Snooping Keystrokes with mm-level Audio Ranging on a Single Phone

Jian Liu[†], Yan Wang[†], Gorkem Kar[*], Yingying Chen[†], Jie Yang[§], Marco Gruteser[*]
[†]Stevens Institute of Technology, Hoboken, NJ 07030, USA
[*]Rutgers University, North Brunswick, NJ 08902, USA
[§]Florida State University, Tallahassee, FL 32306, USA
[†]{jliu28, ywang48, yingying.chen}@stevens.edu, [*]{gkar87, gruteser}@winlab.rutgers.edu,
[§]jyang5@fsu.edu

ABSTRACT

This paper explores the limits of audio ranging on mobile devices in the context of a keystroke snooping scenario. Acoustic keystroke snooping is challenging because it requires distinguishing and labeling sounds generated by tens of keys in very close proximity. Existing work on acoustic keystroke recognition relies on training with labeled data, linguistic context, or multiple phones placed around a keyboard — requirements that limit usefulness in an adversarial context.

In this work, we show that mobile audio hardware advances can be exploited to discriminate mm-level position differences and that this makes it feasible to locate the origin of keystrokes from only a single phone behind the keyboard. The technique clusters keystrokes using time-difference of arrival measurements as well as acoustic features to identify multiple strokes of the same key. It then computes the origin of these sounds precise enough to identify and label each key. By locating keystrokes this technique avoids the need for labeled training data or linguistic context. Experiments with three types of keyboards and off-the-shelf smartphones demonstrate scenarios where our system can recover 94% of keystrokes, which to our knowledge, is the first single-device technique that enables acoustic snooping of passwords.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Unauthorized access

General Terms

Security

Keywords

Keystroke Snooping, Time Difference of Arrival (TDoA), Audio Ranging, Single Phone

1. INTRODUCTION

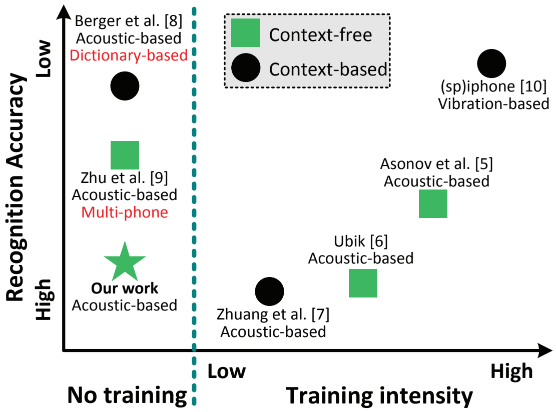


Figure 1: Design Space: comparing to related work.

Mobile device hardware is increasingly supporting high definition audio capabilities targeted at audiophiles. In particular, this includes microphone arrays for stereo recording and noise cancellation as well as 4x improvement in audio sampling rates. For example, the Samsung Galaxy Note 3 includes three microphones and its audio chips are capable of 192kHz playback and recording. One can debate whether all these advances actually lead to improvements in music playback and audio recording quality that are perceivable by the human auditory system and not all these hardware capabilities are currently made available by drivers and operating system software. Such advances, however, could have a significant impact on the accuracy of audio ranging and localization.

Audio localization has been explored with mobile devices to achieve centimeter level accuracy in various applications, such as phone-to-phone ranging and 3D localization [1, 2], mobile motion games [3], and driver phone use detection [4]. Will advanced mobile audio hardware capabilities lead to order of magnitude improvements and let us achieve mm-level accuracy or do the limiting factors lie in multipath distortions and the accuracy of signal detection techniques?

We explore these questions in the context of keystroke snooping, a particularly challenging localization technique and one with important security implications. To eavesdrop on keystrokes, an adversary can inconspicuously leave a phone near a keyboard of the target user. Or, an adversary can co-opt the target users own phone, for example by adding malware into an app with microphone access. Keystroke snooping is particularly challenging because of the large number of different keys to distinguish and

MobiCom'15, September 7-11, 2015, Paris, France.
© 2015 ACM. ISBN 978-1-4503-3543-0/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2789168.2790122.

the small cm-level separation between individual keys. It has important security implications because using keyboard is still an important way of entering sensitive information into computing systems and crucially, passwords remain the primary means to authenticate with remote systems, including financial- and health-related services. Besides these security and privacy breaches there is also potential to create improved input methods for mobile devices that do not directly require typing on the confined mobile screens. Additionally, the proposed solution also has the capacity to facilitate other applications that benefit from fine-grained localization, such as extending interactions with the touch screen of a mobile device to its adjacent surfaces for controlling music players or video games; tracking speakers in multiparty conversations in a meeting room; and locating trapped disaster victims. The proposed audio ranging solution leveraging geometry based information (i.e., time difference of arrival) and unique acoustic characteristics extracted from potential sound sources could deal with many limitations of mobile devices, such as only two stereo recording microphones, limited sampling rate, and restrained distance between two microphones.

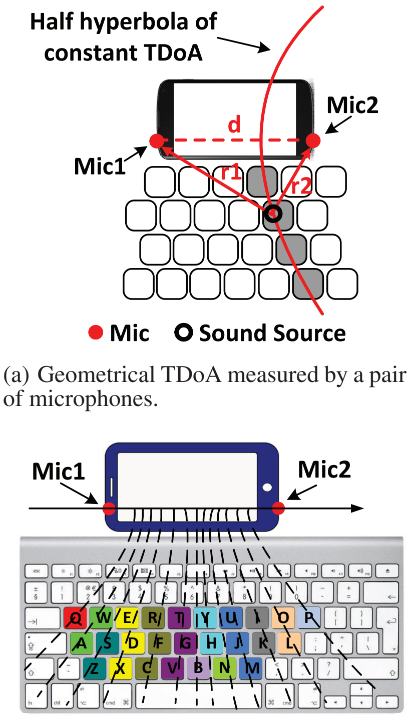
Prior work. Existing research has already recognized the significance of this question and found limited potential to recover keystrokes from audio recordings. In particular, Asonov *et al.* [5] conducted an initial study that observed that each key produces unique acoustic emanations and designed a supervised learning method to recognize individual keys. UbiK [6] improves accuracy for keystrokes on solid surfaces (i.e., a paper keyboard on a table) fingerprinting acoustic differences due to multi-path fading. These approaches require extensive labeled training data from the exact keyboard setting to learn the acoustic profiles for each key, which can be challenging to obtain in adversarial scenarios. Later, Zhuang *et al.* [7] propose to add language constraints to improve recognition accuracy. Berger *et al.* [8] further trades off training requirements for accuracy through a dictionary-based approach that leverages the similarity of acoustic signals from nearby keys. Such methods, however, improve keystroke recognition only for natural language and fail for strong passwords composed of random characters. Zhu *et al.* [9] proposes to utilize microphones on three phones to identify keystrokes of a nearby keyboard based on time difference of arrival (TDoA) measurements. The requirements of three collaborating phones and the achieved moderate accuracy make their approach less feasible for real attack scenarios. There is also a related line of work that has explored vibrations sensing of keystrokes using accelerometers such as (e.g., [10]). The accuracy of such approaches generally remains lower than that of audio sensing. Figure 1 illustrates the design space and the results offered by existing work. Due to the limited accuracy, the use of multiple recording devices, the need for linguistic context, or training with extensive labeled data, none of these techniques can easily be applied to snoop on passwords.

Approach. This paper demonstrates that the mobile audio hardware advances can indeed be exploited for high accuracy mm-level ranging and that practical scenarios exist where it is possible to localize keystroke sounds with an accuracy sufficient to snoop on passwords. It explores a novel point in the keystroke recognition design space by showing the feasibility of keystroke snooping that is (i) training-free, (ii) context-free, (iii) based on single phone. The approach is training-free because it does not require a-priori labeled training data, which is often difficult to obtain for an adversary. Comparing to the training-based keystroke recovering solutions, e.g., using labeled keystroke data to train a neural network to recognize subsequent keystrokes [5], our work develops unsupervised algorithms without any labeled data to cluster a set of keystrokes. The approach is context-free because it does not require on any linguistic models such as letter, letter sequence (n-gram), or word likelihoods and can therefore be applied to random key sequences such as passwords. And the approach is based on a single phone because it does not require multiple phones or recording devices to be placed around the keyboard; it only relies on two microphones in a single phone.

Our work achieves this by discriminating keystrokes based on the time-difference-of-arrival (TDoA) of the keystroke sound at the two phone microphones and by refining such estimates using acoustic differences in the sound emitted by each key. For certain placements of the phone, relative to the keyboard, there exist measurable differences in TDoA value between most keys. Different from general acoustic TDoA localization approaches which require at least three distributed microphones, our work only uses two microphones with highly constrained distances on a single phone, which produce a limited range of single-dimensional TDoA measurements for locating the keystroke. While a single TDoA measurement will not allow determining a unique 2D location for the keystroke, it does restrict the possible locations for this keystroke to a hyperbola. Given multiple keystrokes of the same key and information about the keyboard geometry, this hyperbola can be placed with mm-level precision so that it uniquely identifies a key. To obtain multiple audio samples of the same key, even under random typing, the approach clusters keystrokes based on the observed TDoA and mel-frequency cepstral coefficients (MFCCs), which capture (slightly) different acoustic signatures of each keystroke such as those due to physical imperfections across keys. Since the acoustic signatures are only used for improving clustering, there is no need for training of acoustic signatures. Further, since the final TDoA values describe relative locations, they can be directly used to label keystrokes if the keyboard geometry and phone position is known (e.g., keyboard with phone/tablet stand) or if it can be inferred (i.e., enough keystrokes can be observed to derive the key layout). The labeling process only requires finding a best match between the measured TDoA and the expected TDoA for each key, given the geometry and placement.

The contributions of our work are summarized as follows:

- We demonstrate that a single off-the-shelf phone can recover keystrokes by exploiting mm-level acoustic ranging and fine-grained acoustic features. We develop a training-free approach on a smartphone that does not require a linguistic model, allowing it to recover random keystrokes (e.g., random passwords).
- We exploit recent mobile audio hardware advances to stretch the limits towards mm-level audio localization accuracy.
- We develop a keystroke snooping framework, which leverages hardware advances (i.e., stereo recording with high sampling rate) of off-the-shelf mobile devices to narrow down possible positions of a keystroke. The framework further exploits the geometry-based information (i.e., TDoA) and unique acoustic signatures of keystrokes to ping-point their positions on a keyboard.
- We conduct extensive experiments with three kinds of keyboards to show that an off-the-shelf phone with $48kHz$ microphone sampling rate can accurately identify a set of keystrokes with over 85% accuracy. With higher sampling rate (e.g., $192kHz$), the accuracy could be increased to over 94% accuracy. Even for a single keystroke input, our system can achieve 97% accuracy of identifying keystrokes in the



(a) Geometrical TDoA measured by a pair of microphones.

(b) Theoretical key groups and corresponding half hyperbolas.

Figure 2: Illustration of the geometrical TDoA on a single-phone and theoretical key groups.

top-3 candidate keys with $48kHz$ sampling rate. We believe that these are the first results to raise serious concerns about acoustic password snooping.

2. ATTACK MODEL & LIMITS OF TDOA WITH A SINGLE PHONE

In this section, we introduce the attack scenarios and the rationale for their selection. We also analyze key factors affecting the accuracy of keystroke snooping when using a single phone and define basic concepts.

2.1 Attack Model

We consider a scenario where an adversary seeks to identify each entered character in a sequence of keystrokes from the acoustic signal generated by depressing a key ("typing sound"). We assume that adversaries have the access to stereo audio recordings from a single mobile device (e.g., smartphone or tablet) that is placed close to a victim's keyboard. Two representative scenarios where this is plausible are: (1) the adversary inconspicuously leaves a prepared recording device next to the victim's keyboard, perhaps in a confined setting such as library seats where physical proximity is not suspicious; (2) the adversary gains software access to the microphones of the victim's phone, perhaps through a malicious app, and waits until the victim places the phone next to a keyboard. We note that it is not uncommon for users to place their phone on the desk while working on a laptop keyboard. Moreover, tablets and large phones are frequently used with external Bluetooth keyboards, where the device is placed directly behind the keyboard. We believe that this latter scenario is particularly likely and use it as the primary example in this paper.

We do not assume any particular structure in the typed information. This means that adversary seeks to identify not only text input matching a known linguistic model but also seeks to identify random input strings such as strong passwords. We also explicitly do not assume that the adversary has access to labeled training data (i.e., audio recordings for each key, where it is known which key was pressed). Such training data is specific to a particular phone-keyboard combination, the exact placement, and the exact acoustic environment. It would therefore be challenging to obtain in many adversarial settings.

2.2 Basic Concepts of Single Phone TDoA Localization

The selection of the attack model is rooted in an understanding of the fundamental limits of acoustic localization. To avoid the need for labeled training data we disregard fingerprinting techniques and focus on time-of-flight measurements, which are attractive given the relatively low propagation speeds of audio signals. Since we do not know when a particular keystroke sound is emitted, we rely on measuring the time-difference-of-arrival (TDoA) of this sound across the two microphones of the device.

A TDoA measurement reveals information about the direction of the incoming sound. Determining an exact position of origin for the sound normally requires triangulation, that is at least two direction measurements from different locations. In the keyboard snooping scenario, however, there is a discrete and relatively small set of candidate positions from which the sound can emanate: the center of each key. If the relative phone position and keyboard geometry is known, it is therefore possible to locate the sound even with a single TDoA measurement by finding the best match between the direction estimate and the expected direction for each key.

This process, however, requires mm-level accuracy, which is an order of magnitude beyond the cm-level accuracies that have been previously demonstrated in audio localization. Operating at this level of accuracy involves estimating precise hyberbolas instead of coarse direction estimates. Consider our primary scenario as illustrated in Figure 2(a). Let's denote the distance between two microphones as d, and the distance between the sound source (i.e., the keystroke made on the keyboard) to that of two microphones as $r1$ and $r2$, respectively. Suppose Δt is the TDoA measured at two microphones, the derived distance difference Δr from the pressed key to two microphones can be represented as:

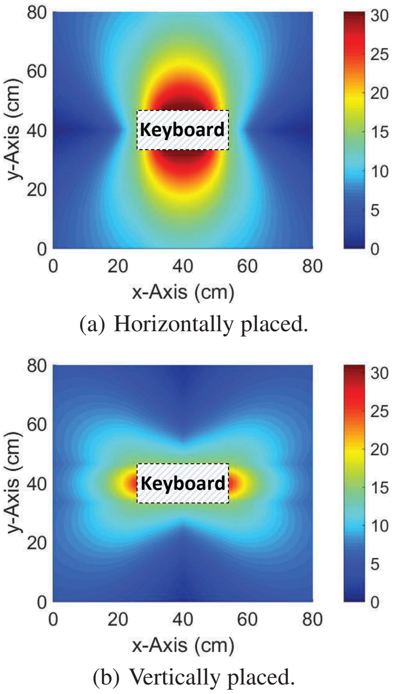
$$\Delta r = r1 - r2 = \Delta t \cdot s_0, \qquad (1)$$

where s_0 is the velocity of sound. All possible locations that satisfy Δr lie on a hyperbola as illustrated by the red curve in Figure 2(a). This hyperbola typically crosses several keys on the keyboard as indicated by the darker keys in the figure (and one key may also be crossed by more than one hyperbola). Narrowing this to a single key therefore involves determining the closest key center to this hyperbola. As can be seen in the figure, shifting the hyperbola by only a few mm would bring it closer to the center of a different key. For this reason, the process require mm-level precision and accuracy.

2.3 Factors Affecting Accuracy

The achievable accuracy and precision with TDoA measurements depend on several key factors.

Sampling Rate. When recording the keystroke sound, the sound is digitized by an Analog to Digital Converter (ADC) with a fixed sampling rate before it becomes accessible to applications. This therefore limits the resolution with which the time difference of arrival can be measured by application software. While signal



(a) Horizontally placed.

(b) Vertically placed.

Figure 3: Good phone locations for keystroke snooping. Warmer color indicates locations from which a higher range of TDOA values can be observed over different keys.

processing techniques exist that promise sub-sample accuracy, this time resolution serves as a useful guideline.

Current state-of-the-art audio hardware on mobile devices supports up to $192kHz$ sampling rate but drivers and or operating system usually still limit this to $48kHz$. At a speed of sound of 343m/s, this results in a resolution for the distance difference Δr of $\approx 1.8mm$ and $\approx 7.15mm$ for the two sampling rates, respectively.

Distance between Two Mics. The number of distinguishable hyberbolas also depends on the range of possible TDoA measurements. The range is bounded by the distance between two microphones on the phone. It can be calculated based on the triangle inequality theorem. As can be inferred from Figure 2(a), the range of Δr is $[-d, +d]$. The TDoA value Δt then falls into the range of $[-d/s_0, +d/s_0]$, which corresponds to the number of distinguishable hyberbolas N at the sample level as expressed below:

$$N = \lceil \frac{2d \cdot f_s}{s_0} \rceil. \tag{2}$$

For example, the distance between the two microphones of the Samsung Galaxy Note 3 smartphone is $d = 15.3cm$. With a sampling rate $f_s = 48kHz$, this yields 42 hyperbolas that can be discriminated at the sample level.

Placement of the Mobile Device. In practice, only a subset of these N hyperbolas may actually cross the keyboard and be useful for distinguishing keystrokes. The size of this subset depends on the size of the keyboard and the relative location of the recording device. Figure 3 shows the size of this subset depending on the phone position around the keyboard, for two different phone orientations. Warm colors indicate good phone positions relative to the keyboard. Horizontal phone orientation means that a line connecting the two microphones would be parallel to the long side of the keyboard (as also the case in Figure 2(a)). Vertical orientation means that the phone is rotated 90 degrees to the left, so that the line is parallel to the short side of the keyboard. This analysis assumes a sampling rate of $48kHz$, a keyboard size of $28cm \times 13cm$ (as for the Apple wireless keyboard MC184LL/A) and microphone distance of $d = 15.3cm$ (as on the Samsung Galaxy Note 3). The results show that a vertically placed phone on the side needs to be in very close proximity but a horizontally placed phone behind the keyboard offers a bit more flexibility. Such placements are consistent, however, with the attack scenarios that we have identified earlier. In our primary scenario (e.g., a Samsung Galaxy Note 3 is placed behind an apple keyboard as illustrated in Figure 2(a)), in particular, this leaves us with 31 hyperbolas crossing the 26 alphabetic keys.

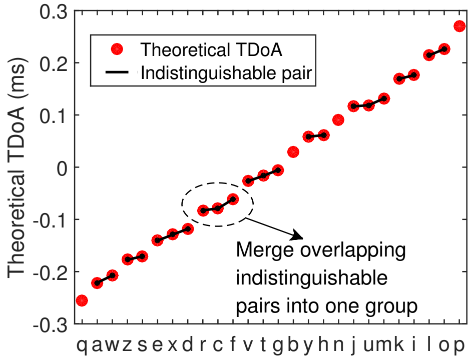


Figure 4: Illustration of sorted TDoAs for theoretical key groups construction.

2.4 Theoretical TDoA and Key Groups

Given known keyboard geometry and phone placement, keystroke snooping can be simplified as a matching process between measured TDoA values and expected TDoA values for each key. By solving for Δt in Equation (1), it is possible to compute an expected TDoA value for each key, which we also refer to as the theoretical TDoA value for a key.

In addition to the limiting measurement factors discussed earlier, measurements will be affected by noise. This further limits the distinguishability of keys and leads us to introduce the notion of theoretical key groups, which are groups of keys whose expected TDoA values are so close that we would expect them to be difficult to distinguish. For instance, the 26 alphabetic keys in our primary scenario are grouped into 13 theoretical key groups, each illustrated through a separate color in Figure 2(b).

These key groups are established as follows. We first sort the keys based on their theoretical TDoAs. We then link any pair of keys whose difference in theoretical TDoA is less than a threshold τ. Based on our experiments with different keyboards and sampling rates, we empirically determine τ as $\frac{1}{480}ms$ (which corresponds to $1, 2, 4$ TDoA samples corresponding to $48kHz$, $96kHz$ and $192kHz$). Each connected set of keys is then considered as one theoretical key group, as illustrated in Figure 4.

We will explain how to use these concepts and how to achieve accuracy below the level of a theoretical key group next.

3. SYSTEM OVERVIEW

To accurately recover keystrokes using a single mobile device, we design an approach that leverages TDoA measurements and

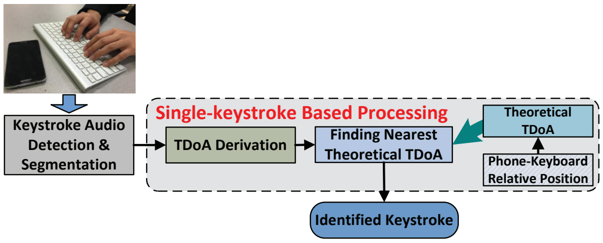

Keystroke Audio Detection & Segmentation
Single-keystroke Based Processing
TDoA Derivation
Finding Nearest Theoretical TDoA
Theoretical TDoA
Phone-Keyboard Relative Position
Identified Keystroke

Figure 5: System architecture: single-keystroke based processing.

fine-grained acoustic signatures of keystrokes. In this section, we discuss the challenges and architecture of our system design.

3.1 Challenges

To achieve the goal of accurately recognizing keystrokes by utilizing a single mobile device without relying on training and contextual information, the design and implementation of our system involve a number of challenges:

Sensing with Single Mobile Device. Using one single mobile device to recover keystrokes is challenging as most commercial mobile devices only support stereo recording with two microphones, while general acoustic TDoA localization approaches require at least three microphones to create multiple half-hyperbolas to locate a sound source. Moreover, the distance between two microphones in a phone is highly constrained, which limits the range of possible TDoA values. Although some mobile devices have three microphones, for example iPhone 5s and Samsung Galaxy Note 3, neither Apple nor Google provides API to record 3-channel audio with three microphones. Therefore, our system must be designed in a way that it can accurately identify keystrokes based on the stereo recording of two microphones.

Imperfect measurement of TDoA. Different from some recent TDoA localization studies [1–4] that utilize customized acoustic signals, such as a high frequency narrow band signal, our work locates more challenging sound sources, i.e., keystrokes, which cannot be controlled and contain rich frequency components. Meanwhile, the range of possible TDoA values is limited by the distance between two microphones and is affected by the placement of the mobile device. Also, the sampling frequency limits the resolution with which the time difference of arrival can be measured. Moreover, the measured TDoA may also be affected by multipath effects and environmental noises. These factors result in imperfect measurements of TDoA which make it hard to uniquely locate each keystroke.

Training-free Keystroke Recognition. Without the cooperation of the targeted user, developing training-free keystroke recognition is critical when performing keystroke snooping, especially when an adversary seeks to derive the user's sensitive typing information. Our system aims to recognize keystrokes without training efforts that involve target users (e.g., requiring the target user to type each key repeatedly to label the data beforehand).

Recovering Keystrokes without Linguistic Model. Users may type not only sentences following English language constraints (e.g., emails and articles), but also random letters or numbers (e.g., passwords and credit card numbers). Our developed method should have the ability to recover sensitive information consisting of random combination of letters and numbers. This requires our system to recognize keystrokes without relying on linguistic models or dictionaries.

3.2 System Architecture

The basic idea of our system is to perform keystroke snooping leveraging the dual-microphone on a single smartphone through studying the fine-grained acoustic signatures inherent from key typing sounds. In particular, we consider two processing approaches, namely *Single-keystroke Based Processing* and *Set-keystroke Based Processing*. These two approaches seek to cover various practical scenarios that have different requirements on the accuracy and response time. The *Single-keystroke Based Processing* can be applied to even a small set of recovered keystrokes, since it can process each keystroke individually. The *Set-keystroke Based Processing* exploits a larger set of keystroke samples to improve the recognition accuracy. It reduces the effect of imperfect measurement of TDoA by combining multiple keystroke samples from the same key. It can identify strokes of the same key by extracting the acoustic cepstral features of keystrokes as well as by using coarse TDoA matching.

In our proposed system, we assume the relative position of the mobile device to the keyboard is known. This information can be obtained if an adversary intentionally places the mobile device close to the keyboard, or when the external keyboard is attached to the mobile device (e.g., Microsoft Surface). There are also other means to obtain such information. For example, the adversary may take a picture of the setting of the keyboard and mobile device. The adversary may also estimate the setting using a bunch of collected keystrokes of multiple keys. It is important to note that such process does not need the participation of the target user as in the traditional training phase. Additionally, we discuss how to derive such information when the relative position the mobile device is unknown in Section 5.3.

Single-keystroke Based Processing. A quick way to recover each individual keystroke is to leverage the theoretical calculated TDoAs based on the relative position between the mobile device and keyboard. The Single-keystroke Based Processing method compares the measured TDoA derived from the input keystroke to the computed theoretical values and determine which key has been pressed. The main steps of this method are depicted in Figure 5 and described as follows: For each captured keystroke sound, this method first perform *Keystroke Audio Detection & Segmentation* to extract the audio signals corresponding to the press and release phases of the keystroke. It then derives the TDoA based on the extracted keystroke acoustic signal using signal processing techniques. Next, it determines which key has been pressed by finding the top-w keys that have the theoretical TDoAs closest to that of the input keystroke (i.e., *Finding the Nearest Theoretical TDoA*).

Set-keystroke Based Processing. This method aims to reduce the impact of the imperfect TDoA measurement by examining a set of input keystrokes and study the statistics of the fine-grained acoustic features in addition to pure TDoA computation. Figure 6 illustrates the steps of the Set-keystroke Processing approach. This method first takes as input a set of different keystroke sounds recorded by a nearby mobile device. It then extracts the audio signals corresponding to the keystrokes and derives the TDoAs. Next, it performs *Pre-grouping of Keystrokes Using TDoA* to categorize the input keystrokes to multiple key groups based on the pre-calculated theoretical key groups in Section 2. To overcome the limited accuracy, this method then extracts the cepstral features (e.g., Mel Frequency Cepstral Coefficients (MFCCs)) from the keystroke sounds through the *Keystroke Acoustic Features (MFCCs) Extraction* component. The MFCC features are utilized to further cluster the keystrokes in the same key group so that each cluster only contains strokes of the same key. This allows calculation

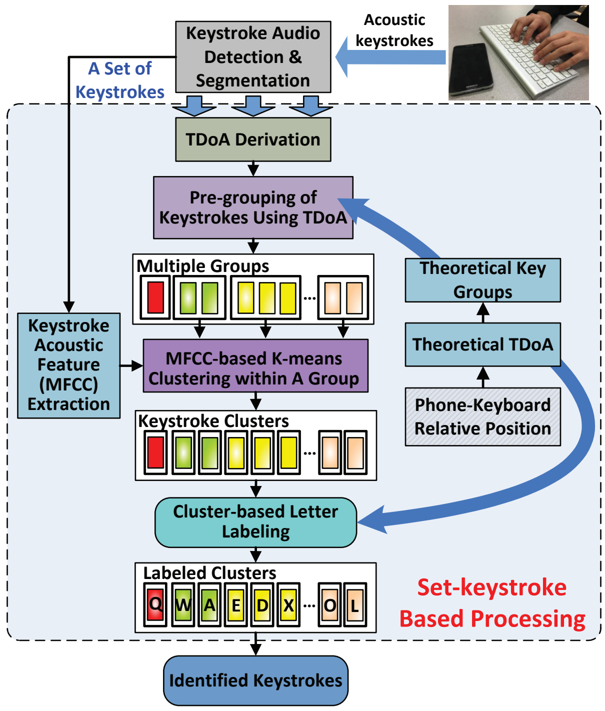

Keystroke Audio Detection & Segmentation
Acoustic keystrokes
A Set of Keystrokes
TDoA Derivation
Pre-grouping of Keystrokes Using TDoA
Multiple Groups
Theoretical Key Groups
Keystroke Acoustic Feature (MFCC) Extraction
MFCC-based K-means Clustering within A Group
Theoretical TDoA
Phone-Keyboard Relative Position
Keystroke Clusters
Cluster-based Letter Labeling
Labeled Clusters
Q W A E D X … O L
Set-keystroke Based Processing
Identified Keystrokes

Figure 6: System architecture: set-keystroke based processing.

of mean TDoA values over several strokes of the key, which helps reduce measurement noise. Finally, the system performs key labeling of each cluster to recover each keystroke by examining the distance difference between the mean TDoA of each cluster to that of the theoretical TDoAs. We discuss the details of this approach in Section 4.

4. SET-KEYSTROKE BASED PROCESSING

4.1 Pre-grouping Keystrokes Into Theoretical Key Groups

After a set of keystrokes are collected, the Set-keystroke Based Processing approach first obtains the TDoA of each keystroke based on the techniques described in Section 5. It then utilizes these derived TDoA values to assign each keystroke into a theoretical key group based on the discussion in Section 2. We denote each key as K_i^n, where i is the key ID and n is the corresponding theoretical key group ID (e.g., K_1^1 is the key "Q" and K_{19}^{12} is the key "L"). We further denote the theoretical TDoAs of keys with the theoretical key group ID n as $D_n = \{\Delta t_i^n\}$, where i is the key ID and Δt_i^n is the theoretical TDoA of the key K_i^n. We then put each input keystroke into one of the theoretical key group by comparing its measured TDoA Δt with the theoretical TDoAs Δt_i^n. The input keystroke will be assigned to the theoretical key group n, if the differential TDoA between Δt and Δt_i^n is the minimum as shown below:

$$G = \arg\min_{n} |\Delta t - \forall \Delta t_i^n \in D_n|. \qquad (3)$$

At the end, each input keystroke is assigned with a theoretical key group ID.

4.2 MFCC Based K-means Clustering

We next explore the acoustic features of keystroke sound to further separate the keystrokes within the same key group.

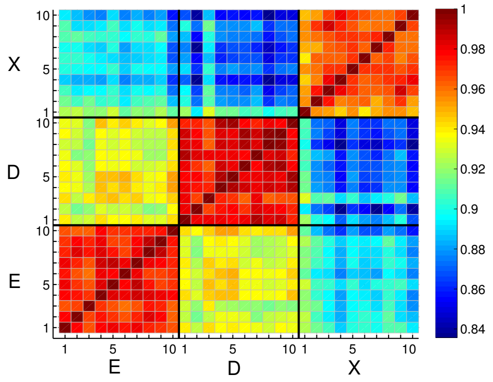


Figure 7: Pearson Correlation between MFCC features of three keys within a group: same key shows higher correlation, while different keys present lower correlation.

MFCC Feature Extraction. In our experiments, we find that the Mel-Frequency Cepstral Coefficients (MFCCs) [11, 12] of keystroke sounds capture acoustic signatures of different keys within the same theoretical key group. MFCC utilizes the magnitude of the Fourier Transform of the time-domain speech frames to analyze acoustic signals. The rationale of using MFCC to distinguish different keystrokes in the same theoretical key group is that physical uniqueness of each key component results in slightly different keystroke sounds for different keys. In addition, the keystroke sounds of keys at different locations experience different multipath effects. In particular, we extract the MFCC features from the entire duration of a keystroke sound. The number of filterbank channels is set to 32, and 16^{th}-order cepstral coefficients are computed in each $10ms$ Hanning window, shifting $2.5ms$ each time. To exclude the frequency range of ambient noise, we only consider acoustic signal from $400Hz$ to $14kHz$ for MFCC extraction.

To illustrate the effectiveness of using the MFCC features to distinguish different keystrokes within a key group, we repeatedly type "E","D", and "X" keys (which are within the same theoretical key group) 10 times respectively and examine the correlation between the MFCC features extracted from the keystrokes. Figure 7 shows the Pearson correlation coefficient [13] between any two MFCC features derived from those keystrokes. We observe that the MFCC features of the same key present higher correlation than that of different keys. It thus appears promising to use MFCC features to distinguish keystrokes within a group. We note only one channel of the keystroke sound is used to extract MFCC features. If dual-microphones have different characteristics, we could combine parallel features to improve the clustering performance [14].

In-group K-means based Clustering. To reduce the effect of the imperfect measurement of TDoA and minimize the impact of environmental noise, we further cluster keystrokes within a group into different clusters based on the MFCC features (if the corresponding theoretical key group contains multiple keys). In particular, we use the *cityblock* distance to measure the distance between MFCC features of different keystrokes using K-means clustering [15]. In order to obtain the optimal clustering results, we minimize the variances of the MFCC features of keystrokes in each cluster by satisfying:

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{n=1}^{N_k} |m_i^k - \mu_k|^2, \qquad (4)$$

where N_k represents the number of keystrokes in the k^{th} cluster, m_n^k denotes the MFCC features of the n^{th} keystroke in the k^{th} cluster, and μ_k is the mean value of the MFCC features in the k^{th} cluster.

4.3 Cluster Based Letter Labeling

Finally we label each cluster. We leverage the statistical information of TDoAs in each cluster to determine which key the cluster belongs to. In practice, the TDoA measured from multiple keystrokes for the same key may have slightly different values as the touch point may change slightly each time. In our experiments, we find that keystroke sounds emitted from different keys within group have different distributions of TDoAs which result in slightly different mean TDoA. Moreover, the mean TDoA of the keystroke sounds emitted by the same key is very close to the corresponding theoretical TDoA. Thus, we compare the mean values of the measured TDoAs of each cluster to the theoretical TDoAs. The keystrokes in the cluster will be labeled as the key whose theoretical TDoA has the minimum distance to the mean TDoA of that cluster.

5. IMPLEMENTATION

5.1 Keystroke Segmentation

A typical keystroke acoustic signal can be divided into three parts [5, 7]: *touch peak*, *hit peak* and *release peak*. These peaks correspond to touch, hit and release the key respectively. Figure 8 shows an example of these three peaks from two different keyboards (i.e., Apple wireless keyboard and Razer Black Widow keyboard).

In order to extract the acoustic sound of a keystroke, we first examine the energy levels of the acoustic signal to determine the starting point of the keystroke sound [6, 7, 9]. Particularly, we calculate the energy levels of a keystroke sound by accumulating the square of the signal amplitude in a sliding time window as shown below:

$$A(t) = \sum_{n=t}^{t+W} r^2(n), \tag{5}$$

where W is the length of the time window and $r(n)$ is the amplitude of the sound signal within the time window. We empirically determine the length of the sliding window as $W = 2ms$ (i.e., 96 samples with $48kHz$ sampling rate). Figure 8 illustrates the energy levels of the keystroke signals from two keyboards.

We identify the starting point of the keystroke sound when the energy level exceeds a threshold. An empirical threshold of 0.05 is used in our work to determine the starting point p_s. We find the length of keystrokes is typically about 100 milliseconds. We thus extract the keystroke sound as the acoustic signal between $[p_s - 5ms, p_s + 100ms]$. Note that our system uses the entire keystroke sound to generate MFCC features, whereas the system only uses about first $20ms$ segment roughly corresponding to *touch peak* and *hit peak* to calculate the TDoA as these two peaks result in more accurate TDoA estimation than using the whole keystroke sound.

5.2 TDoA Derivation

Once we have input keystroke segment, we could find out how many delayed samples between two digital audio signals recorded at two microphones at a mobile device to obtain the time delay between two microphones when receiving keystroke sound. Suppose the acoustic signal of a keystroke is recorded at the two microphones as $r_1(n)$ and $r_2(n)$ with length L respectively, where $n = 1, \ldots, L$. We use cross-correlation between the two recorded signals to derive the TDoA. Cross-correlation is a standard signal processing technique to measure the similarity between two signals and is calculated as:

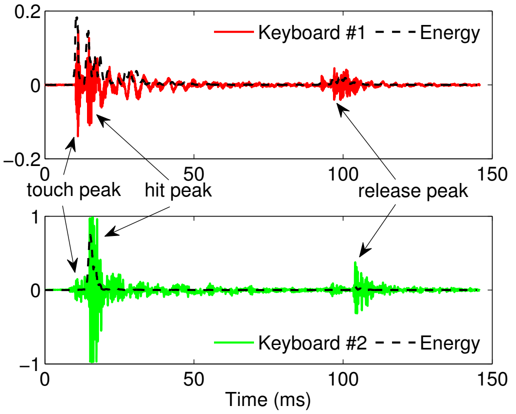


Figure 8: Keystroke acoustic signals emitted from two keyboards and corresponding short-time energy, keyboard-1 (Apple wireless keyboard) and keyboard-2 (Razer Blackwidow keyboard).

$$cc(d) = \frac{\sum_n \left[(r_1(n) - \mu_{r_1}) \cdot (r_2(n-d) - \mu_{r_2})\right]}{\sqrt{\sum_n (r_1(n) - \mu_{r_1})^2 \cdot \sum_n (r_2(n-d) - \mu_{r_2})^2}}, \tag{6}$$

where μ_{r_1} and μ_{r_2} are the means of the corresponding signals. $cc(d)$ provides the similarity between $r_1(n)$ and shifted (delayed) copies of $r_2(n-d)$. If the Equation 6 is computed for all delays $d = 0, 1, \ldots, L-1$ then it results in a cross correlation series of the original $r_1(n)$ or $r_2(n)$. Then, the TDoA Δt (i.e., time delay) between $r_1(n)$ and $r_2(n)$ can be obtained as:

$$\Delta t = \frac{1}{f_s} \cdot (\arg\max_d cc(d) - L). \tag{7}$$

5.3 Relative Position Estimation

The relative position of the phone to the keyboard is needed in our method to calculate the ground truth of TDoA (i.e., theoretical TDoA values). This information could be obtained if the adversary intentionally places the phone at a pre-identified location or if the phone/tablet is used in a tablet stand. If the adversary plants a malware into the victim's smartphone, such information could be inferred based on the keyboard layout and the measured TDoA of keystrokes.

Keyboard layout can be obtained offline as long as the keyboard model is known. The keyboard model could be detected by capturing Bluetooth identifiers or through manual visual identification. With the keyboard layout, we then can define the coordinates of keys. For the sake of simplicity, we assume there are m keys K_i with known coordinate or location loc_i, where $i = 1, 2, ..., m$. Given the measured TDoA of a collection of keystrokes, we have the TDoA value of each key to that of two microphones with certain measurement error. We assume the measured TODA of each key to that of two microphones is $\hat{\Delta t_i}$, and the sorted one is $\hat{\Delta t_j}$, with $j = 1, 2, ..., m$.

With the above information, we can estimate the locations of two microphones M_1 and M_2, with the constraint $\|M_1 - M_2\| = d$, where d is the known distance between two microphones. With the arbitrary locations of microphones and known location of the key loc_i, we can calculate the theoretical TDoA of each key to that of two microphones as Δt_i, with the sorted one Δt_j. The optimal location of the two microphones thus can be estimated as follow:

$$\arg\min_{M_1,M_2} \sum_{j=1}^{j=m} \|\hat{\Delta t}_j - \Delta t_j\|, \tag{8}$$

where $\|\hat{\Delta t}_j - \Delta t_j\|$ represents squared distance between $\hat{\Delta t}_j$ and Δt_j.

Note that we may not get the measured TDoA of each key in practice. Even so we can calculate the location of two microphones as long as we obtain several measured TDoA values. Moreover, the measured TDoA of different keystrokes for the same key may be different slightly. Empirical study shows that the difference is small (i.e., about one or two samples). We could then group similar TDoA values together and use the averaged value to represent the TDoA of one key.

6. SYSTEM EVALUATION

In this section, we first present the experimental methodology, and then evaluate the performance of both set-keystroke based and single-keystroke based approaches. We also discuss the impact of multipath propagation on the keystroke snooping.

6.1 Experimental Methodology

6.1.1 Keyboard & Phone

Keyboard. Although we do not study the sound intensity level of each key, we evaluate our system with three different kinds of keyboards (i.e., an Apple wireless keyboard MC184LL/A, a Microsoft surface keyboard and a mechanical keyboard Razer Black Widow Ultimate) that produce different keystroke sound intensity levels. In particular, the keystroke sound from the mechanical keyboard is much louder than that from the Apple keyboard. And the keystroke sound from the Microsoft surface keyboard is the weakest. These keyboards have different designs and dimensions resulting in different layout of keyboards and different characteristics of keystroke sounds. Specifically, the Apple wireless keyboard and Microsoft surface keyboard have comparable dimension (i.e., $\sim 28mm \times 13mm$), whereas Razer keyboard is much larger (i.e., $\sim 47mm \times 17mm$). Moreover, the depth of key caps on the Apple and Microsoft keyboards (i.e., $\sim 2mm$) is much smaller than that of the Razer keyboard (i.e., $\sim 6mm$).

Mobile Phone. In our experiments, we utilize the Samsung Galaxy Note 3 as the mobile device to launch attacks. The operating system of the phone is Android 4.4.2. Although the Samsung Galaxy Note 3 has equipped with three microphones on the top, bottom and right bottom, the microphone on the right bottom edge is only used for noise cancelation. We thus use the top and bottom microphones to record the keystroke sound. The distance between these two microphones is about $15.3cm$.

6.1.2 Sampling Rate

The audio chipset on smartphone (i.e., Samsung Galaxy Note 3) is Qualcomm Snapdragon 800 MSM8974 [16], which supports $24bit$ nominal quantization at $192kHz$ sampling rate. Although the Android 4.4.2 system only supports up to $48kHz$ sampling rate, Smartphone Operating Systems are increasingly supporting higher sampling rate, for example recently released Android 5.0 claims it could support up to $96kHz$ sampling rate [17]. We thus envision that the software restriction on the sampling rate will be loosed and the smartphone could use $192kHz$ for audio recording in a near future. In the evaluation, we study the impact of different sampling rates on the performance of keystroke snooping. We simulate the high sampling rate (e.g., $96kHz$ and $192kHz$) by utilizing a pair of omni-directional microphones connected to a laptop through a USB adapter (i.e., Diamond Tube). We place the two microphones $15.3cm$ apart from each other to simulate the Samsung Galaxy Note 3 with $96kHz$ and $192kHz$ sampling rates.

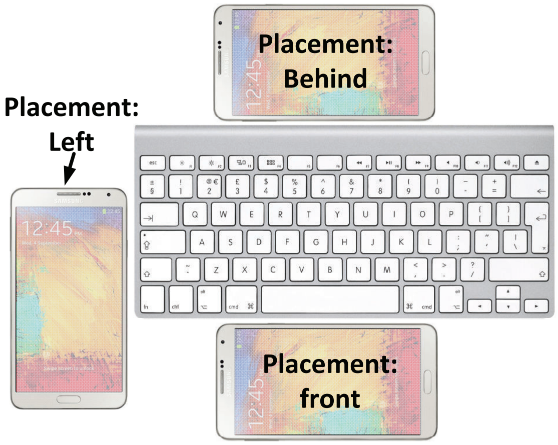


Figure 9: Three typical placements of the phone to the keyboard in the experiments.

6.1.3 Placement

We concentrate on the primary usage scenario, where the mobile device is placed behind the keyboard. We further study two more placement scenarios, where the mobile device is typically placed by the user when using the keyboards: in front of the keyboard and left side of the keyboard. These three placements are shown in Figure 9.

6.1.4 Data Collection

We focus on experiment on the 26 alphabet letters, but our method also applies to the whole keyboard. Three participants are involved to randomly type the 26 keys *a-z* on keyboards in typical office environments (i.e., two laboratory rooms with ambient noise (e.g., HVAC noise)). For each experimental setup (i.e., a specific type of keyboard, placement, and sampling rate), 520 keystrokes are collected. In total there are $3,640$ keystrokes from three participants for our experimental evaluation.

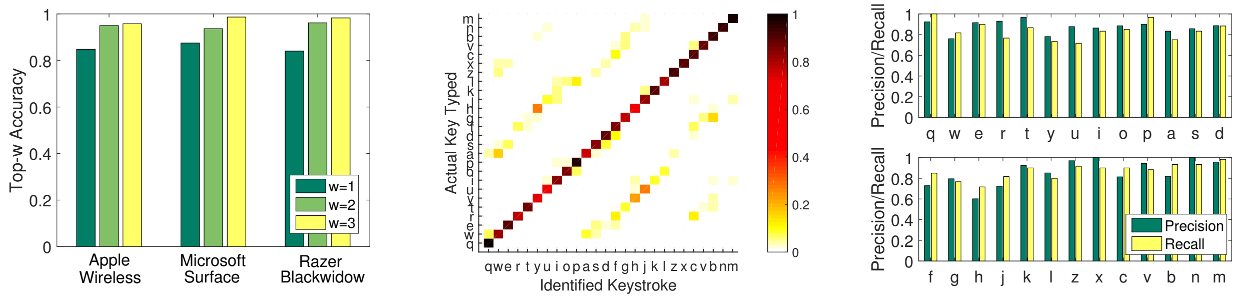
6.1.5 Metrics

We use the following three metrics to evaluate the performance of keystroke snooping:

Precision. Given N_k keystrokes of a key k, precision of recognizing the key k is defined as $P_k = \frac{N_k^T}{N_k^T + M_k^F}$, where N_k^T is number of keystrokes correctly recognized as the key k, M_k^F is the number of keystrokes corresponding to other keys mistakenly recognized as the key k.

Recall. Recall of the key k is defined as the percentage of the keystrokes that are correctly recognized as the key k among all keystrokes of the key k, which is $R_k = \frac{N_k^T}{N_k}$.

Top-w Accuracy. Given w identified key candidates, we want to know whether the pressed key is among these w candidates. The *top-w* accuracy measures overall performance of the keystroke recognition. Assuming the number of keys on keyboard is K, the *top-w* accuracy is defined as $A = \frac{\sum_{k=1}^{K} P_k^{T,w}}{\sum_{k=1}^{K} N_k}$, where P_k^T is the number of the keystrokes that are correctly identified as one of



(a) Top-w accuracy with different keyboards.

(b) Confusion matrix.

(c) Precision and recall.

Figure 10: Performance of set-keystroke based processing using three keyboards and off-the-shelf phone (48kHz).

the keys among the top-w candidates, N_k is the total number of keystrokes for key k.

6.2 Performance of Set-keystroke Based Processing

6.2.1 Overall Performance

We evaluate the overall performance of the set-keystroke based processing with the primary attack scenario (i.e, the phone is placed behind the keyboard). The sampling rate is set as $48kHz$. Figure 10(a) shows the overall accuracy for keystroke identification with three different keyboards. We find that the phone can capture different levels of keystroke sound intensity from all three keyboards when the mobile phone is placed close to the keyboard. We observe that all three keyboards have comparable high accuracies. In particular, the top-1 accuracy is about 85.5%, whereas the top-2 and top-3 accuracy increase to 94.9% and 97.6%, respectively. These results show that our training-free and context-free approach provides sufficient accuracy to snoop on passwords composed of random characters.

Figure 10(b) plots the confusion matrix for the keystroke recognition after combining the results from three keyboards. We find that there are only few keystrokes are mistakenly identified as incorrect keys. These mistakenly recognized keystrokes usually correspond to the neighboring keys that have the same TDoA value. For example, a few keystrokes of the key w are mistakenly recognized as the key a which is crossed by the same hyperbola as that of the key w, as shown in Figure 2(b). Moreover, two neighboring keys may produce similar keystroke sounds resulting in high similarity of MFCC features. This could also lead to a few keystrokes mistakenly recognized as different keys.

The precision and recall of recognizing each alphabetic key is shown in Figure 10(c). It combines the results for all three keyboards. Overall, the average precision is about 87% and the average recall is about 85%. This result shows that our system could recognize each individual alphabetic letter without linguistic model. Thus, our system could recover passwords consisting of random combination of letters.

6.2.2 TDoA Ranging

We next study how accurately we can measure TDoAs with the phone's microphone capability of $192kHz$ sampling rate. We compare the measured distance difference (i.e., measured TDoA multiplies velocity of sound) of the keystroke sound to the true distance difference of the key at the two phone microphones. We use Apple wireless keyboard and each alphabet key is typed ten times in the experiment. Figure 11 illustrates mean and standard deviation of error for each key. We observe that the average ranging error is about $2mm$ indicating that mm-level accuracy could be achieved at $192kHz$ which is the frequency supported by the smartphone audio hardware.

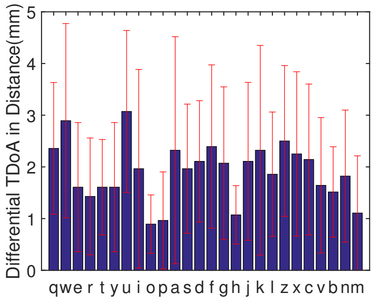


Figure 11: Differential between measured TDoA and theoretical TDoA with 192kHz sampling rate.

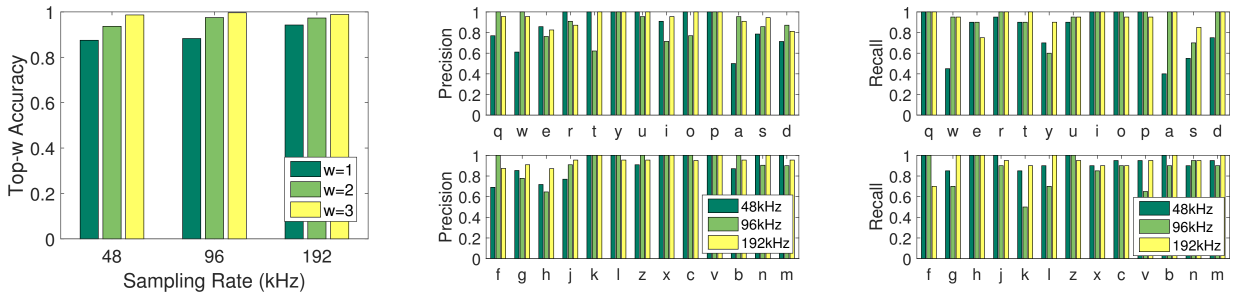
6.2.3 Effect of Sampling Rate

The impact of the sampling rate on the recognition accuracy is shown in Figure 12(a). The Microsoft surface keyboard is used in the experiment with the sampling rates of $48kHz$, $96kHz$ and $192kHz$. From Figure 12(a), we observe that higher sampling rate indeed improves the recognition accuracy as it provides higher TDoA resolution to discriminate the close by keys. In particular, the accuracy is improved from about 84.8% to 94.2% for top-1 candidate when increases the sampling rate from $48kHz$ to $192kHz$. However, the improvement on the top-3 candidates is marginal since these top-3 candidates usually covers these keys are spaced closely with the similar TDoA. The improved sampling frequency thus has limited improvement for the top-3 candidates.

Figure 12(b) and Figure 12(c) show the precision and recall for each key, respectively. We find higher sampling frequency in general improves the precision and recall, especially for the keys that hard to be distinguished at lower sampling frequency. For example, the keys w and a are physically close and the corresponding recalls and precisions are very low (at around 50%) when the sampling frequency is $48kHz$. They are improved to over 90% for both w and a at the sampling frequency of $192kHz$. This is also because higher sampling rate provides better TDoA resolution to distinguish close by keys.

6.2.4 Effect of Phone's Placement

Next, we study the performance under different phone placements. As shown in Figure 9, the phone is placed at three different



(a) Top-w accuracy with different sampling rates.

(b) Precision.

(c) Recall.

Figure 12: Performance of set-keystroke based processing using different sampling rates.

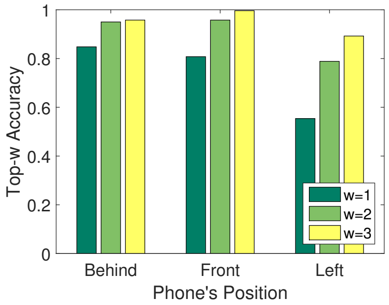


Figure 13: Top-w accuracy of set-keystroke processing with different placements of the phone to the keyboard.

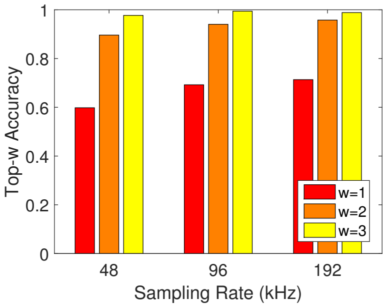


Figure 14: Top-w accuracy of single-keystroke processing with different sampling rates.

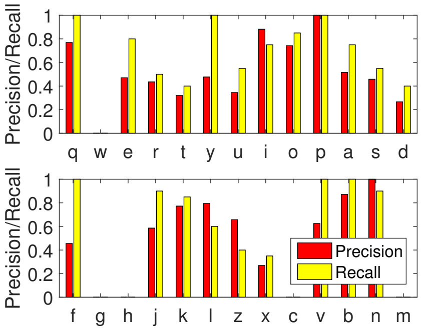


Figure 15: Precision and recall of single-keystroke processing with $48kHz$ sampling rate.

positions (i.e.,$behind$, $front$ and $left$) close to the Apple wireless keyboard. Figure 13 depicts the top-w accuracy for three phone placements. We observe that the placements of $front$ and $behind$ result in higher accuracy than that of $left$. This is inline with our analysis on phone placement shown in Figure 3(a). This also shows that the primary placement of phone-keyboard (i.e., $behind$) when the users use external keyboard is more vulnerable to keystroke snooping. In particular, top-1 2 and 3 accuracies are about 84.8%, 95%, and 95.7% for the primary placement respectively, whereas they are about 80.1%, 95.7%, and 99% for $front$ placement respectively.

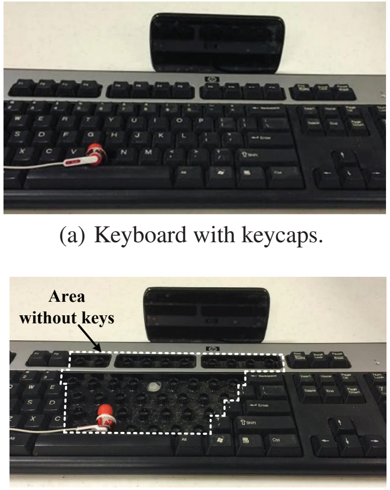
6.3 Performance of Single-keystroke Based Processing

We evaluate the naive approach, the single-keystroke based processing, by using the same dataset as we used for the set-keystroke based processing. Figure 14 shows the overall accuracy under different sampling rates. As expected, the naive approach has worse performance for top-1 accuracy when comparing to the set-keystroke based processing. This is because the single-keystroke based processing identifies keys based on a single TDoA value without exploiting acoustic features and statistic information of keystrokes of the same key. In particular, the top-1 accuracy of the single-keystroke based processing is about 60% at $48kHz$. The accuracy however increases dramatically to 89.6% for the top-2 accuracy and to 97.7% for the top-3 accuracy. This is due to that the top-2 and top-3 candidates usually cover the close-by keys that are hard to distinguish with one single TDoA.

In addition, the accuracy can be further improved by increasing the sampling rate to $96kHz$ or $192kHz$. With $192kHz$, the single-keystroke based processing can achieve 95% and 98% accuracy for top-2 and 3 candidates respectively. Figure 15 further shows the precision and recall of each key at $48kHz$ sampling rate. Since the single-keystroke based processing is hard to distinguish two keys with theoretical TDoAs within one sample, several keys are mistakenly recognized as others, such as keys *w,g,h,c* and *m* shown in Figure 15.

6.4 Multi-path Investigation

Multi-path Effects through Keys . Like many other wireless signals, multi-path effects may change the characteristics of acoustic signal. For keystrokes, because the sound sources are mostly inside each key and always below neighboring keys, it is important to understand the impact of multipath on the TDoA estimation in



(a) Keyboard with keycaps.

(b) Keyboard without keycaps.

Figure 16: Experimental Setups for multi-path investigation.

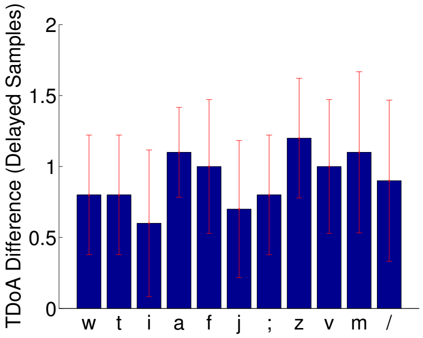
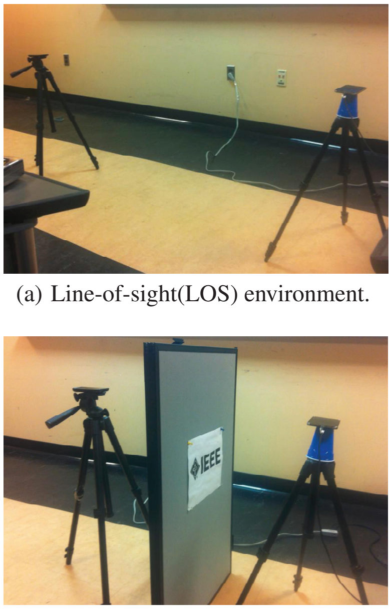


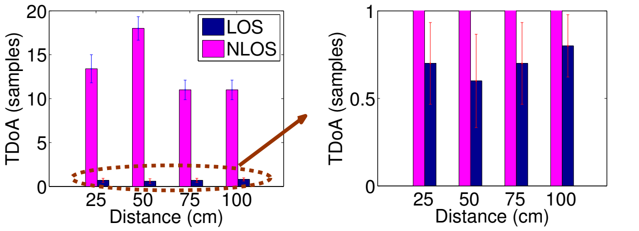
Figure 17: Differential TDoA between higher multi-path keyboard and lower multi-path keyboard (removed keys).



(a) Line-of-sight(LOS) environment.

(b) Non-line-of-sight(NLOS) environment.

Figure 18: Experimental Setups for multi-path investigation.



(a) LOS and NLOS TDoA estimation.

(b) Zoomed LOS and NLOS TDoA estimation.

Figure 19: TDoA estimation for LOS and NLOS environment.

our system. Particularly, we conduct following experiments: as shown in Figure 16(a), we first use the Samsung Galaxy Note 3 to play a pre-recorded chirp sound signal via a earbud, which is to make sure the sound comes below neighboring key caps and thus has multipath effects, for 20 times at each target key's position on a regular keyboard. Next, we repeat the same experiment, but remove the neighboring key caps as shown in Figure 16(b). Note that the phone is placed at 90 degree angle with the keyboard and the microphones are at a higher level than the keys on the keyboard to better study the multipath effects through keys. We remove all the keys between the earbud and the phone in order to simulate a lower multi-path environment. Figure 17 shows the difference of measured TDoA between such keyboards with different levels of multi-path effects. The average difference is only about 1 sample, and we thus conclude that the impact of multi-path (key caps on keyboard) does not have much influence on the TDoA estimation.

Non Line of Sight Effects. In the experiment, we use two mobile phones(i.e., Samsung Galaxy Note 3 and HTC Evo 4G) on two tripod at heights 1 meter above the ground as shown in Figure18(a). Similarly, we use the Samsung phone to record the chirp sound played by the HTC phone for 20 times. We align two phones to make sure the measured TDoA is close to 0 in the line-of-sight scenario. Next, we repeat the experiment, but with a thick card board separator placed in between the two phones to simulate the non-line-of-sight scenario as shown in Figure 18(b). Figure 19(a) shows the overall statistics of TDoAs in both the line-of-sight(LOS) scenario and non-line-of-sight(NLOS) scenario. Figure 19(b) is the enlarged part within the circle in Figure 19(a). Compared to the LOS scenario, the measured TDoAs increase dramatically in the NLOS scenario.

7. DISCUSSION

Environmental Accuracy. There are several factors that have an important effect on accuracy including phone placement, multipath, and noise. Our system has been evaluated with different phone placements close to the keyboard. Accuracy would significantly degrade for recordings taken at larger distances and meter-level distances would require much larger microphone separation, for example by using multiple cooperating devices. We believe, however, that close proximity is possible even in adversarial settings, for example if the adversary co-opts the users own phone or if the attack takes place in relatively confined space (e.g., airplane). As any time-of-arrival related localization technique, our system relies on a detectable signal arriving on the line-of-sight path. If this path is significantly attenuated by an obstacle, our system will measure a reflected signal which leads to errors too large to allow for recovery of keystrokes (as illustrated in section 6.4). Phone placement close to the keyboard makes such an obstacle unlikely, however. We evaluate our system in typical office environments (i.e., two laboratory rooms with ambient noises (e.g., HVAC noise)), and our results show little impact under such ambient noises. Although we observe that loud noises (e.g., people talking) could impact the detection accuracy, we believe that additional filtering or context-based word correction could further improve the accuracy.

Security Concerns. To our knowledge, this is the first demonstration of acoustic keystroke recovery that raises more serious concerns regarding password snooping. It appears practical that malicious background apps with microphone access could recover passwords entered from a nearby keyboard (either an associated Bluetooth keyboard or a keyboard used for another device). If high definition stereo audio trickles down from professional video conference systems, to voice over ip and video calling apps, keys typed during a call could potentially be recovered by the remote party. It may also be possible for an adversary to inconspicuously place a phone near a victim's keyboard, particularly in tight settings such as an airplane. That said, the attack is currently only possible with select phone models that expose stereo recording and have large microphone separation and even at future expected sampling rates of 192kHz there is only a moderate chance of accurately capturing a long random password on first attempt. Still, this significantly reduces the password entropy to a small set of candidates that can be brute-forced and the accuracies would be sufficiently higher for the many weaker passwords in use, when combining the keystroke recognition results with knowledge about common password patterns.

While there is already considerable awareness of privacy risks associated with microphones, this awareness usually extends only to spoken words and not necessarily to keystrokes. Users might therefore type sensitive information even if they know that recording devices are present. Overall, these results indicate that microphone access on mobile device should be tightly controlled and we hope to raise awareness to that the recoverable information from mobile device audio recordings extends far beyond spoken conversations.

Localization Implications. More generally, the results show that low-multipath scenarios exist where mobile audio enable mm-level ranging and localization. Such high accuracies could be exploited for numerous applications from motion tracking [3], over driver phone use detection [4], to user interface improvements [18]. Currently, app-level access to these audio capabilities is still very limited; the capabilities are primarily used for specific functions such as noise cancellation during calls or high definition audio playback. In light of these localization results, we argue that app-level software access to multiple microphones and high sampling rates for localization purposes should become a higher priority.

8. RELATED WORK

There have been active research efforts in keystroke recognition based on the acoustic emanation or vibration of the keystroke [5–10, 19–21]. Acoustic emanation based approaches [5, 7–9] mainly rely on the observation that each key produces unique acoustic signal when typed, whereas the vibration based methods [10, 19–21] capture the correlation between the vibration of the keystroke and the location of the keystroke occurred. Vibration based methods all require training efforts to label the keystrokes and usually have less recognition accuracy than that of acoustic emanation based approaches.

In particular, Asonov *et al.* [5] observe that the sound of keystrokes differs slightly from key to key and build a supervised learning based approach to recognize keystrokes. This problem is then revisited by Zhuang *et al.* [7] through adding the language modeling to boost the English text recognition. Berger *et al.* [8] propose a dictionary-based approach leveraging the observation the keystroke sounds correlate to their physical positions on the keyboard. UbiK [6] proposes to locate the location of keystrokes made on solid surfaces leveraging multi-path fading with moderate training efforts. More recently, Zhu *et al.* [9] proposes to utilize microphones on three phones to identify the keystroke of nearby keyboard. The requirements of three phones and the achieved accuracy (i.e., 72.2%) make their approach less feasible for real attack scenarios. Comparing to the above research efforts, our approach is able to achieve high keystroke recognition accuracy by using a single phone without any training.

Another body of related work is smartphone based localization or ranging using acoustic signals [1–4, 22–29]. Beepbeep [1] and SwordFight [3] propose phone-to-phone ranging systems that can achieve centimeter level accuracy. Qiu *et al.* [2] develops a 3D continuous localization system for phone-to-phone scenarios with about 10 cm accuracy. The above work however requires application-level communication between two involved phones. Yang *et al.* [4] introduces an acoustic relative-ranging system that classifies phone's position inside the car. This approach relies on customized beep sound for acoustic signal detection. Tarzia *et al.* [22] introduces a technique based on ambient sound fingerprint achieve room-level accuracy. Constandache *et al.* [23] deploys extra acoustic infrastructure inside the building for correcting users' movement traces captured by the accelerometer and compass. In our work, we exploit dual microphones on smartphone to locate the keystroke with high accuracy without customized beep sound or phone-to-phone communication.

9. CONCLUSION

In this paper, we show that microphones on a single off-the-shelf phone can be used to discriminate mm-level position differences, which not only creates potential security and privacy concerns related to recovering keystrokes being typed on a nearby keyboard, but could also benefits a broad range of applications relying on fine-grained localization (e.g., sensing touch interaction on surfaces around mobile devices and tracking speakers in multiparty conversations in a meeting room). The implemented system does not require any training or linguistic model, which makes it applicable in real-world adversarial context and has the capability to recover random typing (e.g., passwords). In particular, our system exploits digital acoustic signals received at the microphones from an off-the-shelf phone and leverages the integration of geometry-based TDoA and fine-grained acoustic signatures to exceed the resolution limit of TDoA and accurately identify keystrokes. Extensive experiments involving three types of keyboards demonstrate that, with $48kHz$ sampling rate, our proposed system can accurately identify a set of keystrokes with over 85% accuracy. The accuracy of our system can achieve as high as 94% with the higher sampling rate (i.e., $192kHz$). Additionally, our system can snoop even a single keystroke input at the accuracy of 97% among the top-3 candidate keys with $48kHz$ sampling rate.

10. ACKNOWLEDGEMENT

We thank our anonymous shepherd and reviewers for their insightful comments. This work was partially supported by the National Science Foundation Grants CNS-0954020, CNS-1409767, SES-1450091, CNS-1505175, CNS-1223977, CNS-1409811 and Army Research Office W911NF-13-1-0288.

11. REFERENCES

[1] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, “Beepbeep: A high accuracy acoustic ranging system using cots mobile devices,” in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, 2007.

[2] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, “On the feasibility of real-time phone-to-phone 3d localization,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (ACM SenSys)*, 2011.

[3] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, “Swordfight: enabling a new class of phone-to-phone action games on commodity phones,” in *Proceedings of the 10th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, pp. 1–14, 2012.

[4] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, “Detecting driver phone use leveraging car speakers,” in *Proceedings of the ACM International Conference on Mobile Computing and Networking (ACM MobiCom)*, 2011.

[5] D. Asonov and R. Agrawal, “Keyboard acoustic emanations,” in *2012 IEEE Symposium on Security and Privacy*, 2004.

[6] J. Wang, K. Zhao, X. Zhang, and C. Peng, “Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization,” in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, pp. 14–27, 2014.

[7] L. Zhuang, F. Zhou, and J. Tygar, “Keyboard acoustic emanations revisited,” in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 373–382, 2005.

[8] Y. Berger, A. Wool, and A. Yeredor, “Dictionary attacks using keyboard acoustic emanations,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 245–254, 2006.

[9] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, “Context-free attacks using keyboard acoustic emanations,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 453–464, 2014.

[10] P. Marquardt, A. Verma, H. Carter, and P. Traynor, “(sp) iphone: decoding vibrations from nearby keyboards using mobile phone accelerometers,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 551–562, 2011.

[11] K. S. R. Murty and B. Yegnanarayana, “Combining evidence from residual phase and mfcc features for speaker recognition,” *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 52–55, 2006.

[12] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to supervectors,” *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[13] “Pearson product-moment correlation coefficient.” http://en.wikipedia.org/wiki/Pearson-product-moment-correlation-coefficient,2015.

[14] Y. Obuchi, “Mixture weight optimization for dual-microphone mfcc combination,” in *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 325–330, IEEE, 2005.

[15] D. J. MacKay, *"Information theory, inference, and learning algorithms"*. Cambridge University Press, 2003.

[16] “Snapdragon 800 processors.” https://www.qualcomm .com/products/snapdragon/processors/800, 2015.

[17] A. KJ, “Android l update to bring sound quality that will please audiophiles.” http://www.ibtimes.co.uk /android-l-update-bring-sound-quality-that-will-please-audiophiles-1454695,2014.

[18] J. Schwarz, D. Klionsky, C. Harrison, P. Dietz, and A. Wilson, “Phone as a pixel: enabling ad-hoc, large-scale displays using mobile devices,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2235–2238, 2012.

[19] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, “Tapprints: your finger taps have fingerprints,” in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, pp. 323–336, 2012.

[20] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, “Accessory: password inference using accelerometers on smartphones,” in *Proceedings of the 12th Workshop on Mobile Computing Systems & Applications*, p. 9, 2012.

[21] Z. Xu, K. Bai, and S. Zhu, “Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors,” in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 113–124, 2012.

[22] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, “Indoor localization without infrastructure using the acoustic background spectrum,” in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, 2011.

[23] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, “Did you see bob?: human localization using mobile phones,” in *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, 2010.

[24] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, “Push the limit of wifi based localization for smartphones,” in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, pp. 305–316, 2012.

[25] C. Jiang, M. Fahad, Y. Guo, J. Yang, and Y. Chen, “Robot-assisted human indoor localization using the kinect sensor and smartphones,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[26] K. Liu, X. Liu, and X. Li, “Guoguo: Enabling fine-grained indoor localization via smartphone,” in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, pp. 235–248, 2013.

[27] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, and Y. Liu, “Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones,” in *Proceedings IEEE INFOCOM*, pp. 370–378, 2014.

[28] B. Xu, G. Sun, R. Yu, and Z. Yang, “High-accuracy tdoa-based localization without time synchronization,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 8, pp. 1567–1576, 2013.

[29] M. Uddin and T. Nadeem, “Rf-beep: A light ranging scheme for smart devices,” in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 114–122, 2013.