$KeyListener$: Inferring Keystrokes on QWERTY Keyboard of Touch Screen through Acoustic Signals

Li Lu*, Jiadi Yu*‡, Yingying Chen†, Yanmin Zhu*, Xiangyu Xu*, Guangtao Xue*, Minglu Li*
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R.China
Email: {luli_jtu, jiadiyu, yzhu, chillex, gt_xue, mlli}@sjtu.edu.cn
†WINLAB, Rutgers University, New Brunswick, NJ, USA
Email: yingche@scarletmail.rutgers.edu
‡Corresponding Author

Abstract—**This paper demonstrates the feasibility of a side-channel attack to infer keystrokes on touch screen leveraging an off-the-shelf smartphone. Although there exist some studies on keystroke eavesdropping attacks on touch screen, they are mainly direct eavesdropping attacks, i.e., require the device of victims compromised to provide side-channel information for the adversary, which are hardly launched in practical scenarios. In this work, we show the practicability of an indirect eavesdropping attack, $KeyListener$, which infers keystrokes on QWERTY keyboards of touch screen leveraging audio devices on a smartphone. We investigate the attenuation of acoustic signals, and find that a user's keystroke fingers can be localized through the attenuation of acoustic signals received by the microphones in the smartphone. We then utilize the attenuation of acoustic signals to localize each keystroke, and further analyze errors induced by ambient noises. To improve the accuracy of keystroke localization, $KeyListener$ further tracks finger movements during inputs through phase change and Doppler effect to reduce errors of acoustic signal attenuation-based keystroke localization. In addition, a binary tree-based search approach is employed to infer keystrokes in a context-aware manner. The proposed keystroke eavesdropping attack is robust to various environments without the assistance of additional infrastructures. Extensive experiments demonstrate that the accuracy of keystroke inference in top-5 candidates can approach 90% with a top-5 error rate of around 6%, which is a strong indication of the possible user privacy leakage of inputs on QWERTY keyboard.**

I. Introduction

Mobile devices equipped with touch screen become more popular and pervasive in the daily lives. These devices are commonly used to input private information, such as payment information, email/chatting messages, and personal documents. According to Federal Reserve System, 43% users in the USA adopt mobile banking for their daily financial activities in 2015 [1]. Moreover, there is a report showing that around 1,500 million users chat online monthly through instant messaging APPs on smartphones [2]. Instead of stationary devices staying at a physically-secure location, mobile devices are often carried by users traveling to different places, where the devices are exposed to possible eavesdropping attacks, such as WiFi-based password eavesdropping, etc.

Currently, most existing studies [3]–[6] about keystroke eavesdropping attacks mainly concentrate on physical keyboards. Compared with a physical keyboard, the size of a virtual keyboard on touch screen is far smaller, and keystrokes on touch screen induce little sound and vibration. Thus, new challenges are introduced in keystroke eavesdropping attacks on virtual keyboards of touch screen. There are some existing works [7], [8] about keystroke eavesdropping attacks on touch screen, which are based on keystroke patterns in the motion sensor data on victims' smartphones. However, these works all involve a direct eavesdropping attack, i.e., the sensor data on the victim's device are compromised to provide side-channel information about keystrokes for the adversary, which is hardly practical and limits the impact of such attacks. Recently, Li et al. [9] propose an indirect eavesdropping attack, i.e., without the requirement of side-channel information directly from victims' devices, to identify PIN inputs on touch screen based on Channel State Information of WiFi signals. However, the attack scenario of this work is constrained to the coverage of WiFi infrastructure, and the attack is only for 9-key PIN keyboard instead of QWERTY virtual keyboard.

Toward this end, we explore how to launch a side-channel attack leveraging audio devices from smartphones to infer keystrokes on QWERTY keyboards of touch screen. Such an attack is powerful without the assistance of additional devices and resilient to various environments. Since acoustic signals have advantages of strong penetrability and slow propagating velocity, the signals have been utilized in recent researches, such as object movement tracking [10]–[12], inconspicuous attack to voice assistants [13], and user authentication [14], [15], etc. Thus, we consider whether it is feasible to utilize acoustic signals for keystroke eavesdropping attacks. To realize such an attack leveraging acoustic signals, we face several challenges in practice. First, the attack should utilize limited audio devices on smartphones to localize keystrokes during inputs. Second, the attack needs to resist ambient noises in received acoustic signals. Finally, the attack ought to identify keystrokes without training information from victims.

In this paper, we first describe the attack scenario and analyze the feasibility of utilizing acoustic signals to infer keystrokes on QWERTY keyboard of touch screen. Through the analysis, we find that the attenuation of acoustic signals can be used to localize each keystroke during inputs. Inspired by the observation, we demonstrate a possible side-channel attack, $KeyListener$, which can infer keystrokes on QWERTY keyboard of touch screen through audio devices on a smartphone.

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.

In $KeyListener$, the speaker of an adversary's smartphone first emits near-ultra acoustic signals (inaudible for humans), and then the signals are received by two microphones of the smartphone. $KeyListener$ first mitigates multipath reflections in received acoustic signals, and segments received acoustic signals into each keystroke and finger movement window, so as to extract the acoustic signal of each input behavior. Then, each keystroke during victims' inputs is localized based on the attenuation of acoustic signals. Since our attack aims to be robust to various environments, we analyze the impact of error induced by ambient noises in the keystroke localization, and construct an area for a localized keystroke, i.e, *keystroke range*, which is the range of error in the acoustic signal attenuation-based keystroke localization. To improve the accuracy of keystroke localization, we further analyze users' finger behaviors during inputs, and find the finger movement between two keystrokes contributes to reducing errors of the keystroke localization. Specifically, $KeyListener$ first tracks the range of a finger movement between two keystrokes via phase changes and Doppler shifts of acoustic signals, and then intersects the range of the finger movement with the localized keystroke range to reduce the error. Finally, based on the keys covered by the keystroke range of each finger keystroke, the adversary can infer victims' continuous keystrokes in a context-aware manner through a binary tree-based search approach. Our extensive experiments demonstrate that $KeyListener$ is robust and efficient to infer keystrokes on QWERTY keyboard of touch screen in real environments.

We highlight our contribution in this paper as follows.

- We demonstrate that a commercial smartphone can recover keystrokes on QWERTY keyboard of touch screen through acoustic signals.
- We exploit the attenuation of acoustic signals to localize keystrokes and analyze errors induced by ambient noises in the keystroke localization.
- We improve the accuracy of keystroke localization through tracking finger movement behaviors during inputs based on phase changes and Doppler effect of acoustic signals.
- We conduct experiments in real environments. The results show that the proposed keystroke inference attack can approach 90% accuracy under top-5 word candidates.

The rest of this paper is organized as follows. We first show the preliminary in Section II. Then, Section III presents the system design of $KeyListener$. The evaluation of the system is presented in Section IV. Finally, we review several related work in Section V and make a conclusion in Section VI.

II. Preliminary

In this section, we introduce the attack scenario and basic principles of inferring keystrokes on QWERTY keyboard of touch screen through acoustic signals.

A. Attack Scenario

The keystroke eavesdropping attack scenario is considered as that an adversary seeks to infer a victim's keystrokes on QWERTY keyboard of touch screen through audio devices integrated in a smartphone. We assume an adversary takes a smartphone and is inconspicuously close to a victim's touch-capable devices. Since the victim tries to avoid the leakage of his/her inputs, the adversary cannot eavesdrop the inputs through line-of-sight observation. Two representative scenarios which validate the plausibility of the assumption are: (1) the adversary inconspicuously sits beside the victim in a confined setting such as library and canteen, where physical proximity is not suspicious, and uses a smartphone to perform the keystroke eavesdropping attack on the victim's input on touch screen; (2) the adversary inconspicuously stands beside the victim while queuing for payment through a mobile device, and uses a smartphone to perform the password eavesdropping attack when the victim pays the bill. Since mobile devices can be carried by victims to arbitrary places, an adversary should perform the keystroke eavesdropping attack without environmental restriction and the assistance of additional infrastructures other than a smartphone.

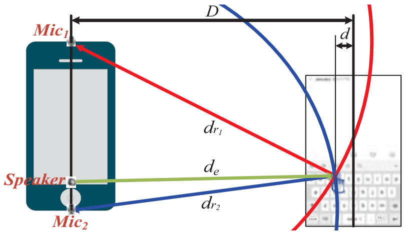


Fig. 1. Basic principles of keystroke localization based on attenuation of acoustic signals.

B. Basic Principles of Acoustic Signal Attenuation-based Keystroke Localization

To implement the side-channel attack, i.e., keystroke eavesdropping attack, we present the basic principles of keystroke localization on QWERTY keyboard of touch screen through the attenuation of acoustic signals via a smartphone.

Recent works [10]–[12] utilize phase changes in acoustic signals for object movement tracking. Intuitively, these works seem to be feasible for identifying a user's inputs through tracking finger movements in the attack scenario. However, there exist several significant problems. First, these works are only designed to track finger movements in a 2-D plane, but a user's input behaviors, i.e., keystrokes and finger movements between keystrokes, are usually not in the same 2-D plane. The phase-based methods would regard all keystrokes and movements between keystrokes as finger movements in a 2-D plane, i.e., treat all finger travels of the two kinds of input behaviors as the moving distances in the same 2-D plane, which induces significant cumulative errors in finger movement tracking during inputs. Moreover, these finger tracking methods need to determine the initial position of a finger movement, so as to identify the absolute position of trajectory in a finger movement. Due to limits in audio devices of smartphones, phase-based methods suffer significant performance degradation when the distance between the smartphone and tracked object increases (e.g., larger than $30cm$) [10].

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.

In order to accurately identify users' inputs on touch screen, we propose the acoustic signal attenuation-based approach to localize keystrokes on touch screen. Due to energy dispersion of acoustic signals and absorption of propagation medium, acoustic signals would attenuate during the propagation, which can be utilized by an adversary to perform the keystroke localization. Specifically, an acoustic signal (e.g., a $20kHz$ acoustic signal which is inaudible for humans) is first emitted from the speaker of an adversary's smartphone with energy I_e, then propagates through a distance d, and finally is received by a microphone of the adversary's smartphone with energy I_r. The attenuation between the two acoustic signals is [16]:

$$I_r = I_e \frac{k}{d} e^{\alpha d}, \tag{1}$$

where k is a normalization coefficient, α is the attenuation coefficient. The attenuation coefficient α is affected by the frequency of acoustic signals, temperature, relative humidity, atmosphere, etc. Usually, these impact factors in the ambient environment remain stable for an attack scenario, which hence leads to a constant α [17].

When a victim keystrokes a key on QWERTY keyboard of touch screen, the adversary can obtain two propagated distances $d_1 = d_e + d_{r_1}$ and $d_2 = d_e + d_{r_2}$ of acoustic signals based on Eq. (1), which propagate from a speaker, then reflected by the victim's keystroke finger, and finally to two microphones, Mic_1 and Mic_2, respectively, as shown in Fig. 1. Given the relative positions of the speaker as well as two microphones on the smartphone, we can determine the distances between the keystroke finger and two microphones d_{r_1} and d_{r_2} through the ellipse-based method [10], [12]. Based on the distances d_{r_1} and d_{r_2}, the adversary can construct two circles, whose centers are Mic_1 and Mic_2 with radii of d_{r_1} and d_{r_2} respectively. Then, the adversary can obtain a unique valid intersection point, i.e., the position of keystroke finger relative to microphones of the adversary's smartphone.

In the attack scenario, assume an adversary places his/her smartphone in the same plane of a victim's smartphone on purpose. If the relative position between the two smartphones, D, can be measured, the adversary can identify an exact character for each keystroke. Since the victim's smartphone usually remains static relative to the adversary's smartphone, we can estimate the distance D through measuring time-of-arrival leveraging acoustic beamforming technique [18].

III. System Design

In this section, we describe the system design of $KeyListener$, which performs the keystroke eavesdropping attack on QWERTY keyboard of touch screen leveraging the audio devices on a smartphone.

A. System Overview

Fig. 2 shows the architecture of $KeyListener$. First, $KeyListener$ mitigates multipath reflections in received acoustic signals so as to extract signals reflected from input behaviors, and further segments received acoustic signals into each input behavior window based on Doppler effect. Then, based on the attenuation of acoustic signals, $KeyListener$ localizes finger keystrokes. Since ambient noises would interfere received acoustic signals, each keystroke can only be localized to a position area, i.e., *keystroke range*. To improve the accuracy of keystroke localization, $KeyListener$ further tracks the distance and direction of a finger movement between adjacent keystrokes to construct a *movement sector*, which depicts all potential trajectories of the finger movement. By intersecting the movement sector with keystroke range, the localized keystroke area can be reduced. Since a keystroke area covers one or multiple keys, $KeyListener$ finally utilizes a binary tree-based search approach to infer a keystroke sequence for the adversary in a context-aware manner.

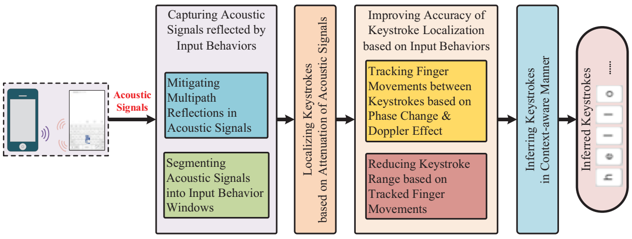

Acoustic Signals
Capturing Acoustic Signals reflected by Input Behaviors
Mitigating Multipath Reflections in Acoustic Signals
Segmenting Acoustic Signals into Input Behavior Windows
Localizing Keystrokes based on Attenuation of Acoustic Signals
Improving Accuracy of Keystroke Localization based on Input Behaviors
Tracking Finger Movements between Keystrokes based on Phase Change & Doppler Effect
Reducing Keystroke Range based on Tracked Finger Movements
Inferring Keystrokes in Context-aware Manner
Inferred Keystrokes
h
e
l
l
o
......

Fig. 2. Architecture of $KeyListener$.

B. Capturing Acoustic Signals reflected by Input Behaviors

1) Mitigating Multipath Reflections in Acoustic Signals: In a real environment, except for reflected acoustic signals from keystroke fingers, acoustic signals received by microphones of an adversary's smartphone usually propagate through multi-path reflections. To precisely localize keystrokes, it is essential to mitigate multipath reflections in received acoustic signals so as to capture signals reflected by keystroke fingers.

The acoustic signal $s(t)$ received by a microphone usually consists Line-Of-Sight (LOS) signal (i.e., the signal directly propagated from speaker to microphones), reflected signals from static and dynamic objects, ambient noises. Since signal reflected from static objects and LOS signal remain stable as time goes on, we employ the signal gradient of received signals [14] proposed in our previous work, which depicts the difference of frequency-domain signals between successive time slots, i.e., $g(t) = s(t) - s(t-1)$, to eliminate these signals.

To further mitigate multipath reflections from other dynamic objects, we adopt FFT power [19] as the energy of received acoustic signal. Based on the signal gradient, the energy of a reflected signal I_r at time t is:

$$I_r(t) = \sum_{f=f_0-\Delta f}^{f_0+\Delta f} g(t), \tag{2}$$

where f_0 is the frequency of pilot tone, Δf is the frequency band induced by a keystroke finger. Usually, the velocity of a finger keystroke is around $0.05 m/s$ [20], which is far less than that of dynamic objects (e.g., normal body movements, etc.) in $[0.85, 3.40] m/s$ [21]. Hence, the acoustic signal induced a finger keystroke is in a narrow frequency band near the pilot tone. To capture such an acoustic signal, we set Δf as $30Hz$ in the attack scenario.

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.

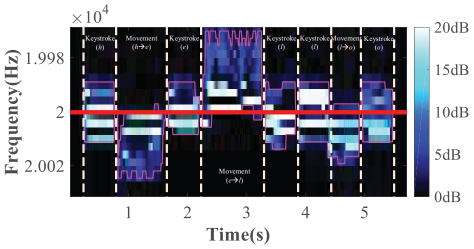


Fig. 3. Time-frequency signals induced by keystrokes and finger movements when a victim inputs '$hello$'.

2) Segmenting Acoustic Signals into Input Behavior Windows: During inputs, there are two kinds of input behaviors, i.e., the keystroke and finger movement between adjacent keystrokes. After acoustic signals reflected by fingers are extracted from received signals, $KeyListener$ needs to further segment the acoustic signals into the two kinds of input behavior windows.

Usually, finger movements induce significant Doppler shifts on acoustic signals, which leads to asymmetric patterns in frequency domain of received acoustic signals. On the contrary, the projection of a keystroke's finger travel on the direction between the victim's finger and adversary's microphone is so small that Doppler shifts induced by keystrokes are insignificant, which produces symmetric patterns in frequency domain of received acoustic signals. Fig. 3 shows time-frequency signals induced by keystrokes and finger movements when a victim inputs '$hello$'. It can be seen that the patterns of acoustic signals in frequency domain on keystroke of 'h', 'e', 'l', 'l' and 'o' are symmetric, while that on finger movements between adjacent keystrokes are asymmetric. Therefore, $KeyListener$ can utilize Doppler effect of acoustic signals to segment received signals into keystroke and finger movement windows.

C. Localizing Keystrokes based on Attenuation of Acoustic Signals

$KeyListener$ utilizes the attenuation of acoustic signals to localize keystrokes as presented in Section II. However, in real environments, ambient noises would have a certain impact on acoustic signals received by microphones. We assume the energy of ambient noises is I_n, which induces the error Δd of measured propagated distance (from the speaker of an adversary's smartphone, then reflected from a victim's keystroke finger, and finally received by a microphone of the adversary's smartphone). Since the propagated distance d of acoustic signal decreases as the energy I_r of acoustic signal received by a microphone increases based on Eq. (1), the attenuation of acoustic signals including ambient noises can be formulated as follows:

$$I_r \pm I_n = I_e \frac{k}{d \mp \Delta d} e^{\alpha(d \mp \Delta d)}. \tag{3}$$

As shown in Fig. 4, errors on two propagated distances of acoustic signals received by Mic_1 and Mic_2 induce two errors Δd_{r_1} and Δd_{r_2}, i.e., errors on distances between the keystroke finger and two microphones, respectively. Based on the two errors, we reconstruct the keystroke localization through the attenuation of acoustic signals. We find that the adversary can only localize to an area for a keystroke, instead of localizing to a precise point, as shown in the blue area of Fig. 4. We denote the area of the localized keystroke as *keystroke range*.

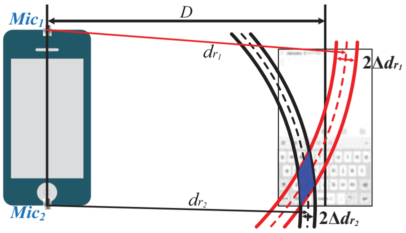


Fig. 4. Example of keystroke range for a localized keystroke.

In real environments, the ambient noise would interfere the energy attenuation of reflected acoustic signals received by microphones, which introduces significant keystroke range in the acoustic signal attenuation-based keystroke localization. Based on Eq. (3), the error Δd increases as the propagated distance d of reflected signal increases, which indicates that the keystroke range increases as the distance D between smartphones of the adversary and victim increases. Therefore, it is necessary to reduce the keystroke range for inconspicuous and precise keystroke eavesdropping attacks.

D. Improving Accuracy of Keystroke Localization based on Input Behaviors

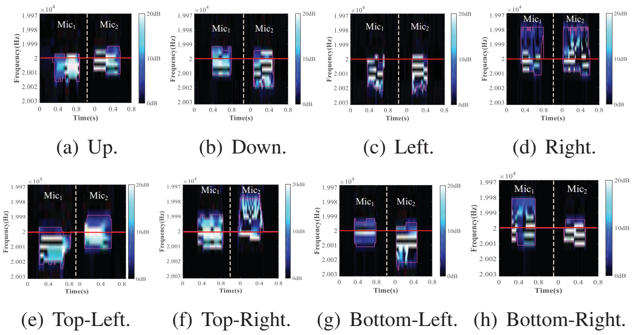
As the analysis above, there still exist significant errors (i.e., keystroke range) in keystroke localization based attenuation of acoustic signals. To improve accuracy of keystroke localization, $KeyListener$ needs to reduce the keystroke range.

1) Tracking Finger Movements based on Phase Change and Doppler Effect: During inputs, there are two kinds of input behaviors, i.e., the keystroke and the movement between adjacent keystrokes. The finger movement between keystrokes can be taken into consideration for improving the accuracy of keystroke localization. Specifically, if the distance and direction of the movement between adjacent keystrokes can be tracked, we can identify the position of the current keystroke based on the previous keystroke position.

Since a finger movement between keystrokes is in a 2-D plane, the distance of finger movement can be tracked through phase change. Specifically, the acoustic signal emitted from the speaker of an adversary's smartphone is $s_e(t) = A\cos(2\pi f_0 t)$, where A and f_0 are the amplitude and frequency of the pilot tone respectively. After propagating through a distance d and reflected from the finger, the acoustic signal received by the microphone of the smartphone is $s_r(t) = A'\cos(2\pi f_0 t - 2\pi f_0 d/c)$, where c is the speed of acoustic signals. To extract the phase change induced by finger movement, we multiply the received signal with the emitted signal, i.e.,

$$\begin{aligned} s_r(t) \times s_e(t) &= A'\cos(2\pi f_0(t - d/c)) \times A\cos(2\pi f_0 t) \\ &= \frac{1}{2} AA'(\cos(-2\pi f_0 d/c) + \cos(2\pi f_0(2t - d/c))). \end{aligned} \tag{4}$$

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.



(a) Up. (b) Down. (c) Left. (d) Right.

(e) Top-Left. (f) Top-Right. (g) Bottom-Left. (h) Bottom-Right.

Fig. 5. Doppler profiles of acoustic signals under eight basic directions of finger movements.

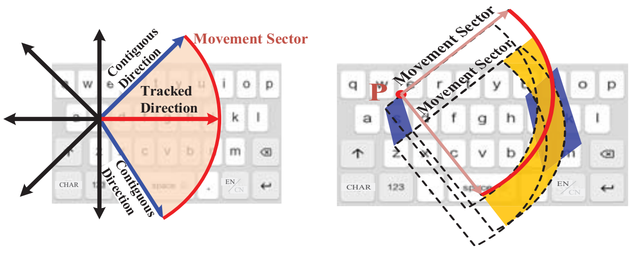
Then, we apply a lowpass filter to remove the high-frequency term, i.e., the second term in Eq. (4). After that, the phase change induced by the finger movement can be extracted as $\phi = -2\pi f_0 d/c$. Since ϕ changes by 2π as the distance of finger movement changes with a wavelength of the acoustic signal $\lambda = c/f$, we only obtain the phase change within 2π instead of the whole phase change induced by the finger movement. Fortunately, the starting and ending positions can be determined based on the attenuation of acoustic signals in advance, so the number of phase cycle can be derived. After that, we can calculate the whole phase change induced by finger movement. Based on the whole phase change, the propagating distance can be measured, i.e.,

$$d = -\frac{\phi_t - \phi_0}{2\pi} \times \frac{c}{f_0}, \tag{5}$$

where ϕ_0 and ϕ_t are phase changes when the finger is at the starting and ending positions of a finger movement respectively.

To track the finger movement between keystrokes, we need to track not only the distance, but also the direction of finger movement. However, phase-based methods cannot achieve satisfactory accuracy in tracking direction of finger movements, and the relative errors are unpredictable. To predict such an error of direction tracking, we utilize Doppler effect of acoustic signals to track the direction of each finger movement. During inputs, the finger can move from a key to another one in various directions. However, it is difficult to enumerate all directions of finger movements during inputs. Hence, we only consider to track eight basic directions of finger movements, i.e., up, down, left, right, top-left, top-right, bottom-left, and bottom-right. Different finger movement directions induce unique Doppler profile of acoustic signals received by microphones. Fig. 5 shows Doppler profiles of acoustic signals received by two microphones under eight basic directions of finger movements. We can see that Doppler shifts induced by eight basic directions are significantly distinguishable from each other. Thus, $KeyListener$ can track an approximate direction of a finger movement, whose errors are in the range of two contiguous directions.

Based on the tracked distance and direction of finger movements, we construct a *movement sector* as shown in the orange area of Fig. 6(a). The position of a keystroke finger after the movement would lie on a point of the movement sector's arc, as shown in the red arc of Fig. 6(a). We can utilize the movement sector to reduce the keystroke range in acoustic signal attenuation-based keystroke localization.



(a) Movement sector. (b) Keystroke range reduction.

Fig. 6. Illustration of improving the accuracy of keystroke localization based on input behaviors.

2) Reducing Keystroke Range based on Tracked Finger Movements: Fig. 6(b) illustrates the accuracy improvement of keystroke localization based on input behaviors. Assume a victim first clicks a key s and then clicks another key j on QWERTY keyboard of touch screen. The adversary first localizes the two keystrokes based on the attenuation of received acoustic signals, and obtains two keystroke ranges $KR(s)$ and $KR(j)$ covering s and j respectively, as shown in the two blue areas of Fig. 6(b).

Then, to depict the finger movement between keystrokes, a movement sector is determined through phase change and Doppler effect. However, since we can only localize the keystroke to a keystroke range $KR(s)$, the center of movement sector could be an arbitrary point in $KR(s)$. For example, after a finger moves between keystrokes on s and j, a point P in $KR(s)$ would move to an arbitrary point in the arc of movement sector, as shown in the red arc of Fig. 6(b). The movements of all points in $KR(s)$ are similar to that of the point P. Hence, an area S after the movement is constructed as shown in the yellow area of Fig. 6(b), which contains all potential points after the movement, i.e., arcs of all sectors whose centers lie in $KR(s)$. Finally, $KeyListener$ intersects S with $KR(j)$ to reduce the keystroke range for improving the accuracy of keystroke localization.

In the proposed keystroke localization, the keystroke range merely depends on the attenuation of acoustic signal reflected by the keystroke, and the movement sector only depends on the finger movement between two adjacent keystrokes through phase change and Doppler effect of acoustic signals, which are both independent of other keystrokes and movements. Therefore, there is no cumulative error during the whole keystroke localization.

E. Inferring Keystrokes in Context-aware Manner

Through the approach above, $KeyListener$ can only localize a keystroke to a keystroke range, which usually covers one or multiple keys on a keyboard. We denote keys covered by a keystroke range as *character candidates* of the keystroke. Since the input of a victim is usually meaningful, the input can be inferred with a dictionary in a context-aware manner.

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.

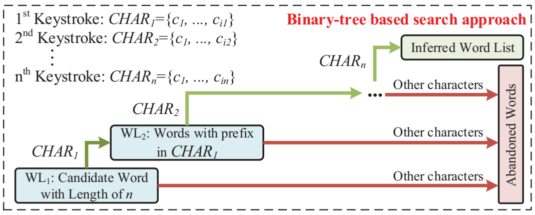

1st Keystroke: CHAR1={c1, ..., ci1}
2nd Keystroke: CHAR2={c1, ..., ci2}
nth Keystroke: CHARn={c1, ..., cin}
Binary-tree based search approach
Inferred Word List
CHARn
Other characters
CHAR2
WL2: Words with prefix in CHAR1
Other characters
CHAR1
WL1: Candidate Word with Length of n
Other characters
Abandoned Words

Fig. 7. Illustration of context-aware keystroke inference.

We present a binary tree-based search approach to generate all possible inference results, i.e., *word candidates*, for adversaries, as shown in Fig. 7. Assume there are n keystrokes in a keystroke sequence. For the i^{th} keystroke in the sequence, there is a set, $CHAR_i$, including all the character candidates of the i^{th} keystroke. From the given dictionary, $KeyListener$ first generates a word list WL_1, in which the first character of the words is in $CHAR_1$. Then, from the WL_1, $KeyListener$ searches all words whose second character is in $CHAR_2$, to generate a word list WL_2. By analogy, to generate a word list WL_i, $KeyListener$ finds the words whose i^{th} character is in $CHAR_i$, from the word list WL_{i-1}. Until $CHAR_n$ is used for searching, $KeyListener$ can generate a word list WL_n including all possible word candidates for adversaries. Through such an approach, $KeyListener$ can infer keystrokes in a context-aware manner.

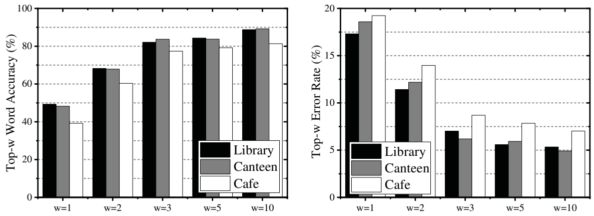
The computational complexity of the approach is $O(ncL)$, where n is the number of keystrokes in a keystroke sequence, c is the number of character candidates in a keystroke, and L is the length of the dictionary. Since the values of c and L are usually small, the computational complexity is approximately $O(n)$. Therefore, the proposed keystroke inference is lightweight and computationally efficient for smartphones.

IV. Evaluation

In this section, we evaluate the performance of $KeyListener$ under collected data in real environments.

A. Experimental Setup & Methodology

We implement $KeyListener$ on a Galaxy S4 with Android 5.1.1 as the smartphone of an adversary. The pilot tone of acoustic signals emitted from the smartphone is set as $20kHz$. We recruit 24 volunteers, including 12 males and 12 females whose ages range in $[18, 45]$, to conduct our experiments. The volunteers are required to use one of four smartphones with different screen sizes, i.e., a 4.7-inches iPhone 7, a 5.2-inches Huawei P7, a 5.5-inches iPhone 7 Plus and a 7.0-inches Huawei Honor X2. We conduct the experiments in three real environments, i.e., sitting in a library (quiet and a few people walking in the surrounding), sitting in a canteen (very noisy and many people walking in the surrounding), and queuing in a cafe (less noisy and some people walking in the surrounding). In each environment, the adversary's smartphone is placed in three different positions relative to the victim, i.e., left, right, and opposite respectively. The distance between smartphones of the adversary and victim ranges in $[45, 60]cm$. We select 5,000 most frequent words [22] for the volunteers, and each volunteer is required to randomly select 500 words from them for inputting. The volunteers are unaware of the experimental purposes, and thus hold the smartphones and input on touch screen following their own habits.



(a) Top-w word accuracy.

(b) Top-w error rate.

Fig. 8. Overall performance of $KeyListener$ under different environments.

To evaluate the performance of $KeyListener$, we define several metrics as follows.

Top-w Word Accuracy. Given w inferred word candidates, the top-w word accuracy is defined to measure the overall performance of keystroke inference. Assuming the number of texts during inputs is k, the top-w word accuracy is defined as $A^w = \frac{i}{k}$, where i is the number of inferences in which the top-w word candidates contain the ground truth.

Number of Word Candidates. Assume a victim inputs n keystrokes as a word. For the i^{th} keystroke, $KeyListener$ provides c_i character candidates. For the inputted word, the *number of word candidates* is defined as $W^n = \Pi_{i=1}^{n} c_i$.

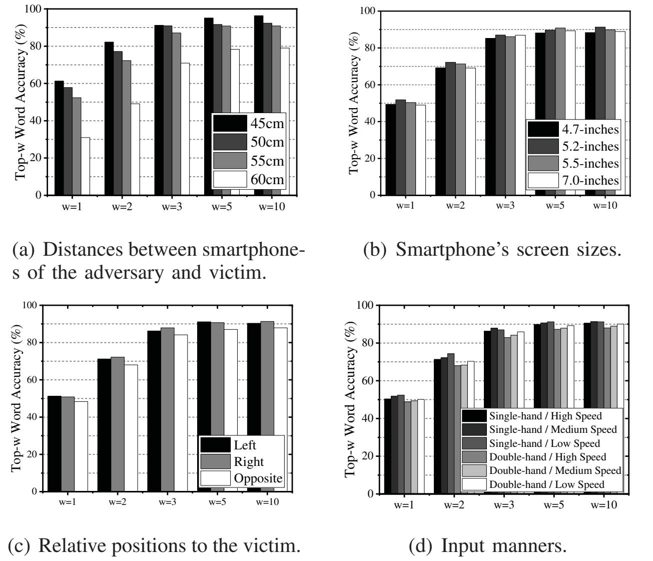
Confusion Matrix. Each row and each column of the matrix represent the actual keystroke and the identified keystroke of $KeyListener$ respectively. The i^{th}-row and j^{th}-column entry of the matrix shows the percentage of samples that are identified as the j^{th} key while actually are the i^{th} key for all samples that actually are the i^{th} key.

F1-score. *F1-score* is a measure combining precision and recall, which evaluates the performance of single keystroke identification. F1-score is defined as $F1\text{-}score_k = 2 \times \frac{P_k \times R_k}{P_k + R_k}$, where P_k and R_k are the precision and recall of identifying key k respectively. *Precision* of identifying key k is defined as $P_k = m_k^T / (m_k^T + m_k^F)$, where m_k^T is the number of keystrokes correctly identified as the key k, m_k^F is the number of keystrokes mistakenly identified as the key k while are actually other keys. On the other hand, given n_k keystrokes of a key k, *Recall* of identifying key k is defined as $R_k = m_k^T / n_k$.

B. Overall Performance of $KeyListener$

Overall Performance. We first evaluate the overall performance of $KeyListener$. For each keystroke inference, $KeyListener$ selects top-w word candidates based on the word frequency of all word candidates. Fig. 8(a) shows top-1 to top-10 word accuracies of $KeyListener$ under three real environments. We can see that the top-1 word accuracies in the library and canteen are both approaching 50%, while that in the cafe is around 40%. The top-10 word accuracies in the library and canteen can both approach 90%, while that in the cafe is 81.3%. The performance of $KeyListener$ in the cafe is a little lower than that in other two environments. This is

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.



(a) Distances between smartphones of the adversary and victim.

(b) Smartphone's screen sizes.

(c) Relative positions to the victim.

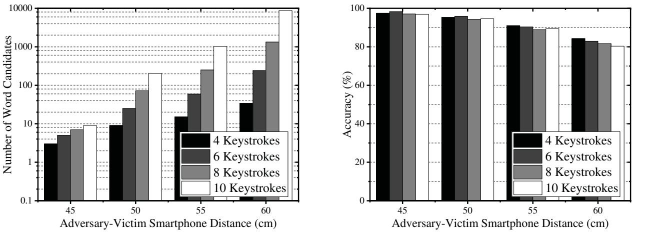
(d) Input manners.

Fig. 9. Top-w word accuracy of $KeyListener$ under different impact factors.

because the victim queuing in the cafe intends to prevent others from seeing his/her inputs on touch screen, which depicts an obstruction between the adversary and victim. We also evaluate top-w error rate of $KeyListener$ under three real environments. The result is similar to the previous analysis, as shown in Fig 8(b). In general, $KeyListener$ can achieve satisfactory performance of keystroke inference in the three real environments.

Impact of Distance between Smartphones of Adversary and Victim. Since we utilize propagation characteristics of acoustic signals to identify keystrokes on touch screen, the distance between smartphones of the adversary and victim, i.e., *adversary-victim smartphone distance*, would have a certain impact on the performance of $KeyListener$. Fig. 9(a) shows the top-1 to top-10 accuracies of $KeyListener$ under different adversary-victim smartphone distances. We can observe that the top-w word accuracies of $KeyListener$ all decrease as the distance increases. When the distance increases from $45cm$ to $60cm$, the top-2 word accuracy of $KeyListener$ decreases from 82.2% to 49.1%. This is because that ambient noises have a significant impact on the measurement of acoustic signal energy as the distance increases. However, as w increases from 2 to 10, the accuracy of $KeyListener$ increases from 49.1% to 79.7% when the distance is $60cm$, which indicates $KeyListener$ can achieve acceptable performance under different adversary-victim smartphone distances.

Impact of Smartphone's Screen Size. Various screen sizes of smartphones lead to different keyboard sizes on touch screen. Thus, we evaluate the performance of $KeyListener$ under four different screen sizes. Fig. 9(b) shows the top-1 to top-10 word accuracies of $KeyListener$ under four different screen sizes. It can be observed that the top-w word accuracies are similar under four different screen sizes. This is because as the increase of screen sizes, both the key size and the width of keyboard increase. Although the increase of key size leads to higher keystroke localization accuracy, the increase of keyboard width depicts larger distance between some keys and the adversary's smartphone, which induces larger errors of keystroke localization.



(a) Number of word candidates.

(b) Accuracy for 9-key keyboard.

Fig. 10. Performance of $KeyListener$ without context-aware inference under different impact factors.

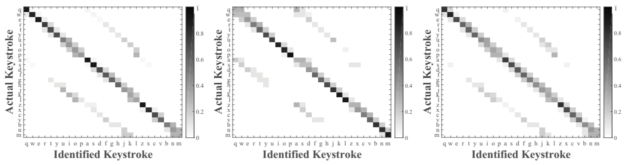
Impact of Relative Position. We also conduct an experiment to evaluate the performance of $KeyListener$ under different relative positions between smartphones of the adversary and victim, i.e., left, right, opposite. Fig. 9(c) shows the top-1 to top-10 word accuracy of $KeyListener$ under different relative positions. We can observe that the top-w word accuracies under three relative positions are all similar. Although the top-w word accuracy under the opposite situation is a little lower than that under other two positions, the difference is not significant. For example, the top-10 word accuracy under opposite position is only 2.47% and 3.4% lower than that under left and right positions respectively.

Impact of Input Manner. In our experiment, the volunteers input following their own habits. Hence, except for single-hand input on smartphones, some volunteers are used to inputting in a double-hand manner. Also, typing speeds are various for different volunteers. We evaluate the performance of $KeyListener$ under different input manners, including single-hand and double-hand manners, as well as different typing speeds (i.e., high speed: larger than 130 keystrokes per minute (KPM), medium speed: in the range of $[90, 130]$ KPM, and low speed: less than 90 KPM). Fig. 9(d) shows top-1 to top-10 word accuracies of $KeyListener$ under different input manners. It can be seen that the top-w word accuracy under a double-hand manner is a little lower than that under a single-hand manner. This is because there are fewer finger movements under a double-hand input than that under a single-hand input. However, the difference between top-w word accuracies of single-hand and double-hand input manner is not significant. For example, top-5 word accuracies under the two input manners are 90.7% and 87.8% respectively. Moreover, we can see that the top-w word accuracies under different typing speeds are similar. For example, the differences of the top-5 word accuracy between high and low speed are 1.4% and 2.0% under single-hand and double-hand manner respectively.

C. Performance of $KeyListener$ without Context-aware Inference

Except for meaningful texts, a victim sometimes inputs random texts (e.g., password), which cannot be inferred through

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.



(a) Left. (b) Right. (c) Opposite.

Fig. 11. Confusion matrix of single keystroke identification under different positions when the adversary-victim smartphone distance is $55cm$.

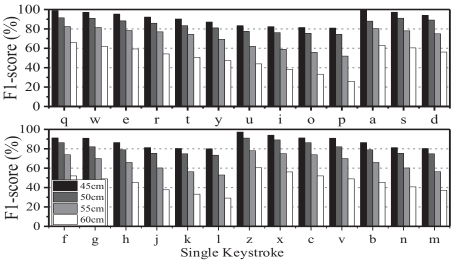


Fig. 12. F1-score of single keystroke identification under different adversary-victim smartphone distances.

a dictionary. To analyze the performance of $KeyListener$ under random input, we conduct an experiment, in which $KeyListener$ regards victims' inputs as random inputs, and only provides all possible combinations, i.e., *word candidates*, as identified keystrokes without context-aware inference.

Fig. 10(a) shows the number of word candidates under different adversary-victim smartphone distances and keystroke lengths. It can be observed that the numbers of word candidates under different keystroke lengths all decrease rapidly as the decrease of adversary-victim smartphone distance. For example, when the distance decreases from $60cm$ to $55cm$, the number of word candidates decreases from 8,712 to 1,023 under 10 keystrokes. This is because the keystroke range significantly decreases as the smartphone distance decreases. Also, the number of word candidates increases as the keystroke lengths increase. Usually, the length of a password is less than 8-digit. Under such a situation, $KeyListener$ can provide less than 300 word candidates when the distance is less than $55cm$. As a keystroke eavesdropping attack, $KeyListener$ can achieve satisfactory performance in keystroke inference.

Recently, most passwords in mobile devices are inputted through 9-key PIN keyboard. To evaluate the performance of $KeyListener$ in password eavesdropping attack, we implement a special version of $KeyListenter$, which localizes each keystroke on 9-key PIN keyboard of touch screen. Fig. 10(b) shows the accuracy of the special $KeyListener$ under different adversary-victim smartphone distances and different lengths of keystrokes. We can see that the special version of $KeyListener$ can achieve above 80% accuracy for all smartphone distances and lengths of keystrokes. This is because the size of a key on 9-key PIN keyboard is far larger than that on QWERTY keyboard. The keystroke range can precisely cover one key during each keystroke localization.

D. Performance of Single Keystroke Identification

We evaluate the single keystroke identification performance of $KeyListener$ under different relative positions between smartphones of adversary and victim. Under each relative position, each volunteer is required to input 100 characters from 26 English characters following their own habits.

Fig. 11 presents the confusion matrix of the single keystroke identification under three relative positions, i.e., left, right and opposite, when the distance between smartphones of the adversary and victim is $60cm$. It can be seen from Fig. 11(a) that keystrokes on some keys have a lower keystroke identification accuracy, such as 'k', 'p', etc., and the average identification accuracy of keystrokes on these keys is 25.1%. This is because in this attack scenario, the adversary's smartphone is placed in the left of the victim's smartphone, i.e., these keys are far from the adversary's microphones, which leads to low identification accuracy. On the other hand, keystrokes on other keys, such as 'a', 's', etc., have much higher accuracy, and the average identification accuracy of keystrokes on these keys can achieve 96.2%. This is because these keys like 'a' and 's' are much near the adversary's microphones, which depicts higher identification accuracy. Fig. 11(b) and 11(c) show the confusion matrices of the single keystroke identification under the right and opposite positions respectively, which present similar results as that in Fig. 11(a).

We further evaluate the impact of distance between smartphones of adversary and victim on the single keystroke identification. In this evaluation, the adversary's smartphone is placed on the left of the victim's smartphone. Fig. 12 shows F1-score of single keystroke identification for 26 characters under four adversary-victim smartphone distances. We can observe that F1-scores of single keystroke identification decrease as the distance between smartphones of adversary and victim increases. When the distance is $60cm$, the average F1-score for the 26 characters is 48.0%. However, the average F1-score rapidly increases to 68.6% as the distance decreases to $55cm$ which is still an inconspicuous distance for the attack. This indicates $KeyListener$ can achieve good performance at an inconspicuous distance. Meanwhile, it can be seen that F1-scores of single keystroke identification on keys far from the adversary are significantly lower than that on keys near the adversary, which is consistent with the analysis above. We also evaluate F1-scores of single keystroke identification in right and opposite positions, which present similar results.

V. Related Work

In this section, we review existing works about side-channel attacks on inputs and acoustic signal-based applications.

Attacks on Input of Physical Keyboard. Currently, most active research efforts [3]–[6] focus on keystroke eavesdropping attacks on physical keyboards. [3], [4], [6] utilize the acoustic emanation of keystroke sounds to identify victims' keystrokes. However, the audible sounds induced by keystrokes are easily affected by ambient noises. [4], [5] utilize motion sensors on smartwatches to localize keystrokes on physical keyboards, but the adversary is required to involve a direct eavesdropping attack, i.e., have access to victims' smartwatches, which usually arouses victims' vigilances.

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.

Attacks on Input of Touch Screen. Recently, more people input private information on touch screen of mobile devices. This motivates some works studying on keystroke eavesdropping attacks on touch screen. Some works [7], [8] reveal that motion sensors on victims' smartphones would leak their inputs on touch screens. However, all these works are direct eavesdropping attacks, i.e., require sensor data on victims' devices compromised to provide side-channel information about keystrokes for the adversary, which limits the impact of such attacks. More recent work [9] utilizes keystroke patterns in CSI of WiFi signals to detect victims' inputs on PIN keyboard of touch screen. Although this work is an indirect eavesdropping attack, its application scenario is constrained to the coverage of WiFi infrastructure, and the attack is only for 9-key PIN keyboard instead of QWERTY keyboard.

Acoustic Signal-based Applications. Recently, acoustic sensing attracts considerable attention. Previous studies utilize acoustic signals for gesture recognition [21], [23], tracking [10]–[12], and even user authentication [14], [15]. Among these works, [10]–[12] propose acoustic-based techniques to track a continuous finger movement in a 2-D plane. However, a user's input behaviors, including keystrokes and finger movements between adjacent keystrokes, are in a 3-D space. Hence, it is difficult to directly adopt these works for keystroke localization through tracking finger movements.

Unlike existing works, our work propose to infer keystrokes on QWERTY keyboard of touch screen through propagation characteristics of acoustic signals, which is an indirect eavesdropping attack and robust to various environments, as well as require no additional infrastructures.

VI. Conclusions

In this paper, we demonstrate acoustic signals from a smartphone can be used to implement a side-channel attack, $KeyListener$, which can infer keystrokes on QWERTY keyboard of touch screen. In particular, we first investigate the attenuation of acoustic signals, and find that the attenuation of acoustic signals can be used to localize each keystroke during inputs. Then, to improve the accuracy of keystroke localization, we track the finger movements during inputs through phase change and Doppler effect of acoustic signals to reduce errors induced by ambient noises. Finally, we present a binary tree-based search approach to infer the victim's continuous keystrokes in a context-aware manner. Extensive experiments demonstrate that $KeyListener$ can achieve satisfactory performance on not only meaningful input inference in a context-aware manner, but also random input identification without inference in real environments. Moving forward, we are interested in further relaxing the relative position between smartphones of adversary and victim so that the adversary can perform the keystroke eavesdropping attack in more scenarios.

Acknowledgment

This research is partially sponsored by National Key R&D Program of China (No. 2017YFC0803700), NSFC (No. 61772338, 61420106010, 61772341, 61472254) and STSCM (No. 18511103002). This work is also supported by the Program for Changjiang Young Scholars in University of China, the Program for China Top Young Talents, and the Program for Shanghai Top Young Talents. This work is also supported by China Scholarship Council.

References

[1] B. of Governors of the Federal Reserve System, "Consumers and mobile financial services 2016," [Online]: https://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201603.pdf, 2016.

[2] Statista, "Number of monthly active whatsapp users worldwide from april 2013 to december 2017 (in millions)," [Online]: https://www.statista.com/statistics/260819/number-of-monthly-active-whatsapp-users/, 2018.

[3] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping Keystrokes with Mm-level Audio Ranging on a Single Phone," in *Proc. ACM Mobicom'15*, Paris, France, 2015.

[4] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "When Good Becomes Evil: Keystroke Inference with Smartwatch," in *Proc. ACM CCS'15*, Denver, CO, USA, 2015.

[5] H. Wang, T. T. Lai, and R. R. Choudhury, "MoLe: Motion Leaks Through Smartwatch Sensors," in *Proc. ACM Mobicom'15*, Paris, France, 2015.

[6] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free Attacks Using Keyboard Acoustic Emanations," in *Proc. ACM CCS '14*, Scottsdale, AZ, USA, 2014.

[7] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: Password Inference Using Accelerometers on Smartphones," in *Proc. ACM HotMobile '12*, San Diego, CA, USA, 2012.

[8] L. Cai and H. Chen, "TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion," in *Proc. USENIX HotSec'11*, San Francisco, CA, USA, 2011.

[9] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When CSI Meets Public WiFi: Inferring Your Mobile Phone Password via WiFi Signals," in *Proc. ACM CCS'16*, Vienna, Austria, 2016.

[10] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. ACM Mobicom'16*, New York, USA, 2016.

[11] S. Yun, Y. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-Grained Acoustic-based Device-Free Tracking," in *Proc. ACM Mobisys'17*, Niagara Falls, NY, USA, 2017.

[12] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proc. ACM CHI'16*, Santa Clara, California, USA, 2016.

[13] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proc. ACM CCS'17*, Dallas, TX, USA, 2017.

[14] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, Y. Liu, and M. Li, "Lippass: Lip reading-based user authentication on smartphones leveraging acoustic signals," in *Proc. IEEE INFOCOM'18*, Honolulu, HI, USA, 2018.

[15] L. Zhang, S. Tan, J. Yang, and Y. Chen, "Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones," in *Proc. ACM CCS'16*, Vienna, Austria, 2016.

[16] ISO, *Acoustics-Attenuation of sound during propagation outdoors - Part 1: Calculation of the absorption of sound by the atmosphere*, Std., 1993.

[17] N. P. Laboratory, "The speed and attenuation of sound," [Online]: http://www.kayelaby.npl.co.uk/general_physics/2_4/2_4_1.html, 2018.

[18] P. Pertilä and A. Tinakari, "Time-of-arrival estimation for blind beamforming," in *Proc. IEEE ICDSP'13*, 2013.

[19] A. N. Mortensen and G. L. Johnson, "A power system digital harmonic analyzer," *IEEE Transactions on Instrumentation and Measurement*, vol. 37, no. 4, pp. 537–540, 1988.

[20] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing*, vol. 11, no. 2, pp. 1565–1573, 2011.

[21] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: Using the doppler effect to sense gestures," in *Proc. ACM CHI '12*, Austin, Texas, USA, 2012.

[22] C. BYU, "Word frequency: based on 450 million word coca corpus," [Online]: https://www.wordfrequency.info, 2018.

[23] S. Yun, Y. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proc. ACM MobiSys '15*, Florence, Italy, 2015.

Authorized licensed use limited to: Rutgers University. Downloaded on September 13,2020 at 17:01:00 UTC from IEEE Xplore. Restrictions apply.