds from Finger's Tapping on the Screen. In *40th IEEE International Conference on Distributed Computing Systems*. IEEE.

[28] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. 113–124.

[29] Jie Zhang, Xiaolong Zheng, Zhanyong Tang, Tianzhang Xing, Xiaojiang Chen, Dingyi Fang, Rong Li, Xiaoqing Gong, and Feng Chen. 2016. Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality. *Mobile Information Systems* 2016 (2016).

[30] Li Zhuang, Feng Zhou, and J Doug Tygar. 2009. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)* 13, 1 (2009), 1–26.