# KeystrokeSniffer: An Off-the-Shelf Smartphone Can Eavesdrop on Your Privacy from Anywhere

Jinyang Huang, Jia-Xuan Bai, Xiang Zhang*, Zhi Liu, Yuanhao Feng, Jianchun Liu, Xiao Sun, Mianxiong Dong, and Meng Li*

***Abstract*—With mobile phones becoming increasingly prevalent and embedding high-quality microphones, attackers have the ability to employ these microphones to eavesdrop user's keyboard input. However, existing work usually assumes that keystroke eavesdropping is performed against known environments and victims, which inevitably makes attack systems lack generalization. To reveal the real threat of the acoustic signal-based attack strategy, this paper proposes a keystroke eavesdropping algorithm called *KeystrokeSniffer*, which is robust to unknown input environments and unknown victims. In particular, to mimic the real input environment of victims, an environment estimation algorithm is first designed by extracting the timbre-related characteristics to predict the keyboard type and identifying large-size key data from collected unlabeled samples to estimate the 3D microphone coordinates. Then, by imitating unknown environments and victim data, this algorithm achieves effective keystroke eavesdropping with a small training set. By further considering the commonalities of different keystroke habits, a robust feature extraction method that reflects the keystroke location is adopted to reduce the impact of individual input habits. Extensive experimental results using various commodity smartphones indicate that the scheme is capable of predicting keyboard input accurately under different unknown scenarios. Specifically, even when both the victims and keyboards are unknown, *KeystrokeSniffer* can still achieve high Top-5 accuracy, reaching $79.5\%$ in predicting keystrokes and $96.7\%$ in predicting meaningful words, which demonstrates *KeystrokeSniffer* has excellent generalization capabilities. By setting different parameter values of various impact factors, *e.g.*, noise and hand length factors, the strong robustness of the system is demonstrated, which proves that *KeystrokeSniffer* can violate privacy in real situations.**

***Index Terms*—Acoustic sensor, keyboard snooping, side channel attack, environment robustness, input habit robustness.**

## I. Introduction

### A. Backgrounds and Motivations

HIGH-PRECISION microphones are now widely equipped in various off-the-shelf smartphones for human-computer interaction [1]. While they provide users with convenience, they also pose significant risks of privacy leakage. Both user behavior and privacy with a specific sound can be threatened by side-channel attacks based on acoustic signals. Among such privacy types, input via the keyboard, as one of the main ways for users to operate electronic devices, usually involves a variety of privacy [2], including passwords and plaintext content [3]. As a result, acoustic eavesdropping on keyboard input content has become a worthy focus in current research.

Some pioneer works focused on the distinctive timbre of different keystrokes and manually extracted frequency domain features from the acoustic signal for classification [4]–[7]. However, most frequency-domain features that reflect keystroke timbre usually change with the typing habits and the keyboard type. These factors inevitably leads to a lack of robustness in these keystroke eavesdropping systems. Giallanza et al. [8] used microphone arrays and deep neural networks to extract features from keystroke sounds for the keystroke classification task, which tried to solve the robustness problems caused by manually extracted features. However, this scheme required multiple smartphones to collect data to achieve satisfactory performance and relied on large-scale training data, which made it difficult to implement.

Another crucial problem of the above works is that they can only perform keystroke eavesdropping in some known input environment and build models from the existing training set. However, in practical situations, attackers who initiate side-channel attacks are often faced with unknown input environments and a lack of training samples. Furthermore, unknown environments significantly downgrade the performance of these works. Although Cecconello et al. [9] presented an algorithm for keystroke eavesdropping when the keyboard type is unknown, it can only accommodate a limited set of keyboard candidates and did not consider the case when the relative position of the microphone to the keyboard was changed. A keystroke eavesdropping algorithm that is robust to unknown input environments is a research hotspot in both academia and industry.

### B. Challenges and Contributions

Two major challenges need to be formally addressed before realizing a practical acoustic signal-based side-channel attack.

- ***Unfamiliar input environments:*** In eavesdropping the user keystroke process, sound-receiving devices often face unfamiliar environments. In particular, the placement of the microphone is random, and the attacked keyboard type is unknown, which results in the performance downgrade of state-of-the-art methods that rely on fixed positions of sound-receiving equipment and known keystroke types. Therefore, it is challenging to accurately identify a user's keystrokes in an unfamiliar environment.

Jinyang Huang, Xiao Sun, and Meng Li, are with the Anhui Province Key Laboratory of Affective Computing and Advanced Intelligence Machine and School of Computer and Information, Hefei University of Technology, Hefei, 230601, China.

Jia-Xuan Bai, and Xiang Zhang, are with CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei, 230026, China.

Zhi Liu, Department of Computer and Network Engineering, The University of Electro-Communications, Tokyo, 1828585, Japan.

Yuanhao Feng, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, 100872, China.

Jianchun Liu, School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230026, China.

Mianxiong Dong, Department of Sciences and Informatics, Muroran Institute of Technology, 0508585, Japan.

*Corresponding author: Xiang Zhang, Meng Li (Email: zhangxiang@ieee.org, mengli@hfut.edu.cn)

- ***Different keystroke habits:*** Different subjects have different keystroke habits. Some people hit keys lightly and slowly, while others hit keys hard and fast. This results in significant differences in the sound signals generated by different subjects hitting the same keys. Thus, how to deal with the distinct sound samples of the same key caused by different keystroke habits to achieve accurate keystroke recognition becomes a question worthy of consideration.

To address the existing challenges in related works, this paper puts forward a side-channel attack to accomplish keystroke snooping when keyboards and victims are both unknown. Instead of collecting a large amount of keystroke data from different victims in different input environments for fitting or for adversarial learning to filter out environmental and input habit effects, which take a long time to collect data, have a huge amount of calculations, and thus are difficult to implement in real scenarios, the proposed innovative approach uses several large-size keys with obvious timbre distinctions to identify different environment parameters in an unfamiliar environment without pre-training, and then scientifically **transform the keystroke recognition problem with the unknown environment into the keystroke recognition problem with the known environment**. Then, a robust feature extraction method that reflects the keystroke location is adopted to reduce the impact of individual input habits since no matter what victim input habits are, the same keystrokes have the same position information. Compared with state-of-the-art keystroke snooping algorithms that need to collect environmental information and to know victim input habits, the proposed system *KeystrokeSniffer* is more practical and highly efficient because it does not require collecting environmental information and obtaining victim input habits.

Specifically, the proposed scheme employs dual microphones equipped on a single commercial smartphone to collect sounds caused by keystrokes. To deal with unknown input environments, we present a novel scheme that extracts the timbre-related characteristics and identifies large-size key data to estimate the unknown input environment, including the keyboard type and the relative position of microphones in the three-dimensional space to the keyboard. After learning this originally unknown information, the attacker can mimic the victim's input environment and collect the training set in that environment, which can significantly enhance the adaptability and robustness of the keystroke eavesdropping algorithm to the unknown environment. To ensure the proposed scheme's robustness to unknown victims, we extract robust features from the acoustic signal, including Time Difference of Arrival (TDoA) and Power Spectral Density (PSD) that fluctuate primarily with keystroke positions and are less affected by the individual input habits of victims.

In brief, our contributions can be broadly summarized as follows:

- To the best of our knowledge, this paper is the first attempt to eavesdrop on the keystrokes from unfamiliar environments without victims' ground truth data.
- A novel unknown environment estimation algorithm is proposed based on special timbre features of large-size keys, which overcomes the problems posed by microphone location changes and a lack of training data for keyboard type.
- By considering the commonalities of different victim keystroke habits, that the same keystrokes have the same position information, we propose a robust algorithm that incorporates effective feature extraction and model training to better reveal the keystroke positions to further realize the identification of keystrokes from unknown victims.
- We implement the proposed approach on a single commodity smartphone (including different smartphone types). Extensive experimental results demonstrate that the proposed scheme outperforms state-of-the-art works in terms of accuracy and robustness.

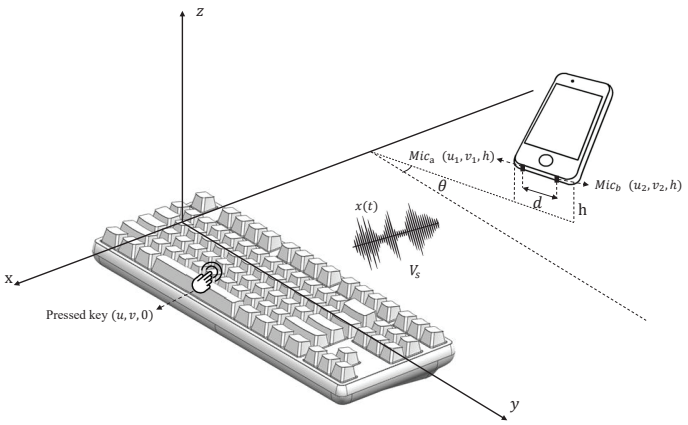



**Fig. 1** The attack scenario of the system.

The rest of the paper is organized as follows. Sec. II discusses numerous related studies. Then, we describe the *KeystrokeSniffer* system design in Sec. III. Implementation, evaluation, and the impacts of various factors on *KeystrokeSniffer* performance are presented in Sec. IV. Finally, we conclude our work in Sec. V.

## II. Related Work

### A. Keystroke Prediction Based on Sensors or Wireless Signal

Some of the earliest successes of side-channel attacks on keystrokes were mostly accomplished via motion sensors [10], [11]. These researchers obtained the keystroke motions by collecting signals from the victim's smartwatch during the input. Thus, the victim's keystrokes can be effectively assessed. However, the efficacy of these methods can be undermined by unpredictable human motions, which leads to challenges in achieving precise correspondence between body movements and keystrokes [12].

Meanwhile, the majority of these attacks required victims to carry or wear specialized detection equipment, which can make these attacks obtrusive. Several works exploited WiFi or cellular network signals to conduct side-channel attacks on keystrokes [13]–[15], which detected the victim's input via Channel State Information (CSI) [16]. However, they may be applicable exclusively in scenarios involving wireless communications and transceivers [17]. Furthermore, different input environments may significantly change wireless signals caused by the same keystrokes, which inevitably results in the performance downgrade of these wireless-based methods [18]. Compared with the prior methods, the proposed scheme does not require complex or expensive equipment and is more adaptable to a variety of input conditions, which makes the proposed scheme more practical.

### B. Keystroke Prediction Based on Acoustic Signal

Several recent studies have employed acoustic signals to carry out side-channel attacks on computer or mobile phone keyboards [4], [6], [7], [19], [20]. In particular, the research presented in [4] focused on calculating the TDoA of two collected auditory channels to narrow down the possible label range of each keystroke to 1-3 choices. The keystrokes were then clustered based on their timbre using the K-means algorithm, and the corresponding labels were assigned according to the average TDoA within each category. Similarly, the use of TDoA was also observed in [6] and [7], where mobile phone and computer keyboards were attacked targets. Several other studies have employed neural networks for keyboard eavesdropping. For instance, researchers in [8] predicted keystrokes using acoustic data collected from two to sixteen smartphone arrays. By considering the temporal relationships between keystrokes, they utilized Convolutional Neural Networks (CNN) to extract features for keystroke recognition and Long Short-Term Memory (LSTM) networks to enhance recognition accuracy. However, no matter whether TDoA-based or neural network-based algorithms, they collectively assumed that the victim's input occurs in a known environment. Unfortunately, unknown environments are common in daily keystroke eavesdropping scenarios, and their performance is significantly downgraded when facing unknown environments.

Although the research in [9] explored keystroke eavesdropping on an unfamiliar keyboard, their approach was limited to capturing keystrokes during phone calls and relied on the assumption of a constant position of the recording device relative to the keyboard. Thus, this system may lack robustness due to its reliance on simplistic feature extraction. By using a single smartphone placed within the range covered by a microphone and speaker, UltraSnoop [21] proposed a placement-agnostic scheme to infer the user's input. Although it can infer the relative position between the smartphone and the keyboard with satisfactory recognition accuracy, it cannot distinguish between different input keyboard types, which inevitably causes a performance downgrade when the keyboard type changes.

On the other hand, existing state-of-the-art methods usually ignored the differences in keystroke habits of different individuals, and they tried their best to ensure the high similarity of keystroke postures in experiments. Once faced with individuals with different keystroke habits, their performance will drop inevitably. For instance, the pioneer work in [7] extracted frequency-domain features for better performance. However, the composition of the frequency domain varied not only among different keystrokes, but also in sound intensity and reflections due to different input habits. Besides, the work in [22] produced ultrasonic waves and received corresponding reflection signals to identify user keystrokes. As the victim's finger moved, the microphone captured the reflected signal, which formed a distinctive waveform that portrayed the movement direction attributable to the Doppler effect. Then, by comparing the energy of the reflected signal to that of the transmitted signals, specific keys could be discerned. However, the victim's finger movement was not always consistent, which potentially undermined the accuracy of keystroke prediction in practical scenarios.

In contrast to prior studies, we first explore techniques to accommodate variations in microphone positions and keyboard types, which ensures the effectiveness of the proposed scheme even in unfamiliar environments. Then, by leveraging the spatial location features to distinct keystrokes, the system susceptibility to alterations caused by victim users or keyboards is significantly minimized. Furthermore, compared with neural network methods that need a lot of data for training, the adaptability of proposed algorithms and features enables this attack to be effectively performed without extensive training on vast data volumes.

**TABLE I** Meaning of symbols in the keystroke eavesdrop environment shown in Fig. 1.

| Symbols | Meanings |
|---|---|
| $Mic_a, Mic_b$ | Two microphones in the smartphone. |
| $\boldsymbol{x}(t)$ | Raw signal from keystrokes in time domain. |
| $\boldsymbol{X}(\omega)$ | Raw signal from keystrokes in frequency domain. |
| $d_1, d_2$ | Distances from the two microphones to the sound source point, respectively. |
| $V_s$ | Velocity magnitude of sound in the air. |
| $h$ | Height of the microphone's plane relative to the keyboard's plane. |
| $\theta$ | The angle between the projection of the line connecting $Mic_a$ and $Mic_b$ in the xOy plane and the y-axis. |
| $d$ | Distance between two microphones. |
| $(u, v, 0)$ | Coordinate of the pressed key. |
| $(u_1, v_1, h)$ | Coordinate of $Mic_a$. |
| $(u_2, v_2, h)$ | Coordinate of $Mic_b$. |

## III. System Design

This section presents the proposed scheme design, which utilizes the microphones of a single off-the-shelf smartphone to perform side-channel attacks on keyboard input when both keyboards and victims are unknown.

### A. Attack Scenario

In order to restore the real scene attack, attackers conduct a side-channel attack on keyboard input with unknown input environments and unknown victims, which only utilizes two microphones in the smartphone, denoted as $Mic_a$ and $Mic_b$, respectively. The unknown input environment implies that the keyboard type is unidentified, and the microphones' positions for capturing acoustic signals may vary within the three-dimensional space surrounding the keyboard. Additionally, unknown victims mean that victims' input habits are uncertain. Therefore, the attack lacks labeled data from the input environments and the corresponding victims for training purposes. Besides, throughout the input process, the acoustic signal may be accompanied by unpredictable ambient noise. To demonstrate the effectiveness of *KeystrokeSniffer* for different type keyboards, the proposed scheme is implemented and evaluated on three primary types

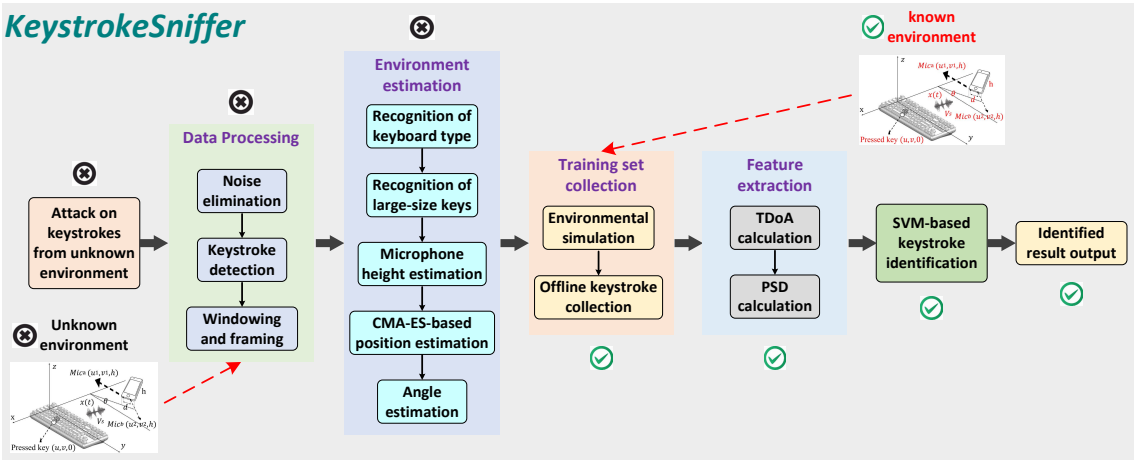



**Fig. 2** Illustration of the workflow to predict the unknown victim keystrokes with acoustic signals in unknown environments.

of keyboards, *i.e.*, mechanical keyboards, membrane keyboards, and laptop keyboards. For simplification of processing, we assume that keyboard layouts of the same type are substantially equivalent.

Fig. 1 presents an illustrative experimental scenario where the $Q$ key on the keyboard serves as the coordinate origin in this paper. Tab. I shows the specific meanings of certain symbols employed in this attack experimental scenario. The $x$, $y$, and $z$ axes of the coordinate system are defined to be perpendicular to the row direction of the keys, parallel to the row direction of the keys, and perpendicular to the plane of the keyboard, respectively. In this attack experimental scenario, the smartphone is placed in an upward position relative to the keyboard and in a plane above the keyboard plane.

The objective of the proposed scheme is to eavesdrop on the keystrokes of 32 keys on the keyboard, which comprise 26 alphabetic keys and 6 special keys, within the mentioned conditions. These special keys, collectively referred to as large-size keys, are larger in size compared to the alphabetic keys and encompass the left Tab, CapsLock, left Shift, left Control, Space, and Enter. It is worth noting that any attackers can effortlessly apply the same principle to expand the attack to additional keys.

### *B. System Overview*

The workflow of the proposed scheme is illustrated in Fig. 2. Specifically, *KeystrokeSniffer* mainly includes the following five parts: data processing, environment estimation, training set collection, feature extraction, and keystroke identification. The innovative insight of *KeystrokeSniffer* is using several large-size keys with obvious timbre distinctions to identify different environment parameters in an unfamiliar environment without pre-training and then scientifically **transform the keystroke recognition problem with an unknown environment into the keystroke recognition problem with a known environment**. The data processing and environment estimation parts are employed to recognize the unknown environment, while the training set collection, feature extraction, and keystroke identification parts are used to identify keystrokes with different input habits in the known environment. This section provides a detailed description of these parts. The subsequent sections present a specific implementation of side-channel attacks on keystrokes.

**Data Pre-processing:** Firstly, the acoustic signals of keystrokes are captured by using two microphones embedded in a single smartphone, and these microphones are positioned in the 3D space surrounding the keyboard. The initial step for the attackers involves detecting the keystrokes within the acquired signal and subsequently segmenting the collected acoustic signals into individual segments to make sure each segment encompasses a single keystroke. Furthermore, an adaptive spectral subtraction technique is employed to extract the acoustic signals associated with the keystrokes from the original signal.

**Environment Estimation and Training Set Collection:** Subsequently, two important pieces of information are extracted by attackers from the unlabeled acoustic data. Firstly, since keystrokes on different types of keyboards have significant timbre differences, the type of input keyboards is determined by extracting the Mel Frequency Cepstrum Coefficients (MFCC), which can effectively represent the timbre characteristics of the keystrokes. Secondly, by further utilizing collected unlabeled samples, the 3D coordinates of the microphones are accurately estimated by attackers.

Upon acquiring two important pieces of information, attackers can mimic the real input environment by deploying the same type of keyboard used by the victim and adjusting the microphone locations to match the estimated relative positions. Consequently, a targeted training set can be constructed for model training within this specific scenario. By leveraging the environment estimation algorithm and the offline data collection strategy, the attacker can construct targeted training sets for the corresponding environment and further enhance the system's robustness to unknown input environments.

**Feature Extraction and Keystroke Prediction:** Two robust features from the acoustic signals are extracted to identify keystrokes. Specifically, the principles of acoustic attenuation-related feature PSD and the signal propagation path-related feature TDoA are utilized to capture the location information of keystrokes. Since Support Vector Machine (SVM) has strong scene generalization and can still maintain good learning results even with a relatively small number of samples, these acquired features are then employed to train an SVM model for keystroke prediction.

Totally we propose a novel offline technique for keystroke prediction. However, given that the objective of the side-channel

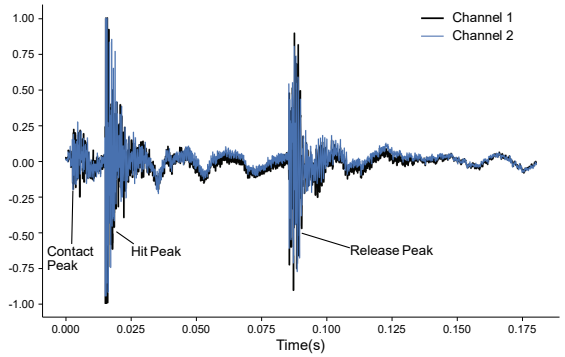



**Fig. 3** The waveform of keystroke acoustic signals.

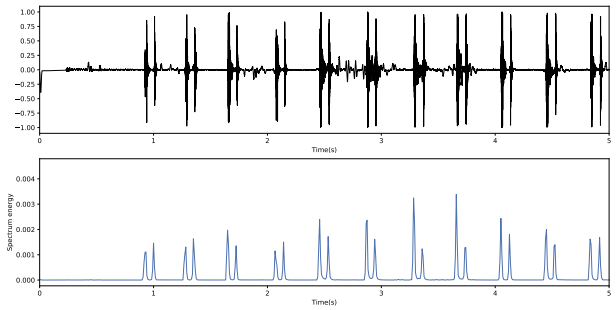



**Fig. 4** The relationship between the sound signal waveform and $E_k$.

attack is to decode the victim's input, there is no clear distinction between online and offline tactics.

### C. Data Pre-processing

This section focuses on the pre-processing of the acquired acoustic signals. Specifically, *KeystrokeSniffer* conducts data pre-processing on the collected two-channel signals through the following three steps.

*1) Keystroke Detection:* A typical keystroke signal consists of three distinct components: the contact peak, the hit peak, and the release peak. Fig. 3 illustrates these three peaks in the auditory wave of a single keystroke. To individually handle each complete keystroke, we identify the occurrence of keystrokes based on their spectral characteristics in order to separate them from the entire raw signal. Specifically, since the energy of most keystrokes is concentrated in the mid-to-high frequency region, and low-frequency sounds often overlap with environmental noises, *e.g.*, background conversation and fan noise, we calculate the spectral energy within the range of 2000-5000Hz, denoted as $E_k$. Fig. 4 demonstrates the correspondence between the keystroke signal waveform and $E_k$, and the calculated $E_k$ can effectively reflect the three peaks during keystrokes. Then, we detect the presence of contact peaks by using empirical thresholds to segment the signals. Particularly, when either of the $E_k$ values from the two channels exceeds these thresholds, we consider that the keystroke signal has entered the contact peak stage and utilize this time point as the start of a keystroke. Subsequently, based on empirical knowledge, we commence at the established starting point and divide the subsequent $180ms$ signal into individual keystroke fragments.

*2) Noise Elimination:* Considering the environment of keyboard input, the acoustic signal of keystrokes is often overlapped with ambient noise. These noises can impact the fine-grained keystroke prediction task and undermine the scheme's robustness. Given the quasi-steady state of the acoustic signal, it can be assumed that the ambient noise mixed in with keystrokes undergoes little change over a short period. Additionally, considering that keystroke recognition is a fine-grained task, our goal is to mitigate the noise impact while preserving the original signal feature. Therefore, we employ adaptive spectral subtraction [23] to eliminate these additive noises while preserving the keystroke responses. Specifically, we divide the entire signal into a sequence of $20ms$ frames for processing and select the 5 frames preceding the keystroke as noise signal samples. To avoid excessive fluctuations in the estimated noise of one frame, we average the spectra of the selected 5 frames preceding the keystroke to obtain the noise spectrum $\boldsymbol{N}(\omega)$, which can be denoted as:

$$\boldsymbol{N}(\omega) = \frac{1}{5}\sum_{i=1}^{5}\boldsymbol{X}^i(\omega), \tag{1}$$

where $\boldsymbol{N}(\omega)$ denotes the obtained noise spectrum, and $\boldsymbol{X}^i(\omega)$ represents the frequency domain representation of the $i$-th frame preceding the keystroke. By incorporating the obtained noise spectrum [24], we further use Eq. (2) to determine the keystroke signal.

$$\boldsymbol{K}(\omega) = \begin{cases} \boldsymbol{X}(\omega) - \alpha\boldsymbol{N}(\omega), \text{ if } \boldsymbol{X}(\omega) - \alpha\boldsymbol{N}(\omega) > \beta\boldsymbol{N}(\omega) \\ \beta\boldsymbol{N}(\omega), \text{otherwise}, \end{cases} \tag{2}$$

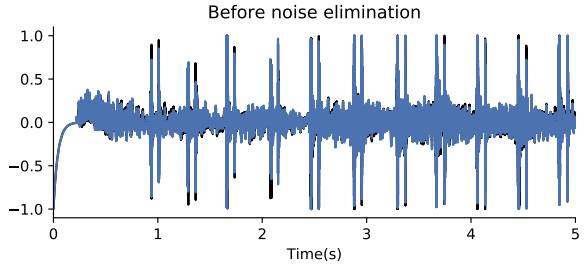
where $\boldsymbol{K}(\omega)$ represents obtained keystroke signals after noise elimination. $\alpha$ and $\beta$ are different parameters, *i.e.*, weight parameter and threshold parameter, used for signal smoothing, respectively. In particular, the size of the weight parameter $\alpha$ determines the sensitivity of the filter to noise, and if $\alpha$ is smaller, the filter is more sensitive to the noise impact and will retain more of the original signal information but may retain some noise. On the other hand, the threshold parameter $\beta$ determines the filter's tolerance for noise. If $\beta$ is small, the filter has a low tolerance for noise, and even if the noise slightly exceeds $\alpha\boldsymbol{N}(\omega)$, it will be filtered out. The value of $\alpha$ can be calculated by the following equation [24]:

$$\alpha = \begin{cases} 4 - \dfrac{3}{20}\text{SNR}, -5 \leq \text{SNR} \leq 20, \\ 5, \text{SNR} < -5, \\ 1, \text{SNR} > 20, \end{cases} \tag{3}$$

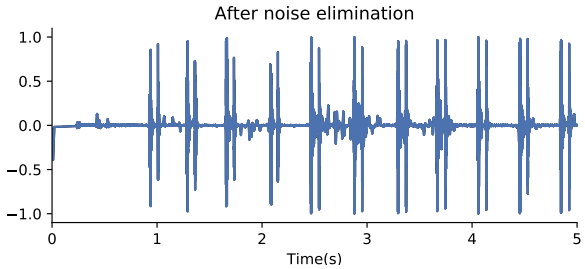
where SNR represents the Signal-to-Noise Ratio. Besides, taking into account the fluctuation characteristics of the sound signal and the tolerance to noise, based on empirical knowledge, the threshold parameter $\beta$ is set to 0.02. Additionally, the proposed scheme updates the noise spectrum dynamically by calculating the corresponding signal SNR. If the obtained SNR falls below a certain threshold, the noise spectrum $\boldsymbol{N}_{\text{new}}(\omega)$ is updated according to Eq. (4), which ensures the effectiveness of the noise elimination algorithm.

$$\boldsymbol{N}_{\text{new}}(\omega) = g \times \boldsymbol{N}(\omega) + (1-g) \times \boldsymbol{X}(\omega), \tag{4}$$

where $g$ is a weighting factor between 0 and 1, when $g$ is close to 0, the adjusted noise spectrum is closer to the original signal spectrum, which means that the noise is preserved more. Otherwise, when $g$ is close to 1, the adjusted noise spectrum




**(a)** The waveform before noise elimination.




**(b)** The waveform after noise elimination.

**Fig. 5** Illustration of the noise elimination algorithm's performance.

is closer to the original noise spectrum, which means that the noise is removed more. Similarly, to control the noise adjustment scope and adjustment speed, based on empirical knowledge, we set $g$ to 0.9. Following these steps, we normalize the values of the collected signals to the range of $[-1, 1]$. Fig. 5 illustrates an example of the noise elimination algorithm's effect in the attack scenario depicted in Fig. 1. In particular, the input environment contains various noises in this daily office scenario, *e.g.*, human conversations, machine operations, and white noise. Nevertheless, it is evident that the proposed scheme effectively extracts the keystroke signals while preserving their original characteristics.

*3) Windowing and Framing:* Subsequently, we apply the framing and windowing procedures to the obtained segments after keystroke detection and noise elimination. Due to the time-varying nature of the acoustic signal and the corresponding fundamental parameters, it exhibits non-stationary behavior. However, within short time intervals, its characteristics remain relatively constant, which indicates a quasi-steady-state process [25]. Hence, we partition the signal into frames to capture the fine-grained features that can more precisely represent the time-domain characteristics. Specifically, we divide the signal into a series of $10ms$ Hanning windows, with a shift of $2.5ms$ per step. Sec. IV-G-3 demonstrates the specific reasons for choosing this sliding window size and shift step. This framing and windowing process enables us to extract frame-based features that more accurately reflect the signal's time-domain characteristics.

### D. Environment Estimation

Although state-of-the-art methods effectively improve keystroke prediction accuracy, they frequently neglect the impact of relative location and input environment on keystroke prediction. To address these challenges, we propose an environment estimation algorithm in this section. This algorithm aims to transform an unknown input environment into a known one in the absence of ground truth training data.

*1) Keyboard Type Recognition:* As keyboards of the same type share similarities in mechanical structure, key positions, and keystroke timbre while exhibiting significant diversity across different categories, keyboard type uncertainty can pose challenges in keystroke prediction. This diversity affects the frequency domain composition of the keystroke signal and can potentially increase the intra-class distance among samples. Since MFCC can effectively denote the timbre characteristics of keystrokes, to capture the variations in keystroke timbre, we employ MFCC for keyboard categorization. MFCC analyzes the acoustic signal by utilizing the amplitude of the Fourier transform of the time-domain acoustic frame. In particular, we extract 16 MFCC from the framed and windowed data. Subsequently, considering that SVM has strong scene generalization and can still maintain good learning results even with a relatively small number of samples, we employ SVM as the classification algorithm to determine the keyboard type based on the extracted MFCC.

*2) Large-size Key Recognition:* Due to the unavailability of labeled ground truth samples and limited conditions, extracting real environmental information becomes challenging for attackers. To overcome this limitation, we leverage unlabeled keystroke signals from the victim to identify labels for a subset of special keystrokes (large-size keys). This is because the timbre of large-size keys, such as Space and Enter, inherently differs from that of other keystrokes on the keyboard. Specifically, by utilizing MFCC features, we can successfully distinguish these larger-size keys. In preparation for each type of keyboard, we gather acoustic signals of frequently used large-size keys in advance without requiring any input information from the victim. These collected signals are then used to train the SVM classifier to detect the large-size keys from unlabeled acoustic data.

*3) Microphone Coordinates Estimation:* The smartphone used for capturing keystroke sounds can be positioned in any pose within a 3-dimensional space surrounding the keyboard. As the microphone placement affects the signal path, we then design an algorithm to estimate the 3D coordinates of two microphones (denoted as $(u_1, v_1, h)$ and $(u_2, v_2, h)$). By considering the distance between the two microphones (denoted as $d$) and the tilt angle (denoted as $\theta$), the coordinates of the microphones can be determined as $(u_1, v_1, h)$ and $(u_1 + d\cos\theta, v_1 + d\sin\theta, h)$, respectively.

Based on these considerations, we sequentially estimate the four unknown parameters in the microphone coordinates to obtain their 3D coordinates. Firstly, we estimate the parameter $h$ by leveraging the recognized large-size keys. Since obtaining the exact value of the microphone height is a challenging fine-grained task, we transform the height parameter estimation problem into a classification problem based on different height planes. Each height plane corresponds to a specific distance from the keyboard plane. Considering that the vertical distance of most keystrokes is less than $3cm$ [26] and the effective pickup height of commodity microphones on mobile phones used for keystroke recognition is about $15cm$ [27], we establish six height planes at intervals of $3cm$ within the $[0, 15cm]$ range surrounding the keyboard plane. During the estimation process, we extract various location-related features, *e.g.*, energy ratio and TDoA, from the identified large-size keys to classify $h$ into the appropriate height plane ($h \in 0, 3, 6, 9, 12, 15$).

Specifically, the relationship between the energy of the keystroke signal and the distance traveled by the acoustic signal

can be expressed as:

$$E_1 d_1^2 = E_2 d_2^2 + \eta, \tag{5}$$

$$d_1 = \sqrt{(u_1 - u)^2 + (v_1 - v)^2 + h^2}, \tag{6}$$

$$d_2 = \sqrt{(u_1 + d\cos\theta - u)^2 + (v_1 + d\sin\theta - v)^2 + h^2}, \tag{7}$$

where the distances between the pressed key and the two microphones are represented as $d_1$ and $d_2$, respectively. $E_1$ and $E_2$ denote the energy of the two channel signals, while $\eta$ represents the noise variance [28]. Due to the effective denoising process, the remaining noise component in the keystroke signal can be ignored.

TDoA represents the time delay between the arrival of the acoustic signal from a keystroke at the two microphones. It is solely influenced by the propagation path of the two-channel acoustic signals. The theoretical TDoA $\Delta t_{the}$ between the two-channel signals can be represented as:

$$\Delta t_{the} = \frac{\mid d_1 - d_2 \mid}{V_s}. \tag{8}$$

where $V_s$ is the velocity magnitude of sound in the air.

Then, we combine Eq.(8) and Eq.(5) to obtain:

$$d_1 = \frac{V_s \Delta t_{the}}{\mid 1 - \sqrt{\frac{E_1}{E_2}} \mid}, \tag{9}$$

$$d_2 = \frac{V_s \Delta t_{the}}{\mid \sqrt{\frac{E_2}{E_1}} - 1 \mid}. \tag{10}$$

Considering the calculation of $d_1$ and $d_2$, Eq.(9) and Eq.(10) can be interpreted as two spherical equations in the coordinate system. Thus, the intersection of these equations determines the potential positions of the keystroke. Since several keys at different positions can determine a plane, we can obtain the keyboard plane by multiple identified large-size keys' positions, which are determined by the value of $h$ (calculated by the energy ratio and TDoA of large-size keys). Building upon the aforementioned analysis, we proceed to extract location-related features, *e.g.*, short-term energy, Cross-power Spectral Density (CSD), and TDoA from the recognized large-size keys for height plane classification. The short-term energy measures the energy of the acoustic signal within each frame. Specifically, the short-term energy of the i-th frame signal can be computed as:

$$E(i) = \sum_{k=1}^{l} \boldsymbol{x}_i^2(k), \tag{11}$$

where $l$ represents the length of each frame. Besides, TDoA can be calculated based on cross-correlation, which can be expressed as:

$$R_{\boldsymbol{x}_a,\boldsymbol{x}_b}[m] = \sum_{k=0}^{l-1-m} \boldsymbol{x}_a[k]\boldsymbol{x}_b[m+k], \tag{12}$$

where $R_{\boldsymbol{x}_a,\boldsymbol{x}_b}[m]$ is cross-correlation of $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ represent the signals received by two microphones, respectively. $m$ is the delay number of the sampling points. By calculating the cross-correlations of these two signals and finding the position corresponding to the maximum value, this position $m_{R\max}$ is the estimated value of the delay sampling points.

$m_{R\max}$ can be used to calculate the corresponding time delay between the keystroke to different microphones (TDoA). Specifically, the measured TDoA $\Delta t_{mea}$ can be expressed as:

$$\Delta t_{mea} = m_{R\max} \cdot \frac{1}{f}, \tag{13}$$

where $f$ denotes the sampling rate. Besides, CSD can be obtained by the frequency representations of cross-correlations [29]. Utilizing these features as input, an SVM model is trained for height plane classification, and the corresponding classification result is regarded as the height of the microphone's plane relative to the keyboard's plane $h$.

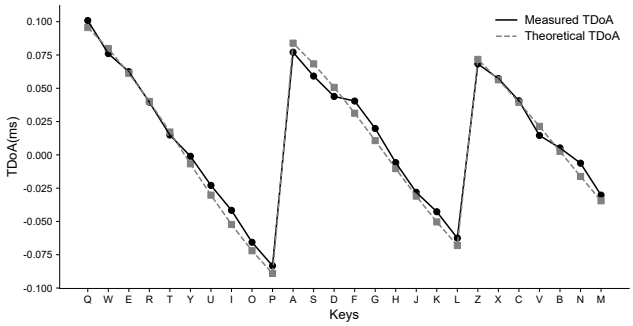



**Fig. 6** Illustration of the difference between measured TDoAs and theoretical TDoAs for 26 letters.

Once $h$ is obtained, we can utilize a TDoA-based optimization algorithm to estimate the remaining parameters $u_1$ and $v_1$ when $\theta$ is known. For each identified large-size key with a coordinate $(u, v, 0)$, its TDoA can be calculated in two ways: the theoretical values $\Delta t_{the}$ computed from the coordinates, and the measured values $\Delta t_{mea}$ obtained from acoustic signals using cross-correlation. The calculation of $\Delta t_{the}$ is based on Eq. (8), while $\Delta t_{mea}$ is estimated using Eq. (12). Different microphone placements can make distinct value differences. Fig. 6 illustrates an example of the difference between $\Delta t_{mea}$ and $\Delta t_{the}$ for 26 letters when the microphone is positioned on top of the keyboard, which demonstrates that the measured TDoA values $\Delta t_{mea}$ are similar with the theoretical TDoA values $\Delta t_{the}$, and the placements of the microphones are appropriate. Moreover, from the perspective of an attacker, microphones are best placed where the measured values $\Delta t_{mea}$ are the same as the theoretical values $\Delta t_{the}$.

The Covariance-Matrix Adaptation Evolution Strategy (CMA-ES) algorithm utilizes the evolutionary history to determine the direction of evolution, which effectively addresses the issue of local optima [30]. Microphone position estimation problems in real-world scenarios may contain multiple local optimal solutions due to multiple reflections or absorption of sound signals. Nevertheless, CMA-ES can still handle such complex problems since it can expand the search range and explore different solution regions, which significantly improve the probability of finding the global optimal solution. Therefore, to find the best microphone placement and perform the optimization, we leverage CMA-ES to optimize the associated $\Delta t_{the}$ values to approximate $\Delta t_{mea}$. Additionally, we incorporate the distance between the smartphone's two microphones as a constraint during the optimization process. Upon completion of the optimization, the parameters of $\Delta t_{mea}$ can represent the potential placements of the microphones.

To minimize estimation errors, we run CMA-ES multiple times on each identified large-size key, yielding a curve that

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2024.3424301



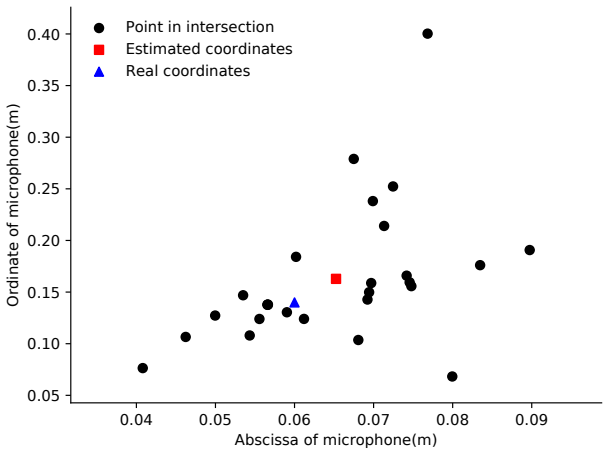



**Fig. 7** Intersection set and estimated position.

represents probable microphone coordinate results. In an ideal scenario, the intersection of these curves derived from the recognized large-size keys would pinpoint the specific location of the microphone. However, due to the presence of calculation errors, we typically obtain an intersection set denoted as $\boldsymbol{U}$, where the centroid of this set serves as the final estimate for the microphone's position. To mitigate the influence of outliers, we discard intersection points whose abscissa or ordinate deviates significantly from the mean value.

Since the value of $\theta$ is unknown, we partition its value range into 36 equally spaced intervals ($\theta \in 0°, 10°, 20°, ..., 350°$) and traverse through them. Similarly, for each $\theta$ value, we employ CMA-ES to estimate the corresponding $u_1$ and $v_1$, which generates the position set $\boldsymbol{U}(\theta)$. When the true value of $\theta$ is reached, the curves obtained from different keys tend to intersect more precisely at a single point, thereby minimizing the variance of $\boldsymbol{U}(\theta)$. Thus, we select the $\theta$ value associated with the smallest variance of $\boldsymbol{U}(\theta)$ as the final estimation. Mathematically, this can be expressed as follows:

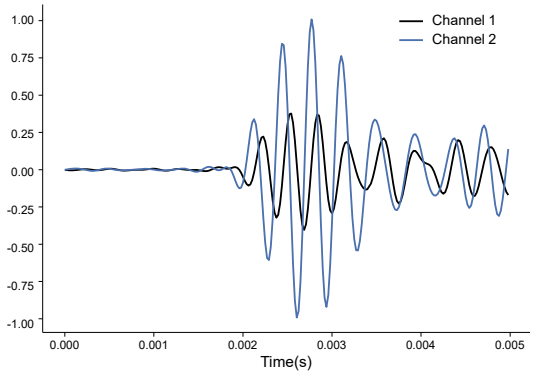
$$\theta = \arg\min\{Var(\boldsymbol{U}(\theta))\} \tag{14}$$

Subsequently, we consider the microphone coordinates corresponding to the estimated $\theta$ value as the outcome of the position estimation. Fig. 7 presents an example of the estimation results for $u_1$ and $v_1$ when a mechanical keyboard is employed and the microphone is positioned on top of the keyboard (as depicted in Fig. 1).

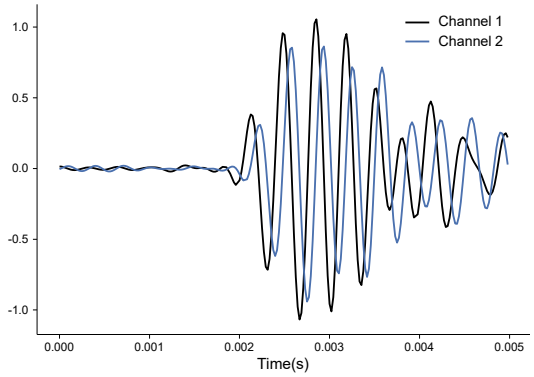
### *E. Feature Extraction & Model Training*

Once the keyboard type and microphone coordinates have been obtained, the attacker can place the smartphone at the corresponding coordinates relative to the same type of keyboard. This allows for the collection of a training set within the given input environment. The combination of environmental estimation and offline training set collection empowers the proposed scheme to be self-adaptive and independent of prior knowledge in unfamiliar environments. With the aid of this training set, attackers can extract valid features for keystroke prediction.

While the proposed scheme effectively addresses the challenge of an unknown environment, the varying input habits of different victims can still result in changes in some characteristics of the keystroke acoustic signal, *e.g.*, sound intensity and frequency. To




**(a)** The acoustic waveform of key 'Q'.




**(b)** The acoustic waveform of key 'P'.

**Fig. 8** Acoustic waveforms for different keystrokes.

tackle this, attackers need to identify robust features within the audio segment of the keystrokes. The guiding principle in feature selection is to minimize the impact of keystroke sound-related factors, *e.g.*, keystroke sound intensity and timbre. Accordingly, since no matter what victim input habits are, as long as the keystrokes are the same, the corresponding position information must be the same, we aim to disregard keystroke sound-related elements and instead concentrate on uncovering the keystroke positions. One of the chosen features for keystroke identification is TDoA, which was previously discussed for its robustness to variations and being solely influenced by the sound propagation path.

We proceed to extract additional robust features from acoustic signals to enhance fine-grained keystroke identification. Acoustic signals experience attenuation as they propagate, with their energy being dispersed in the air and absorbed by the surrounding medium. The extent of acoustic signal attenuation primarily depends on the propagation distance and its frequency. Given that the initial energy and propagation medium of the two collected signals are identical, the energy differences between them reflect the variation in distance. To quantify the disparity in sound attenuation between the two channels, we utilize PSD as one of the features, as it can capture the signal power changes with frequency. The PSD calculation is performed in the frequency band of 2000-5000Hz, where most keystroke signals are predominantly distributed. By using the Wiener–Khinchin theorem (Eq. (15)) [31], the PSD $P_X(w)$ can be mathematically represented as:

$$P_{\boldsymbol{X}}(w) = \int_{-\infty}^{\infty} R_{\boldsymbol{x}}(\tau) e^{-jw\tau} d\tau, \tag{15}$$

where $R_{\boldsymbol{x}}(\tau)$ represents for the auto-correlation of signal $x(t)$. $e$ is the base of natural logarithms and $j$ denotes the imaginary unit.

TDoA and PSD are chosen as features due to their robustness to changes in victims and keyboards. To effectively capture the distinctive attributes of the keystroke signal, we extract features from different segments of the signal. Specifically, since the first component of the keystroke signal follows a direct path from the sound source to the microphone, minimizing the impact of multipath reflections and noise, the TDoA calculation is performed on the initial $20ms$ segment. Subsequently, we compute the PSD for each frame within the 3 peaks of the keystroke signal.

Fig. 8 depicts an example of the first $5ms$ waveforms of the acoustic signals produced by the 'Q' and 'P' keys, which are obtained from the mechanical keyboard with microphones positioned on top of it. As shown in Fig. 8(a), Channel 1 exhibits a phase delay relative to Channel 2, accompanied by a reduction in signal amplitude. Conversely, Fig. 8(b) demonstrates the opposite scenario. This observation highlights the accurate reflection of keystroke positions by TDoA and PSD. Notably, the similarity between the signal waveforms of the two acoustic channels enables the determination of TDoA through cross-correlation.

The combination of our effective environment estimation algorithm and robust feature extraction enables us to achieve accurate keystroke prediction with a small training set, where the ratio of training data to test data is set at $5 : 1$. For keystroke classification, we employ SVM due to its ability to yield satisfactory results even with limited training data and various sample noises. This choice aligns with the realistic scenario of the sample scarcity in side-channel attacks.

### F. Word-Level Precision

When anticipating meaningful content, the keystroke classification performance can be enhanced by considering the alphabetical order of words, and the related word prediction can be provided by the *KeystrokeSniffer* system simultaneously. In English input, the presence of the Space and Enter keys signifies the separation between words. We identify these keys in signals and consider the segment between them as the signal corresponding to a word, which can obtain the keystroke sequence for each word.

For each keystroke predicted by the system, we can obtain a pair of parameters that represent the expected results and their corresponding confidence level. By processing all the keystrokes within a word, we can obtain a sequence of such parameter pairs. To enhance accuracy and fault tolerance, we utilize the Top-5 outcomes of each keystroke prediction as input for word-level prediction, which are sorted by confidence level.

For word-level predictions, we construct a dictionary comprising the 1500 most frequently used words. After identifying the keystrokes within a word and obtaining a series of results, *KeystrokeSniffer* first searches the dictionary for words of the same length. Then, the confidence of each word is calculated as the average of the confidence associated with its constituent letters. This approach provides us with confidence levels for all words.

## IV. Experimental Evaluation

### A. Implementation & Methodology

This section provides a detailed evaluation of the *KeystrokeSniffer* performance. The evaluations in this part are performed on the HUAWEI Mate20 Pro smartphone, whose two microphones are positioned at the bottom of the device. For the purpose of carrying out the side-channel attack, an Android application is developed using Java programming language, which can help collect acoustic signals. During the attack, the phone initiates the collection of acoustic signals at a sampling rate of $48kHz$.

To ensure the system robustness, we conduct the implementation using different keyboards with diverse timbres and sound intensities. The evaluation encompasses five mechanical keyboards, two membrane keyboards, and two laptop keyboards from Lenovo and Alienware laptops. Specifically, mechanical keyboards exhibit higher sound intensities, while the weakest keystroke sounds are observed on laptops. Furthermore, due to the variations in design and dimensions across different keyboard types, it is necessary to identify them prior to the evaluation.

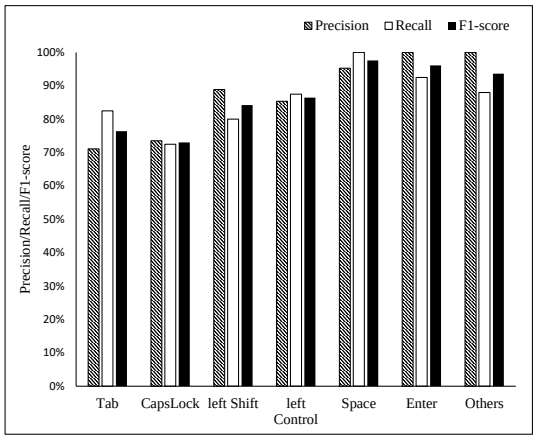
As discussed in the previous section, our evaluation and data collection focus on 32 keys on the keyboard. We recruit 12 volunteers to participate as victims, which consists of three female volunteers and nine male volunteers. In order to comprehensively assess the system's performance, we conduct data collection in diverse test scenarios. Each experimental scenario is defined by two variables: the relative position of the microphone and the type of keyboard. To cover a wide range of possibilities, we establish 48 experimental scenarios, encompassing all types of keyboards, victims, and $24$ microphone placements. The microphone positions include the top, bottom, left, and right on each of the six different planes, which are located at distances of $0cm$, $3cm$, $6cm$, $9cm$, $12cm$, and $15cm$ from the plane where the keyboard is positioned.

### B. Metrics

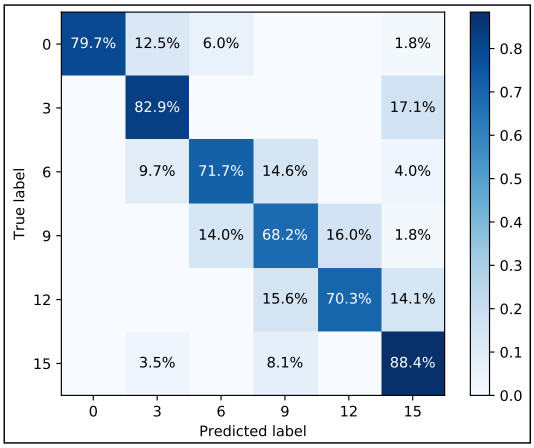
In this paper, we leverage the following metrics to evaluate the performance of the proposed attack system:

1) *Recall.* Given a series of keystrokes, the *recall* can be defined as:
$$Recall = \frac{TP}{TP + FN}, \tag{16}$$
where $TP$, $FN$ represent True Positive and False Negative, respectively.
2) *Precision.* Given a series of keystrokes, the *precision* can be expressed as:
$$Precision = \frac{TP}{TP + FP}, \tag{17}$$
where $FP$ represents the False Positive.
3) *F1-score.* The *F1-score* can be represented as:
$$\textit{F1-score} = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{18}$$
4) *Top-k Accuracy.* The Top-k Accuracy measures the probability of the correct result being among the top k sequences ranked by confidence.

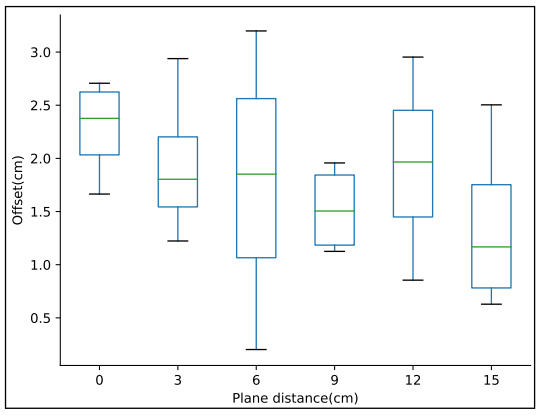
In particular, to demonstrate the expression rigor, referring to the definition of authoritative survey papers [32]–[34] on keystroke recognition, we define "accuracy" as the recognition precision of a single character or word above $80\%$, "practicality"



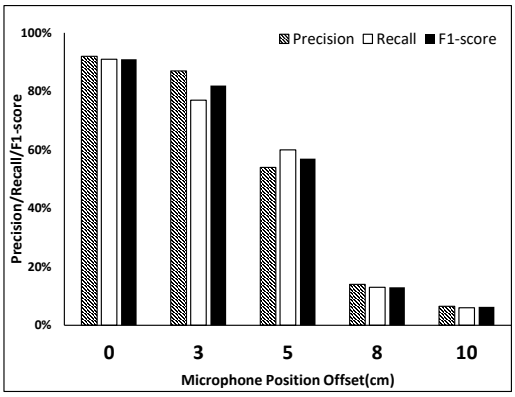

**(a)** Performance of recognizing large-size keys.




**(b)** Performance of microphone height estimation.




**(c)** Offset in position estimation.




**(d)** Influence of position estimation offset.

**Fig. 9** Performance of the CMA-ES-based microphone position estimation algorithm.

**TABLE II** Performance of the keyboard type recognition algorithm.

| | Precision | Recall | F1-score |
|---|---|---|---|
| Mechanical | 96.55% | 100.00% | 98.25% |
| Membrane | 100.00% | 96.97% | 98.46% |
| Laptop | 99.35% | 99.33% | 99.34% |

as the recognition precision of a single character or word above 70% when the input environment is different, and "satisfactory" as the recognition precision of a single character or word above 85%.

### *C. Evaluation of Microphones' Position Estimation*

The evaluations in this part primarily assess the performance of the environmental estimation algorithm, which contain keyboard type recognition, large-size key identification, and microphone position estimation. As attackers, we can perform environment estimation using a pre-collected training set. Since the accuracy of environment estimation predominantly depends on large-size keys, we collect 20 keystrokes from each of the 6 large-size keys across the 24 experimental scenarios for evaluation. To ensure the full representation of experimental effects in unfamiliar environments, the data from the keyboards and volunteers used for each testing session are not included in the training set.

To assess the usability of the environmental estimation algorithm, we first evaluate the keyboard type recognition method. For each evaluation, we use the data from a single keyboard as the testing set, while the corresponding training set consists of data from other keyboards of the same type. The average results of keyboard type recognition are reported in Tab. II. The experimental findings demonstrate that the proposed scheme effectively recognizes three commonly used keyboard types.

The subsequent part of the evaluation focuses on identifying the large-size keys contained in the sampled data. Alongside the existing dataset, we collect additional data on alphabetic keys for each experimental scenario to evaluate this aspect. We utilize data from one experimental scenario at a time as the testing set, while the remaining data serves as the training set. The average results of detecting six commonly used large-size keys during typing are depicted in Fig. 9(a). The average precision is 86.5%, and the average recall is 86.2%. The experiments confirm the system's capability to accurately identify large-size keys from a series of acoustic inputs. The recognition of the Space and Enter keys exhibits slightly higher accuracy compared to other keys, attributed to their distinctive timbre. Although occasional identification errors may occur, these samples can be easily

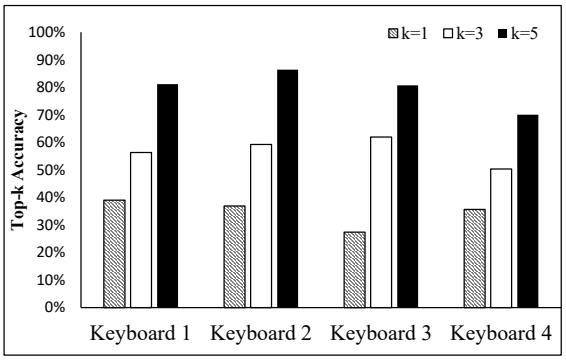



**Fig. 10** Performance on predicting keystrokes from unknown environments.

deleted due to their significant offset in location estimation.

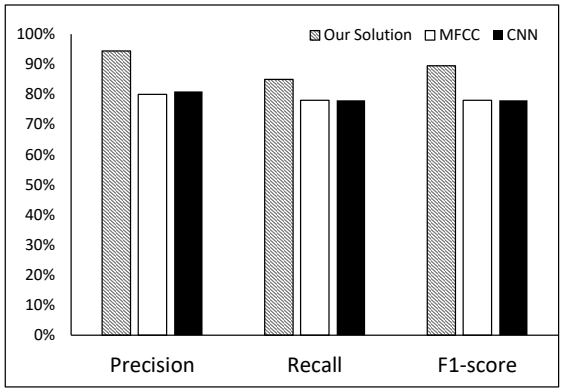
Next, we assess the performance of the microphone position estimation algorithm, which encompasses the classification of height planes and the offset of the final estimated coordinates. Height plane classification is conducted for each experimental scenario, and the mean values of the classification results for each plane are calculated. Fig. 9(b) shows the confusion matrix of the height plane classification results. The average recall achieves $76.9\%$. Experimental findings clearly demonstrate the effectiveness of the proposed algorithm within the range of $[0, 15cm]$, which satisfies most cases in daily typing situations.

By obtaining the value of $h$, attackers can proceed with the final estimation of the microphone coordinates. For each experimental scenario, we use different large-size keys to conduct three different estimations. The average results of coordinate estimation on each plane are presented in Fig. 9(c). Experimental results consistently demonstrate that the coordinate estimation deviation remains within $3cm$. These results illustrate that it is possible to estimate the relative position by only using a single smartphone and the recognized large-size keys without the need for prior knowledge of the keystroke labels.

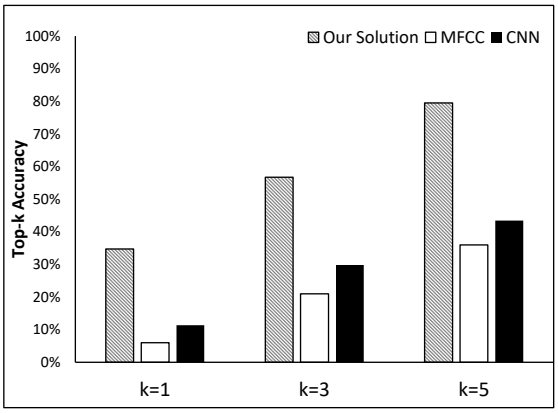
Since estimation errors are still present in the microphone coordinate results, we conduct tests to assess the impact of the position estimation offset on the system performance. Specifically, we select five microphone placement points at distances of $0cm$, $3cm$, $5cm$, $8cm$, and $10cm$ from the original position where the training set is collected. These evaluations are performed with the microphone placed at the top, bottom, left, and right of the keyboard, respectively. Then, we evaluate the prediction performance of unknown keystrokes collected from the chosen positions. The overall average performance is illustrated in Fig. 9(d). In particular, when the offset is $3cm$, the F1-score loss is below $8\%$. This finding suggests that an estimation error of approximately $3cm$ is acceptable for both microphone position estimation and keystroke prediction.

### *D. Evaluation in Unknown Environment*

One of our work's primary contributions is the proposed system's adaptability to different contexts and its ability to predict collected keystrokes without prior knowledge. In practical scenarios, attackers typically lack information about keyboards or victims prior to the attack. Thus, our objective is to develop a system that can accurately predict keystrokes from unknown keyboards and victims.




**(a)** Performance of different systems on recognizing keystrokes.




**(b)** Performance of different systems on predicting keystrokes from unknown environments.

**Fig. 11** Performance comparison with existing works.

In our data collection evaluation, we utilize various available keyboards and volunteers. These available keyboards and volunteers are divided into two groups: the attacker group and the victim group. The victim group includes 4 keyboards and 4 volunteers, while the remaining participants formed the attacker group. The victims perform keystrokes in various experimental scenarios, and the attackers attempt to eavesdrop on these keystroke samples in the absence of the victim's data.

Specifically, we first collect keystroke data of victims in 24 experimental scenarios across 6 planes, which serves as the testing set. Subsequently, the attackers utilize the victims' data to estimate the environmental information and create corresponding estimated scenarios. By employing training sets gathered from 24 simulated experimental scenarios, the attackers can capture the keystrokes from the victims' actual experimental scenarios. The size ratio of the training set to the testing set is set to $5:1$.

The system performance of predicting keystrokes from various unfamiliar keyboards is illustrated in Fig. 10. We can observe that the average accuracy for Top-1, Top-3, and Top-5 predictions is $34.75\%$, $56.75\%$, and $79.50\%$, respectively. The experimental results clearly demonstrate *KeystrokeSniffer* ability to accurately predict keystrokes across different keyboard types, even in unfamiliar environments.

### *E. Comparison with Previous Works*

In this section, we compare our system with two state-of-the-art studies, *i.e.*, one deep learning-based method [8] and

**TABLE III** Prediction of words. Position in results means which position in the prediction results is the real word.

| Words | degree | university | children | what | book | promotion | campaign | honey | quiet |
|---|---|---|---|---|---|---|---|---|---|
| Length | 6 | 10 | 8 | 4 | 4 | 9 | 8 | 5 | 5 |
| Position in results (User1) | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| Position in results (User2) | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 2 |
| Position in results (User3) | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 |
| Average position | 1.00 | 1.00 | 1.00 | 1.33 | 1.67 | 1.00 | 1.33 | 1.00 | 1.67 |
| Words | bring | misfortune | friend | country | nothing | proceed | could | just | zoom |
| Length | 5 | 10 | 6 | 7 | 7 | 7 | 5 | 4 | 4 |
| Position in results (User1) | 1 | 1 | 1 | 1 | 1 | 2 | 5 | 1 | 3 |
| Position in results (User2) | 1 | 1 | 3 | 4 | 1 | 1 | 1 | 1 | 2 |
| Position in results (User3) | 1 | 1 | 1 | 5 | 1 | 1 | 1 | 1 | 1 |
| Average position | 1.00 | 1.00 | 1.67 | 3.33 | 1.00 | 1.33 | 2.33 | 1.00 | 2.00 |

one signal analysis-based method [9]. The work in [8] utilized CNN for keystroke recognition, which employed multiple mobile phones as recording devices. The second work, conducted by [9], focused on classifying keystrokes by extracting the MFCC of acoustic waves. Through experiments conducted in two different scenarios, we evaluate the *KeystrokeSniffer* performance for keystroke recognition and compare it with the performance of these two baseline strategies.

We first test the keystroke recognition performance when the keyboard type and microphone placement are known. We conduct experiments using all acquired datasets from each experimental scenario and present the results of a 10-fold cross-validation. Fig. 11(a) illustrates the overall performance of the proposed system and two baseline systems, which represents the average results for recognizing keystrokes across 9 keyboards and 12 volunteers. The experiments demonstrate that *KeystrokeSniffer* achieves higher average precision, recall, and F1-score, reaching $94.4\%$, $85.0\%$, and $89.5\%$, respectively, which significantly surpasses the recognition results obtained in [8] and [9]. This clearly highlights the superior performance of the proposed system in keystroke prediction, which can be attributed to the adoption of a robust feature extraction algorithm that is related to the keystroke location.

Furthermore, we compare the performance of keystroke prediction in unfamiliar environments. These experimental scenarios correspond to the ones described in Sec. IV-D. As depicted in Fig. 11(b), the proposed system exhibits a significant performance advantage over the two baseline schemes, which is attributed to its efficient environment estimation scheme and robust feature selection. Although the Top-1 Accuracy decreases when confronted with unknown victims and keyboards, a favorable Top-5 Accuracy can still be achieved, which proves valuable in identifying the typing words in unfamiliar settings. This illustrates that it is feasible for *KeystrokeSniffer* to accurately identify keystrokes from unknown environments through the novel environment estimation scheme and the robust feature extraction algorithm.

### *F. Performance of Word Prediction*

To demonstrate the accuracy of word prediction, we select 120 words of varying lengths and letter compositions from the dictionary and predict them. The experimental settings are identical to those outlined in Sec. IV-D, where victims input 5 different words in each experimental scenario. The selected words are chosen randomly from the dictionary, which ensures they exhibit the following characteristics: 1) frequent usage, 2) diverse letter compositions, and 3) inclusion of words with both the same and different lengths. The average Top-k accuracy ratios for the words contained in the testing set are $66.7\%$, $81.7\%$, and $96.6\%$ for k=1, 3, and 5, respectively. We further select frequently used words from the test results and present them in Tab. III. The position value represents the confidence ranking of the true result within the sequence of predicted results.

The evaluation demonstrates *KeystrokeSniffer* ability to accurately distinguish words with similar structures and to effectively handle a wide range of words, which highlights its strong generalization capabilities for word-level prediction. *KeystrokeSniffer* successfully restores the victim's input, which validates the improved practical performance due to considering the alphabetical order of meaning words. Consequently, these side-channel attacks pose more significant risks to user privacy than previously anticipated, particularly in light of the widespread use of smart devices today.

### *G. Deep Dive into KeystrokeSniffer*

The main purpose of this section is to demonstrate the impacts of various factors, *e.g.*, environmental factors, equipment factors, processing factors, and human factors, on the *KeystrokeSniffer* system. In order to understand the impact of a single factor on the experimental results and effectively control variables, the experiments in this section are conducted with keyboard type and microphone placement location known. In order to minimize the influence of extraneous factors, we maintain a fixed microphone placement throughout the experiments.

*1) Impact of Noises:* To evaluate the system's robustness against noises, we perform experiments in three different scenarios with varying noise levels. The scenarios and corresponding

**TABLE IV** Types of noises.

| Scene of occurrence | Type of noises |
|---|---|
| Apartment | White noise ($31dB$) |
| Conference room | White noise ($30dB$), Conversation ($61dB$), Music ($42dB$) |
| Laboratory | White noise ($25dB$), Keystrokes ($45dB$), Machine ($43dB$) |

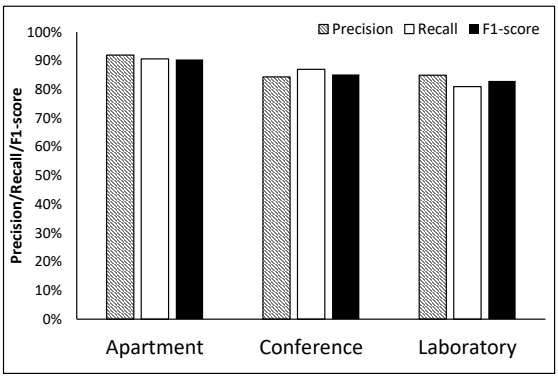



**Fig. 12** Performance in noisy environments.

noise types are detailed in Tab. IV. Specifically, noise types include white noise, conversation noise, music noise, irrelevant keystroke noise, and machine operating noise, and their noise intensity ranges from $25dB$ to $61dB$.

As illustrated in Fig. 12, *KeystrokeSniffer* exhibits stability and accuracy in common typing scenarios with various types of noises. By employing 10-fold cross-validation, the F1-scores for keystroke prediction in three different scenarios are found to be $90.43\%$, $85.22\%$, and $83.0\%$, respectively. Although a slight negative impact on keystroke prediction may arise as the environmental noise component gradually increases, the accuracy and robustness of *KeystrokeSniffer* are still maintained. This robustness is attributed to the noise elimination algorithm and the robust feature selection, which affirm the superiority of the proposed system in common noise scenarios.

*2) Impact of Different Mobile Phones:* To demonstrate the impact of microphone variations across different mobile phones on recognition performance and assess *KeystrokeSniffer* robustness to diverse sound collection equipment, we include five additional mobile phones, namely, HUAWEI Mate40 Pro+, Vivo X60, Samsung Galaxy Z Fold2, Xiaomi 13, and OnePlus 6. We then calculate the corresponding keystroke recognition precision, recall, and F1-score. Each mobile phone imitates the way that the HUAWEI Mate20 Pro smartphone collects signals and performs the same experiments as introduced in Sec. IV-E.

Fig. 13 shows the various recognition performance metrics, *i.e.*, Precision, Recall, and F1-score, of the *KeystrokeSniffer* system influenced by different sound collection equipment from different phone models. From Fig. 13, we can observe that *KeystrokeSniffer* achieves satisfactory and similar recognition performance across different sound collection equipment. Furthermore, even if a OnePlus 6 released six years ago (2018.5.17) is used for sound signal collection, the *KeystrokeSniffer* system can still achieve an F1 score of $87.7\%$. These results clearly demonstrate the strong robustness of the *KeystrokeSniffer* system to different sound collection equipment from various smartphones, which is conducive to deploying the *KeystrokeSniffer* system in different devices to achieve effective identification of keystroke content.

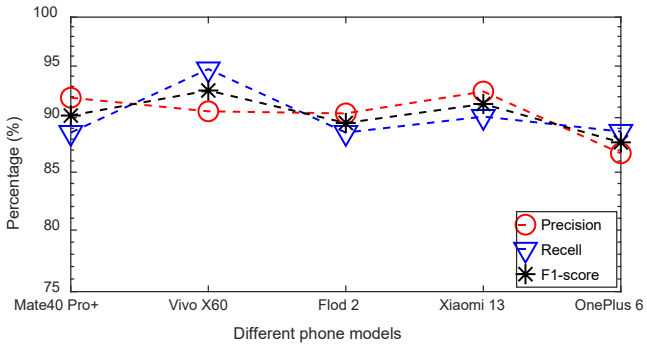



**Fig. 13** The performance influenced by various sound collection equipment from different phone models.

**TABLE V** Impact of different window sizes and shift steps on the system F1-score.

| Window size/ Shift step | $2ms$ | $5ms$ | $\mathbf{10ms}$ | $20ms$ | $50ms$ |
|---|---|---|---|---|---|
| $0.1\times$ | $54.6\%$ | $70.6\%$ | $78.4\%$ | $74.2\%$ | $81.5\%$ |
| $\mathbf{0.25\times}$ | $60.4\%$ | $72.4\%$ | $\mathbf{89.5\%}$ | $86.5\%$ | $79.4\%$ |
| $0.5\times$ | $65.7\%$ | $74.8\%$ | $87.7\%$ | $81.9\%$ | $77.8\%$ |
| $1\times$ | $66.1\%$ | $73.5\%$ | $74.6\%$ | $69.7\%$ | $67.3\%$ |

*3) Impact of Window Size and Shift Step:* The segmentation window size and the shift step inevitably affect the accuracy of keystroke recognition. A window size that is too large can easily increase the computational complexity of keystroke detection. On the contrary, a window that is too small cannot maintain enough information, which results in a decrease in keystroke recognition accuracy [35]. Furthermore, the shift step affects the smoothness of the processed signals. To find the appropriate segmentation window size and the shift step, the impact of the window size $W$ and the shift step $S$ on the system performance is presented. Specifically, we set the window size $W$ to one of the $2ms$, $5ms$, $10ms$, $20ms$, and $50ms$ and chose the shift step $S$ as $0.1\times$, $0.25\times$, $0.5\times$, and $1\times$ times the corresponding window size.

Tab. V shows the F1-scores of the proposed system under the impact of different window sizes and shift steps. We observe that the recognition performance (F1-score) basically increases with a larger window size when $W$ is less than 10ms and has a slight drop when $W$ is larger than 10ms. This is because the $10ms$ sliding window contains enough keystroke information, and the shorter sliding window contains less keystroke information. However, a sliding window that is too long cannot guarantee the smoothness of the sound signals, and the saliency of the keystroke features can also be affected. Besides, shift steps also affect keystroke recognition performance. This is because the shift step size affects the difference and stability of continuous sampling. When the window size $W$ is $10ms$ with a shift step size $S$ of $0.25$ times, it can help the system to better capture the keystroke characteristics and achieve the best recognition performance.

*4) Impact of Hand Length and Palm Size:* Apparently, hand length and palm size significantly affect how keystrokes are performed. Fig. 14 depicts keystroke postures performed by subjects with different hand lengths. Specifically, the hand length


**(a)** Keystroke posture performed by the subject with a short finger length.

**(b)** Keystroke posture performed by the subject with a long finger length.

**Fig. 14** Keystroke postures for different finger lengths.

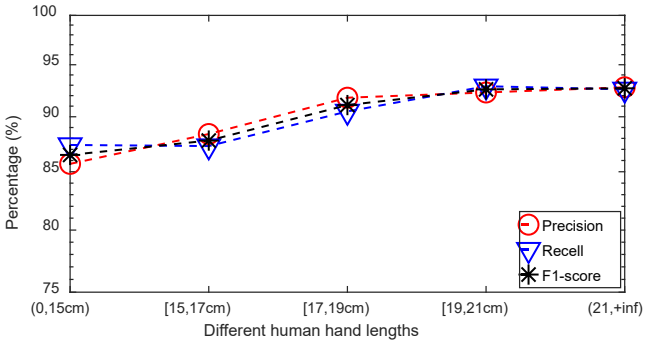



**Fig. 15** The performance influenced by different human hand lengths.

is defined as the distance from the tip of the middle finger to the bottom of the palm when the subject is in a relaxed state. We observe that the keystrokes of subjects with long fingers are more inclined, and the relative keystroke angle is smaller. These differences in keystroke angles and postures may affect the recognition results.

To demonstrate the robustness of *KeystrokeSniffer* to different hand lengths, we divide the length of human hands into five intervals $(0, 15cm)$, $[15, 17cm)$, $[17, 19cm)$, $[19, 21cm)$, and $(21cm, \infty)$ [36], and hire three different volunteers to perform keystrokes in each interval. We train and test the collected sound data in each hand length interval separately and present the results of a 10-fold cross-validation.

Fig. 15 depicts the recognition performance of the *KeystrokeSniffer* system to the keystrokes performed by different hand-length persons. As the length of the subject's hand increases, the system's recognition performance first rises significantly and then remains stable. This is because the longer the hand, the greater the muscle changes caused by keystrokes, which finally results in the more obvious signal fluctuations. Accordingly, it is easier to analyze the movement changes from the obvious fluctuation signals. Nevertheless, we can observe that the system achieves satisfactory keystroke recognition accuracy for all kinds of hand lengths. Even for the keystrokes conducted by subjects with hand lengths not exceeding $15cm$, *KeystrokeSniffer* can still achieve a comparable recognition accuracy of more than $85.7\%$. This clearly demonstrates the strong robustness of the *KeystrokeSniffer* system for various subjects with different hand lengths since timbre-independent and position-related keystroke features are effectively extracted.

## V. Conclusion

This paper presents a practical side-channel attack (*KeystrokeSniffer*) on keyboard input using acoustic signals captured by an off-to-shelf smartphone. To address unfamiliar environments, we propose an efficient approach for estimating keyboard types and microphone positions. Additionally, by further disregarding keystroke sound-related elements and concentrating on uncovering the keystroke positions, *KeystrokeSniffer* extract two timbre-independent and position-related keystroke robust features to mitigate the impact of environmental variations while ensuring precise keystroke classification and effectively tackles the problem caused by the varying input habits of different victims. Extensive experimental results demonstrate that compared with state-of-the-art methods, the proposed system achieves enhanced robustness and accuracy in keystroke prediction. Even with limited training data, this side-channel attack strategy poses a significant threat to the victim input, which makes it practical to perform and challenging to mitigate. By setting different parameter values of various experiment impact factors, we further verify the strong robustness of *KeystrokeSniffer* to different factors, *e.g.*, noises, mobile phone models, processing window size, and human hand length, which proves that *KeystrokeSniffer* can create privacy threats in real situations.

## VI. Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 62302145), Anhui Province Science Foundation for Youths (Grant No. 2308085QF230), and Young teachers' scientific research innovation launches special A project (Grant No. JZ2023HGQA0100). We would like to thank the editors and anonymous reviewers for their insightful comments and constructive feedback.

## References

[1] Yasha Iravantchi, Yi Zhao, Kenrick Kin, and Alanson P Sample, "Sawsense: Using surface acoustic waves for surface-bound event recognition," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–18.

[2] Soumik Mondal and Patrick Bours, "Person identification by keystroke dynamics using pairwise user coupling," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1319–1329, 2017.

[3] Ahmad Bazzi and Marwa Chafii, "Secure full duplex integrated sensing and communications," *IEEE Transactions on Information Forensics and Security*, 2023.

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

[5] Jia-Ling Huang, Yun-Shu Wang, Yong-Pan Zou, Kai-Shun Wu, and Lionel Ming-shuan Ni, "Ubiquitous wifi and acoustic sensing: Principles, technologies, and applications," *Journal of Computer Science and Technology*, vol. 38, no. 1, pp. 25–63, 2023.

[6] Zhen Xiao, Tao Chen, Yang Liu, and Zhenjiang Li, "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, 2020.

[7] Yazhou Tu, Liqun Shan, Md Imran Hossen, Sara Rampazzi, Kevin Butler, and Xiali Hei, "Auditory eyesight: Demystifying {$\mu$s-Precision} keystroke tracking attacks on unconstrained keyboard inputs," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 175–192.

[8] Tyler Giallanza, Travis Siems, Elena Smith, Erik Gabrielsen, Ian Johnson, Mitchell A Thornton, and Eric C Larson, "Keyboard snooping from mobile phone arrays with mixed convolutional and recurrent neural networks," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–22, 2019.

[9] Stefano Cecconello, Alberto Compagno, Mauro Conti, Daniele Lain, and Gene Tsudik, "Skype & type: Keyboard eavesdropping in voice-over-ip," *ACM Transactions on Privacy and Security (TOPS)*, vol. 22, no. 4, pp. 1–34, 2019.

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

[11] Anindya Maiti, Murtuza Jadliwala, Jibo He, and Igor Bilogrevic, "Side-channel inference attacks on mobile keypads using smartwatches," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2180–2194, 2018.

[12] Tiantian Zhu, Lei Fu, Qiang Liu, Zi Lin, Yan Chen, and Tieming Chen, "One cycle attack: Fool sensor-based personal gait authentication with clustering," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 553–568, 2020.

[13] Edwin Yang, Song Fang, Ian Markwood, Yao Liu, Shangqing Zhao, Zhuo Lu, and Haojin Zhu, "Wireless training-free keystroke inference attack and defense," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1733–1748, 2022.

[14] Kang Ling, Yuntang Liu, Ke Sun, Wei Wang, Lei Xie, and Qing Gu, "Spidermon: Towards using cell towers as illuminating sources for keystroke monitoring," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 666–675.

[15] Jingyang Hu, Hongbo Wang, Tianyue Zheng, Jingzhi Hu, Zhe Chen, Hongbo Jiang, and Jun Luo, "Password-stealing without hacking: Wi-fi enabled practical keystroke eavesdropping," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 239–252.

[16] Hongbo Wang, Jingyang Hu, Tianyue Zheng, Jingzhi Hu, Zhe Chen, Hongbo Jiang, Yuanjin Zheng, and Jun Luo, "Muki-fi: Multi-person keystroke inference with bfi-enabled wi-fi sensing," *IEEE Transactions on Mobile Computing*, 2024.

[17] Jiayu Zhou, Wenjie Ding, and Wen Yang, "A secure encoding mechanism against deception attacks on multisensor remote state estimation," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1959–1969, 2022.

[18] Jinyang Huang, Bin Liu, Chenglin Miao, Xiang Zhang, Jiancun Liu, Lu Su, Zhi Liu, and Yu Gu, "Phyfinatt: An undetectable attack framework against phy layer fingerprint-based wifi authentication," *IEEE Transactions on Mobile Computing*, 2023.

[19] Shiqing Luo, Anh Nguyen, Hafsa Farooq, Kun Sun, and Zhisheng Yan, "Eavesdropping on controller acoustic emanation for keystroke inference attack in virtual reality," in *The Network and Distributed System Security Symposium (NDSS)*, 2024.

[20] Zhen Xiao, Tao Chen, Yang Liu, Jiao Li, and Zhenjiang Li, "Keystroke recognition with the tapping sound recorded by mobile phone microphones," *IEEE Transactions on Mobile Computing*, 2021.

[21] Yanchao Zhao, Yiming Zhao, Si Li, Hao Han, and Lei Xie, "Ultrasnoop: Placement-agnostic keystroke snooping via smartphone-based ultrasonic sonar," *ACM Transactions on Internet of Things*, vol. 4, no. 4, pp. 1–24, 2023.

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

[23] Bin Chen, Minghao Zhang, Zhankun Lin, and Hao Xu, "Acoustic-based whistle detection of drain hole for wind turbine blade," *ISA transactions*, vol. 131, pp. 736–747, 2022.

[24] Wee-Soon Ching and Peng-Seng Toh, "Enhancement of speech signal corrupted by high acoustic noise," in *Proceedings of TENCON '93. IEEE Region 10 International Conference on Computers, Communications and Automation*, 1993, vol. 2, pp. 1114–1117 vol.2.

[25] Do Yeong Lim and In Cheol Bang, "A novel non-destructive acoustic approach for investigating pool boiling phenomena," *International Journal of Heat and Mass Transfer*, vol. 222, pp. 125166, 2024.

[26] Jiadi Yu, Li Lu, Yingying Chen, Yanmin Zhu, and Linghe Kong, "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 337–351, 2019.

[27] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu, "Context-free attacks using keyboard acoustic emanations," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 453–464.

[28] Gang Wang, Yudong Xiao, KC Ho, and Lei Huang, "Unified near-field and far-field tdoa source localization without the knowledge of signal propagation speed," *IEEE Transactions on Communications*, 2023.

[29] Mingyang Hao, Fangli Ning, Ke Wang, Shaodong Duan, Zhongshan Wang, Di Meng, and Penghao Xie, "Acoustic non-line-of-sight vehicle approaching and leaving detection," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[30] Eugenio J Muttio, Wulf G Dettmer, Jac Clarke, Djordje Perić, Zhaoxin Ren, and Lloyd Fletcher, "A supervised parallel optimisation framework for metaheuristic algorithms," *Swarm and Evolutionary Computation*, vol. 84, pp. 101445, 2024.

[31] Adriaan Peetermans and Ingrid Verbauwhede, "Characterization of oscillator phase noise arising from multiple sources for asic true random number generation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2024.

[32] Blaine Ayotte, Mahesh K Banavar, Daqing Hou, and Stephanie Schuckers, "Group leakage overestimates performance: A case study in keystroke dynamics," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2021, pp. 1410–1417.

[33] Min Peng, Xianxin Fu, Haiyang Zhao, Yu Wang, and Caihong Kai, "Likey: Location-independent keystroke recognition on numeric keypads using wifi signal," *Computer Networks*, p. 110354, 2024.

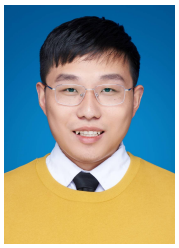
[34] Sandeep Gupta, Carsten Maple, Bruno Crispo, Kiran Raja, Artsiom Yautsiukhin, and Fabio Martinelli, "A survey of human-computer interaction (hci) & natural habits-based behavioural biometric modalities for user recognition schemes," *Pattern Recognition*, vol. 139, pp. 109453, 2023.

[35] Jinyang Huang, Bin Liu, Chao Chen, Hongxin Jin, Zhiqiang Liu, Chi Zhang, and Nenghai Yu, "Towards anti-interference human activity recognition based on wifi subcarrier correlation selection," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6739–6754, 2020.

[36] RS Guerra, I Fonseca, F Pichel, MT Restivo, and TF Amaral, "Hand length as an alternative measurement of height," *European journal of clinical nutrition*, vol. 68, no. 2, pp. 229–233, 2014.


**Jinyang Huang** received the Ph.D. degree in School of Cyberspace Security from the University of Science and Technology of China in 2022. He is currently a lecturer in the School of Computer and Information at Hefei University of Technology. His research interests include Wireless Security and Wireless Sensing. He is a TPC Member of ACM MM, IEEE ICME, and Globecom. He is now an editorial board member of Applied Sciences.

**Jia-Xuan Bai** received the Master degree in School of Cyberspace Security from the University of Science and Technology of China in 2022. He is currently working on wireless communications at Huawei Technologies Co., Ltd. His research interests include Wireless Sensing and terminal communication.



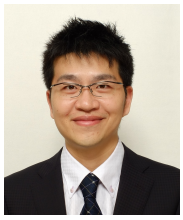
**Xiang Zhang** received the B.E. degree from Hefei University of Technology, China, in 2017, and his D.E. degree from the same university in 2023. Currently, he is a postdoc with the School of Cyber Science and Technology, University of Science and Technology of China. His research interests include wireless sensing and affective computing. He is a TPC Member of IEEE ICME and Globecom. He has served as a reviewer for IEEE TNNLS, TMM, and Pattern Recognition.

**Zhi Liu** (S'11-M'14-SM'19) received the Ph.D. degree in informatics in National Institute of Informatics. He is currently an Associate Professor at The University of Electro-Communications. His research interest includes video network transmission and mobile edge computing. He is now an editorial board member of Springer wireless networks and IEEE Open Journal of the Computer Society. He is a senior member of IEEE.

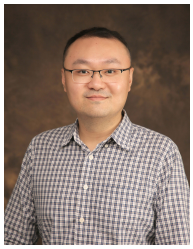
**Yuanhao Feng** is now a Postdoctoral Fellow in the Department of Computing, The Hong Kong Polytechnic University. He received the Ph.D. degree from the University of Science and Technology of China in 2023. He is an outstanding graduate of the School of Computer Science in USTC 2023. Dr. Feng has long-term cooperation with the research group of Dr. Wang Meng of Hefei University of Technology. He has published many top journal and conference papers, such as MobiCom, INFOCOM, and TMC. His research interest is AIoT, which includes wireless sensing and communications.

**Jianchun Liu** (Member, IEEE/ACM) received the Ph.D. degree in School of Data Science from the University of Science and Technology of China in 2022. He is currently an associate researcher in the School of Computer Science and Technology at University of Science and Technology of China. His main research interests are software defined networks, network function virtualization, edge computing and federated learning.

**Xiao Sun** was born in 1980. He received the M.E. degree from the Department of Computer Sciences and Engineering, Dalian University of Technology, Dalian, China, in 2004, the first Ph.D. degree from the University of Tokushima, Tokushima, Japan, in 2009, and the second Ph.D. degree Dalian University of Technology in 2010. He is currently working as an Professor with the Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machines, Hefei University of Technology, Hefei, China. His research interests include affective computing, natural language processing, machine learning, and human–machine interaction.



**Mianxiong Dong** received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is the Vice President and Professor of Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at the University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. He is the recipient of The 12th IEEE ComSoc Asia Pacific Young Researcher Award 2017, Funai Research Award 2018, NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology, The Young Scientists' Award from MEXT in 2021, SUEMATSU-Yasuharu Award from IEICE in 2021, IEEE TCSC Middle Career Award in 2021. He is Clarivate Analytics 2019, 2021, 2022 Highly Cited Researcher (Web of Science) and Foreign Fellow of EAJ.



**Meng Li** is an Associate Professor and Dean Assistant at the School of Computer Science and Information Engineering, Hefei University of Technology (HFUT), China. He is also a Post-Doc Researcher at Department of Mathematics and HIT Center, University of Padua, Italy, where he is with the Security and PRIvacy Through Zeal (SPRITZ) research group led by Prof. Mauro Conti (IEEE Fellow). He obtained his Ph.D. in Computer Science and Technology from the School of Computer Science and Technology, Beijing Institute of Technology (BIT), China, in 2019. He was sponsored by ERCIM 'Alain Bensoussan' Fellowship Programme (from 2020.10.1 to 2021.3.31) to conduct Post-Doc research supervised by Prof. Fabio Martinelli at CNR, Italy. He was sponsored by China Scholarship Council (CSC) (from 2017.9.1 to 2018.8.31) for joint Ph.D. study supervised by Prof. Xiaodong Lin (IEEE Fellow) in the Broadband Communications Research (BBCR) Lab at University of Waterloo and Wilfrid Laurier University, Canada. His research interests include data security, privacy preservation, applied cryptography, blockchain, TEE, and Internet of Vehicles. In this area, he has published 70 papers in international peer-reviewed journals and conferences, including TIFS, TDSC, ToN, TKDE, TODS, TSC, TSG, TII, TVT, TNSM, TNSE, TGCN, COMST, MobiCom, ICICS, SecureComm, TrustCom, and IPCCC. He is a Senior Member of IEEE. He is an Associate Editor for IEEE TIFS, IEEE TNSM, and IEEE IoTJ.