Auditory Eyesight: Demystifying μs-Precision Keystroke Tracking Attacks on Unconstrained Keyboard Inputs

Abstract

In various scenarios from system login to writing emails, documents, and forms, keyboard inputs carry alluring data such as usernames, passwords, addresses, and IDs. Due to commonly existing non-alphabetic inputs, punctuation, and typos, users' natural inputs rarely contain only constrained, purely alphabetic keys/words. This work studies how to reveal unconstrained keyboard inputs using auditory interfaces.

Audio interfaces are not intended to have the capability of light sensors such as cameras to identify compactly located keys. Our analysis shows that effectively distinguishing the keys can require a fine localization precision level of keystroke sounds close to the range of microseconds. This work (1) explores the limits of audio interfaces to distinguish keystrokes, (2) proposes a μs-level customized signal processing and analysis-based keystroke tracking approach that takes into account the mechanical physics and imperfect measuring of keystroke sounds, (3) develops the first acoustic side-channel attack study on unconstrained keyboard inputs that are not purely alphabetic keys/words and do not necessarily follow known sequences in a given dictionary or training dataset, and (4) reveals the threats of non-line-of-sight keystroke sound tracking. Our results indicate that, without relying on vision sensors, attacks using limited-resolution audio interfaces can reveal unconstrained inputs from the keyboard with a fairly sharp and bendable "auditory eyesight."

1 Introduction

Devices with auditory interfaces are ubiquitous in various personal, business, and public scenarios. For instance, smart devices with voice interfaces are used in home, office, or even hotel and traveling environments [1–3, 9, 12]. Always-on microphones widely exist in smart speakers, smartphones, smart TVs, and remote controllers [11, 31]. Security-conscious individuals might be aware of information leakage while typing in the field of view of surveillance cameras. However, users may not fully expect the risks of recovering data from their natural, unconstrained keyboard inputs with audio interfaces.

This work studies how to reveal arbitrary, unconstrained keyboard inputs using auditory interfaces. By design, auditory interfaces are not meant to have the "eyesight" to distinguish keystrokes on compactly spaced keys from a distance. Different from light sensors such as cameras that capture lights in the field of view, microphones receive sound waves from the entire environment with diffraction and reverberation.

First, we observe the key factors in localizing keystroke sounds. Contrary to existing perceptions that recording sample rates are the main factors to determine the precision [39] and inevitable errors [46], we find that the commonly used sample rates for general-purpose voice applications do not necessarily limit keystroke sound localization. The precision limit is related to how we process the signals by taking into account the keystroke sound physics and imperfect measuring (Section 2.1). For example, it is necessary to know that keystroke sounds are not only generated from the center points of the keys. Strong sound components can be generated by the vibrated keyboard base when the key hits it. When trying to localize the keystrokes, there will be an interfering self-masking effect due to sound components originated from the entire vibrated area and the keyboard. The diffraction and reverberation of sounds further strengthen this masking effect.

We develop a signal processing and analysis-based approach to understand the internal keystroke sound components. Inspired by observations on keystroke physics and measuring, we analyze the physical properties, measuring and processing of keystroke sounds that can affect the localization precision. To deal with the complex keystroke sound components, we design a multi-round structure with customized processing chains to address different scales of localization errors and to distinguish the keys.

Users' natural inputs, such as credentials, numbers, dates, emails, and addresses, are not purely alphabetic. Even when users input real-world texts, there are still punctuation, number, backspace, and capital letters. It is challenging to reveal users' natural, unconstrained keyboard inputs from a side channel due to following unaddressed problems.

First, the attack will deal with drastically expanded solution

space. Without excluding non-alphabetic keys nor assuming that the inputs follow known sequences in a dictionary or training dataset, the solution space will expand from alphabetic keys or known sequences to arbitrary inputs. Second, the attack needs to distinguish a large number of compactly spaced keys from a distance. By including number, symbol, and editing keys, the main block of a keyboard contains about 50 commonly used keys in a 27.2×7.1 cm area (Fig. 5). Third, there are interfering sound components generated along with the keystrokes (e.g., sounds from the entire vibrated keyboard area and reverberation). In both time and frequency domains, the sound components of a keystroke are heavily blended, and the interference cannot be simply eliminated. Finally, there is a lack of an existing reference study on side-channel keystroke attacks with users' natural unconstrained inputs.

Our work solves the above problems by exploring side-channel signal processing and analysis of keystroke sounds, and developing an attack study on unconstrained user inputs. We explore the keystroke localization precision limits and show how privacy and security-sensitive information can be recovered by the attacks without vision sensors. We include different contents, different users with their natural typing speeds/styles in our study to evaluate the threats.

In summary, our paper makes the following contributions:

- We develop the first keystroke side-channel attack study on natural, unconstrained inputs, including non-alphabetic keys and sequences that are not necessarily in a known dictionary or training dataset, such as real-world text inputs (with typos, punctuation, and capital letters) and strong passwords.
- We identify key properties related to keystroke localization precision that are necessary to analyze keystroke sounds. Inspired by the physics and measuring of keystroke sounds, we propose a novel multi-round keystroke localization approach. The approach is based on customized signal processing methods to distinguish keystroke sound signals in the differential range of microseconds.
- We further reveal the threats of bendable auditory eyesight. Our approach can reveal the users' inputs in scenarios when the line-of-sight is blocked and a camera cannot perceive the typed keys. We show that the attack can be launched with an adversary's laptop placed in non-line-of-sight (NLOS) settings, and our approach can mitigate excessive errors under non-line-of-sight transmission of keystroke sounds to distinguish keys without inducing significant overlapping.

2 Keystroke Sound Anatomy

We observe that keystroke sound signals contain different, complex components. These components are generated and mixed in a short time during the keystrokes. Existing works focused on using the entire keystroke sounds to directly extract information [21, 23, 39, 46, 48]. However, when all components of the signals are handled equally in a coarse-grained, large-scale manner, the detailed information can be lost due to various interfering and imperfect factors.

In this section, we will analyze and understand the subtle differences among the internal components of keystroke sound signals. We will reveal a keystroke sound anatomy model for perceiving and analyzing the detailed information within keystroke signals.

2.1 Imperfect and complex keystroke sounds

Keystroke sound signals are complex real-world signals recorded by microphones. Different from a known signal created by an electronic sound emitting device (e.g., a speaker), keystroke sounds are unknown signals affected by various physical properties such as mechanical properties, friction, users' statuses, typing styles, speeds, forces, transmission, and environmental reverberation.

We find following issues in acoustic side-channel analysis of keystrokes: (1) **Complex underlying physics**. The keystroke sound signal is not generated from a single point such as the center of the keycap. Multiple vibrated components (e.g., the key and the keyboard base) can generate sounds at the same time in the process of a keystroke. As a result, keystroke sounds are combinations of different kinds of sounds caused by vibration and friction when applying forces to the mechanical components of the keyboard. (2) **Imperfect measuring**. The recorded signals will not be the same as the generated sounds because of reverberation and diffraction. For instance, signals caused by reverberation do not carry correct localization information since they do not directly come from the typed key; such signals will inevitably interfere with the analysis. Additionally, there can be distortion and loss of resolution while measuring and digitizing keystroke sounds using microphones. (3) **Indistinguishable components in coarse-grained analysis**. The different components and interfering effects are naturally blended in a very short time and cannot be distinctively observed in coarse-grained analysis that uniformly processes the entire signals.

2.2 Perceiving the (im)precision

Assuming the keystroke signals recorded by two microphones are $y_1[i]$, $y_2[i]$, $(i \in \{0,1,2,3,...\})$ and the delay measurement is ΔN samples. Ideally, when we shift one of the signals by the delay $(y'_1[i] = y_1[i+\Delta N], y'_2[i] = y_2[i])$, we should observe the signals $y'_1[i]$ and $y'_2[i]$ perfectly matched and aligned to each other. However, in practice, we observe that there can often be misalignment. Such imprecision can be in the μs range and is not obvious with a large, coarse-grained scale. In order to perceive the imprecision, we need to interpolate

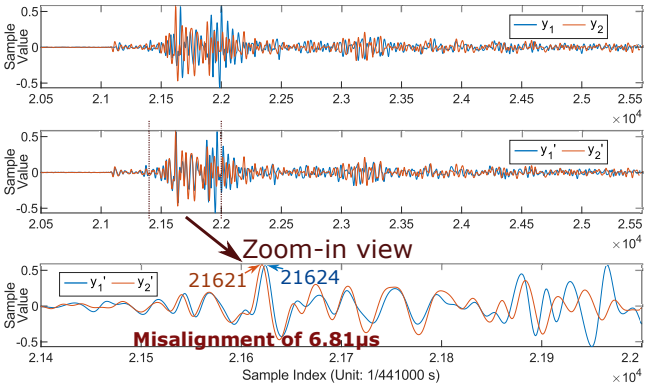

Figure 1: Top: Keystroke signals recorded by a microphone pair. Middle: Keystroke signals aligned based on cross-correlation results. Bottom: Zoom-in view. We can observe the misalignment from a microscopic view. Such errors are usually in the range of several to tens of microseconds.

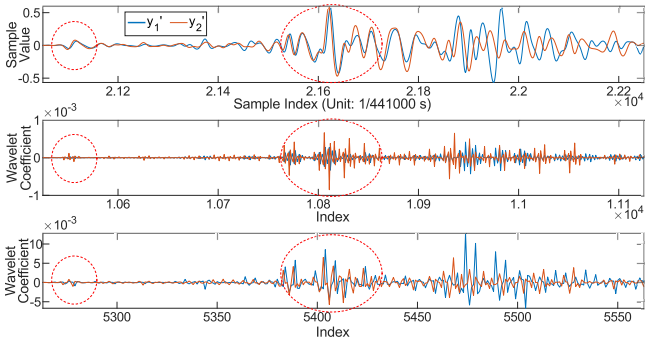

Figure 2: Top: Keystroke signals aligned based on cross-correlation results. Middle: Level-1 wavelet coefficients. Bottom: Level-2 wavelet coefficients. We can perceive the misalignments in the signals especially their transient parts (in red circles) more clearly from the wavelet decomposed signals. Even in the same level of coefficient, the other parts of the recorded signals (y_1 and y_2) differ significantly. These components will result in the miscalculation of the time delay.

the recorded signals to scale the sample interval close to the μs range or smaller. We then compute the time-difference-of-arrival (TDoA) [28, 37] using cross-correlation and align the keystroke signals based on the calculated delay. As shown in Fig. 1, we record the keystrokes of an Apple magic keyboard using a standard audio sample rate (44.1 kHz) from a 0.5-m distance. We interpolate the signals with a resample rate of 441 kHz to adjust the sample interval of the digitized signals to 2.27 μs. The zoom-in view shows the misalignment between the signals.

Causality analysis. By zooming in and observing the signals, we notice different parts in keystroke signals. We observe that the parts at the beginning of the signals can be aligned more closely and consistently than the following parts. As shown in Fig. 1, the front parts of the signals y_1' and y_2' are quite similar. However, after the beginning parts, the alignment between y_1' and y_2' becomes more irregular. These irregular parts tend to have more noises, including artifacts caused by reverberation as well as sounds generated from other components such as the keyboard base. Signals in these irregular parts could provide coarse-grained information but will mask high-precision localization data.

Insight 1: There are different parts in recorded keystroke signals. The signals in the beginning parts are relatively consistent; the following parts are more susceptible to irregularities caused by the interfering effects related to the physics, transmission, and imperfections of keystroke sound signals.

Keystroke signal decomposition analysis. We then decompose the signals to gain deeper understanding of the internal components. We use wavelet [44] to decompose the signals. Compared to FFT [21] or short-time Fourier transform (STFT), wavelet is particularly suitable to process short, natural signals like keystrokes while providing satisfying temporal resolution. Further, the wavelet coefficients can easily capture the transient parts of the keystroke signals.

In the decomposed signals with Symlets 6 (sym6) [22] wavelets, we can also observe the different parts. Specifically, we identify the transient parts and noisy parts. For instance, the transient parts in the beginning of the signals show more consistent alignment in the wavelet coefficients (Fig. 2). However, the following noisy parts can differ a lot for the same level of coefficients. This means that even in the specific bandwidth of the decomposed signals, the noisy parts still cause significant irregularity between the recorded signals (y_1 and y_2). Compared to the transient parts, these noisy parts have more distributed energy and noises.

Insight 2: We identify the transient parts and noisy parts in keystroke sounds by decomposing the signals. When a keystroke is recorded by two microphones, the alignments of signals in the transient parts are relatively more regular compared to the noisy parts.

The decomposed wavelet coefficients also show the misalignment more clearly in the transient parts at the beginning of the signals. The first transient part is related to the event of the finger tip landing on the keycap; the second transient part is related to the finger pushing down the key and activating the internal switch. These parts usually contain more intense, relatively high-frequency components than the other parts of keystroke sounds.

In this subsection, we have observed different parts in keystroke signals, including the transient parts and the noisy parts. Signals in the transient parts can be utilized to perceive the precision by observing their alignments, while signals in the noisy parts can lead to miscalculation of the time delay between the recorded keystroke signals (y_1 and y_2).

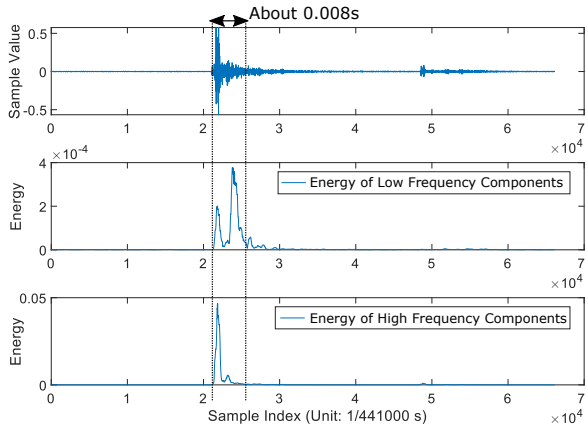

Figure 3: We separate the signal components with low/high pass filters (4 kHz) and use convolution with Hanning windows to observe their differences in the relative temporal energy distribution. The energy of both high/low-frequency parts is concentrated in a very short range (e.g., 0.008s), which is less than $1/10$ of the length of the entire keystroke sound signals (e.g., 0.1s). The high-frequency energy concentrates in less than 0.005s when the sound initiates.

Insight 3: The (im)precision of the calculated time delay can be perceived by scaling the unit close to the μs range, aligning the keystroke signals based on the time delay, and observing the transient parts (Fig. 2) in the beginning of the signals.

2.3 Internal energy distribution

We observe that: 1) while an entire keystroke signal lasts for 0.1s [21, 46, 48], most of the energy in both the high and low frequencies is concentrated in a very small time range (e.g., 0.008s), which can be less than $1/10$ of the entire keystroke signal. 2) the energy distribution is related to the physics of keystroke sounds. The high-frequency (*e.g.*, >4kHz) energy is stronger and more focused because it contains more high-amplitude, short-duration transient energy at the beginning of the keystroke sound (Fig. 3). The low-frequency (*e.g.*, <4kHz) energy is weaker and distributed more widely in the time domain because it contains more harmonics generated by the vibrated mechanical components after the transient parts. These longer low-frequency parts can also be contaminated by other interfering effects, such as echoes that usually comes later after the keystroke sound initiates.

Insight 4: Both low and high-frequency energy is concentrated within a very short time range (e.g., 0.008s). The high-frequency energy can be focused in an even smaller range (e.g., 0.003s). Thus, directly using and uniformly processing all parts of the entire keystroke signal may not be the optimal approach; the long low-signal-to-noise-ratio (SNR) parts (about 9/10 of the entire keystroke signal) can obfuscate the details and degrade the analysis effectiveness.

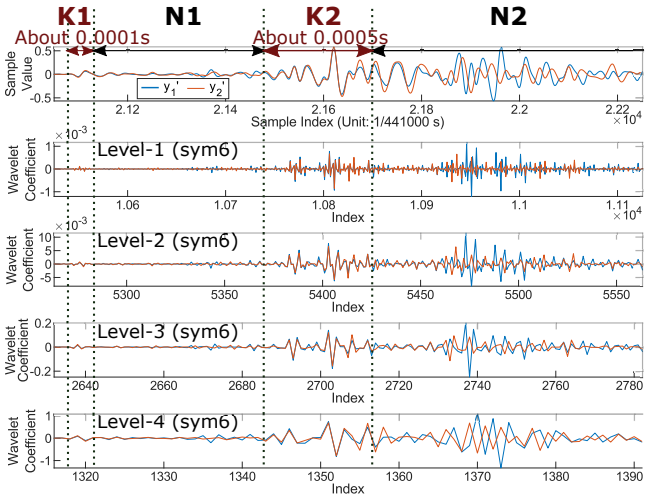

Figure 4: We dissect the keystroke sound signals into key parts K_1, K_2 and noisy parts N_1, N_2. For different levels of coefficients, the transient parts reside in the same time range.

The observations will be helpful in guiding the keystroke signal processing. For instance, we can improve the analysis by focusing more on the information from the part with higher keystroke energy, which also indicates a higher SNR. If we focus our analysis on the low-SNR parts with little keystroke energy, the information may very likely be extracted from environmental factors such as noises. If we directly use the entire keystroke signals and process all parts in the same manner, the long low-SNR parts (about 9/10 of the entire keystroke signal) will obfuscate the details and degrade the effectiveness of the analysis.

Further, the energy distribution of high/low-frequency components will help us to find reference points in keystroke signals for selecting and handling specific parts of the signals for computation. This can be achieved by changing the window size or frequency range of the components. In Fig. 3, we filter and calculate the energy distribution using convolution with a Hanning window. We will explore this more deeply in section 4.4.

2.4 Keystroke sound anatomy model

Prior works [21, 23, 39, 46, 48] universally processed the entire keystroke sounds to extract information. However, these analyses did not reveal the (im)precision and alignment issues of different parts within the keystroke signals (Section 2.2). Further, without understanding the internal components and the complex physics of keystroke sounds, it can be difficult to process the signals for more effective and targeted analysis. For instance, the long low-SNR parts in keystroke sounds can obfuscate the analysis (Section 2.3).

To enable more detailed and targeted analysis within

keystroke signals, we dissect a keystroke sound signal into key parts (K_1 and K_2) and noisy parts (N_1 and N_2). As shown in Fig. 4, we align the signals with an accurate time delay. We can then observe that the K_1 and K_2 include the short burst of energy at the start of the keystroke sound (transients). In these parts, the signals and the corresponding wavelet coefficients are aligned much better compared to Fig. 2. We identify K_1 and K_2 as key areas for localization and measuring precision. Compared to K_1 and K_2, the parts N_1 and N_2 include "tails" after the transient parts and are more susceptible to noises.

3 Threat Analysis

3.1 Precision Analysis

Target. While the concept of localizing a signal source with time-difference-of-arrival (TDOA) has been explored in many scenarios [24, 28, 37], localizing keystrokes in an acoustic side channel can be different because the goal is no longer tracking a single entire source. Ideally, one may expect that keystroke sounds are generated from the exact center points of the keycaps. However, due to the size of human fingers and the accuracy of movements, the finger's landing position usually deviates from the center point. Moreover, the vibration and friction of keyboard components inevitably contribute to the complex sounds. When trying to distinguish more than 50 commonly used keys compactly spaced in a small area (e.g., 27.2×7.1cm for an Apple Magic keyboard), the localized sound sources can significantly overlap because of the physics and imperfect measuring. Thus, it requires processing keystroke sounds and analyzing the internal components to track specific keys instead of the entire keyboard area that emits sounds,

Precision. In localization, the standard deviation (σ) is often used to assess the precision [24, 35]. In order to distinguish a large number of closely spaced keys, the distribution of the localized keystrokes from the same key should be as concentrated as possible. Assuming the ideal time delay measurement is ΔT, with a standard deviation of σ, 68% of the localization measurements will be within $[\Delta T-\sigma, \Delta T+\sigma]$, and 80%, 90% of the measurements will be within $[\Delta T-1.28\sigma, \Delta T+1.28\sigma]$, $[\Delta T-1.65\sigma, \Delta T+1.65\sigma]$ respectively. Fig. 5 shows the maximum standard deviation for identifying keys. In order to make distributions of adjacent keys distinguishable, to achieve a 68% confidence interval within $\frac{\Delta T}{2}$, the max σ will be $0.5\Delta T$. In other words, 32% of the measurements will still have large errors and fall into the confidence intervals of other keys. Note that this analysis is based on ideal cases assuming the signals are generated from the center points of the keycaps. In practice, the standard deviation σ has to be lower due to the imperfect sound source.

Existing approaches for typical indoor sound localization have relatively large error tolerance such as 0.5m with distributed microphone networks [25, 34]. More accurate localization in the submeter range could be achieved with known signals (e.g., ultrasonic chirp) from mobile devices [32, 37]. Researchers [39, 46] also utilized the general localization method [24, 28, 37] to identify a small subset of keys at close distances (e.g., 10cm). These approaches are intended for localizing an entire, general sound source in a relatively coarse-grained manner (Fig. 25), but it requires much finer precision (Fig. 5) to practically localize and identify keystrokes.

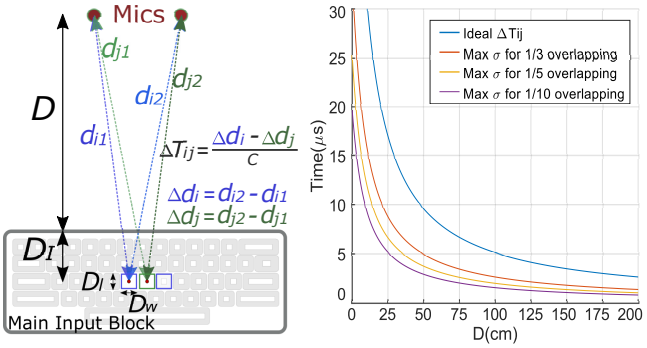

Figure 5: Left: The difference in the ideal time delay (TDoA) measurement of adjacent keys. Right: The minimum precision required to distinguish the keys in ideal situations with a perfect signal source. In practice, because users' typing sounds are not generated from the exact centers of keycaps, the average standard deviation has to be very low to avoid significant overlapping.

3.2 Adversary Model

Unconstrained user inputs. User inputs can include numbers, passwords, user names, addresses, real-world texts with punctuation and capital letters, etc. Further, since it can be challenging for an adversary to trick victims to type consistently with a specific or fixed typing speed/style, we consider users that type with their natural typing styles. The users are free to make/correct typos and adjust their typing styles/speeds.

The adversary may not always limit the solution space to specific known sequences. Because users' natural input sequences are not completely included and previously specified by a given dictionary, the adversary deals with an arbitrary solution space instead of constrained inputs from a single-case alphabetic dictionary or training dataset.

Attack Scenarios. The adversary may deploy devices such as smart speakers and microphones in public, office, or other accessible environments. The devices may appear to the victim as only intended for voice applications that are common in various scenarios [1–3, 9, 12]. An adversary may also gift/sell hacked/modified microphone array-based devices which act as voice assistants in a smart home or office scenario but allow the adversary to surreptitiously track the victim's inputs. Ad-

ditionally, considering the growing number of manufacturers and variety of hardware (*e.g.*, [4, 8, 10, 18, 19]) of voice-based devices, an adversary may exploit vulnerabilities [5, 13, 14, 17] or potential backdoor of existing products. Finally, an attacker may analyze the audio in live streaming and online learning/meeting to localize the victim's keystrokes unintentionally recorded by laptop microphones. An adversary may also deploy devices in non-line-of-sight (NLOS) scenarios (Section 6).

Users can anticipate privacy leakage in speech [36] but may not expect the leakage of accounts, IDs, SSH credentials, emails, and other highly sensitive information that is typically not communicated via speech. By revealing the users' natural, unconstrained inputs, the attack can result in compromised computer systems and leakages of accounts, real-world texts, confidential addresses/dates, and other secret data.

4 Methodology

4.1 Overview

We propose a multi-round structure for tracking keystrokes. This methodology is inspired by observations on different scales of errors in localizing keystrokes and the fact that the keys reside in a compact keyboard area. Since the keys are close to each other and the imperfect keystroke sound components (Sec. 2.1) can obfuscate the localization measurements, our multi-round approach aims to improve the localization capability in more detailed ranges in each round for tracking the typed keys.

Uniformly processing the entire keystroke signal or directly truncating the signal to a filtered small part in a single round can lead to information loss; due to the short duration of transient parts in keystroke sounds (Fig. 4), the result will not be accurate or stable. The multi-round is designed to distinguish compactly spaced keys robustly without causing stability issues.

Multi-round keystroke tracking approach. In the initial round (I-Round), our approach tracks the positions of the keystrokes in the entire space. However, when applying cross-correlation-based methods [37] to localize unknown and short signals like keystroke sounds, it is easy to generate large errors by matching the wrong parts of the signals.

After identifying these large errors (outliers in Fig. 6), we are able to estimate the keyboard range and shift the signals based on the center point for second-round (B-Round) calculation. In B-Round, we calculate the time delays of the shifted signals in the bounded range to avoid large-range errors caused by matching the wrong parts of the signals. However, in this round, there can still be significant overlapping between different keys. Such overlapping is induced by the complex physics and imperfect measuring of keystroke sounds. The errors causing the overlapping are usually in the range of microseconds or tens of microseconds (Fig. 6).

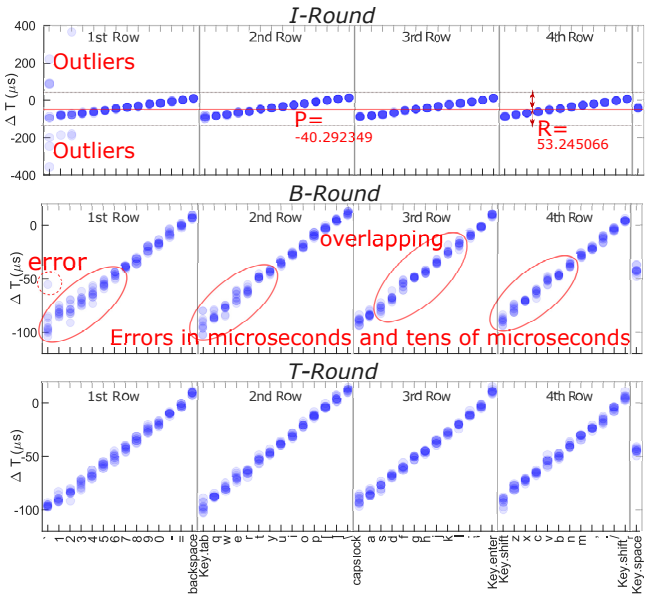

Figure 6: Top: I-Round localization results usually include large-scale errors (outliers). Middle: B-round results still have errors and significant overlapping between keys. Bottom: T-Round reduces the μs-scale errors and the overlapping.

To address such microsecond-scale errors, we need to understand the internal components of keystroke sounds. Our analysis shows that while an entire keystroke sound can last 0.1s, the transient parts are very short **[Sharon: can you give an example here?]** (Sec. 2.2). Further analysis on the internal energy distribution (Sec. 2.3) shows that the high-SNR parts are concentrated in a short time range, such as around 0.001s, corresponding to the K1, N1, and K2 parts of keystroke sounds (Sec. 2.4). Therefore, in the third round (T-Round), we will align the signals based on the results from B-Round, and start focusing on the transient parts of keystroke sounds to derive more precise localization measurements.

Results in Fig. 6 provide a more intuitive illustration to understand the multi-round process. The main principle is to process and compute short keystroke signals in different rounds to gradually deal with different scales of errors in distinguishing the keystrokes. To validate the multi-round approach, we design a customized signal processing chain in each round. Here we discuss the essential mechanism to construct a multi-round structure in keystroke tracking.

Align and Recalculate. Sec. 2.2 discussed how to perceive the (im)precision by aligning the keystroke signals based on the time delay measurements. The *aligning* of signals is essential in each round because keystroke signals are very short, and we want to search the localization results in a more detailed range after each round.

Assuming the keystroke signals recorded by two microphones are $y_1[i]$, $y_2[i]$, $(i \in \{0, 1, 2, 3, ..., L\})$. The cross-

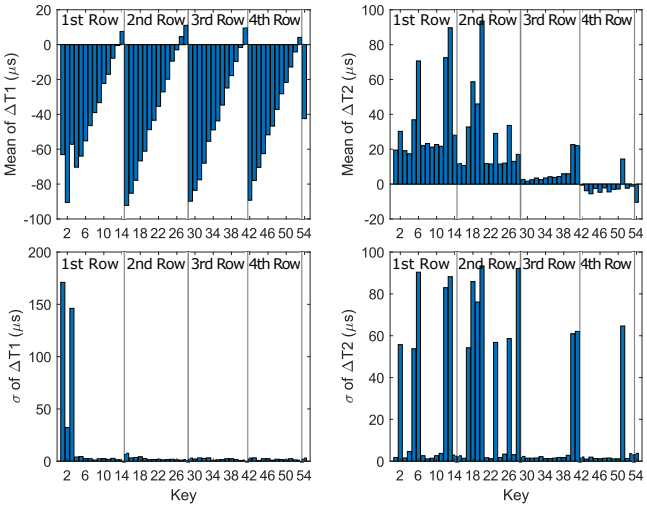

Figure 7: I-round localization statistics of 598 keystrokes on an Apple keyboard. For all statistics figures in this paper, x-axis indexes 1 to 14 stand for keys from ‘ to backspace in the 1st row of the main keyboard block; Indexes 15 to 28 stand for keys Tab to Backslash in the 2nd row; Indexes 29 to 41 stand for keys Capslock to Enter in the 3rd row. Indexes 42 to 53 stand for LeftShift to RightShift in the 4th row. Index 54 stands for key Space.

correlation between signals of microphone i and microphone j at a lag of k_1 (unit: $\frac{1}{F_S}$; F_S is the sample rate) will be calculated as follows,

$$CC_{12}[k] = \sum_{k=1}^{L} y_1[i] y_2[i-k] \tag{1}$$

During the calculation, the signals with a negative index or an index larger than L will be padded zeros. We can determine the delay of the signals at mic *1* and mic *2* with,

$$k_1 = \arg\max_{k_1} CC_{12}[k_1] \tag{2}$$

The default range of k_1 is the signal length (L) of keystroke sounds. In the next round, we pre-align the keystroke signals to $\hat{y_1}[i], \hat{y_2}[i]$, $(i \in \{0,1,2,3,...,L+\Delta N\})$, where $\Delta N = k_1$ (except for the B-Round), we recalculate the time delay:

$$k_2 = \arg\max_{k_2} \widehat{CC}_{12}[k_2] \tag{3}$$

where

$$\widehat{CC}_{12}[k] = \sum_{k=1}^{R_2} \hat{y_1}[i] \hat{y_2}[i-k] \tag{4}$$

[Sharon: Have you explained how to use k_1 and k_2?] In each round, we align keystroke signals based on the information from the last round. The computation will then be conducted on the pre-aligned signals to derive finer localization measurements. This *align and recalculate* mechanism is the essential mechanism that enables the multi-round keystroke tracking approach. By aligning the signals prior to recalculation, we can also mitigate the asymmetry at the front and end parts of the two signals. When the signals are short, such asymmetry can also interfere with the calculation.

4.2 I-Round: Initializing and Preprocessing

In the initial round (I-Round), we 1) filter the signals; 2) interpolate the keystroke signals to scale the time unit to sub-microsecond; 3) calculate the time delay measurements and detect large-scale errors (outliers); 4) estimate the range of time delay measurements of the keystroke signals.

Initial Processing. We filter the keystroke signals with a high-pass filter; we use a zero-phase Butterworth filter (1kHz) to mitigate noises without affecting the localization information in keystroke signals. The unit of the calculated time delay (k_0) result is not the continuous time since the recorded keystroke signals are digital. To perceive and subsequently mitigate the imprecision in μs-scale ranges, we will process the digital signals to scale the unit to smaller than 1 μs. For example, the unit of standard-resolution (44.1 kHz) audio recording devices is as large as 0.0227 s. We interpolate the signals with a 40-times resample rate of 1,761 kHz to scale the unit to 0.5686 μs.

Outlier Identification. I-Round results usually contain large-scale localization errors (Fig. 6). As shown in Fig. 7, the standard deviations $\sigma(\Delta T)$ for certain keystrokes are very large. Because all keys reside within a small physical area and these measurements are far away from others, we can identify these errors by outlier detection.

We identify far-away outliers by detecting measurements that are more than three scaled median absolute deviations (MAD [38]) from the median of all measurements. We then identify remaining outliers based on percentile ranges [s%, (100-s)%]. This range can be adjusted with s (typically, $s \in [2,10]$), depending on the number of outliers.

After temporarily removing the outliers from the measurements, we can estimate the center reference point P and range R for the keyboard. P is the mean of all measurements that are not outliers. R is half of the difference between the maximum and minimum measurements. Keystrokes corresponding to the outliers will not be discarded; the measurements for all keystroke signals will be recalculated in the following rounds.

4.3 B-Round: Bounding the Range

In the second round (B-Round), we align all keystroke signals based on the center reference point P. We then calculate the time delays of the pre-aligned signals in the range of [-$R\alpha$, $R\alpha$]. The value of α can be adjusted around 1. We set it to $\frac{100+s}{100}$ by default to compensate for the removed outliers in outlier detection and range estimation in I-Round. As shown in Fig. 8, the outlying large errors in I-Round are reduced.

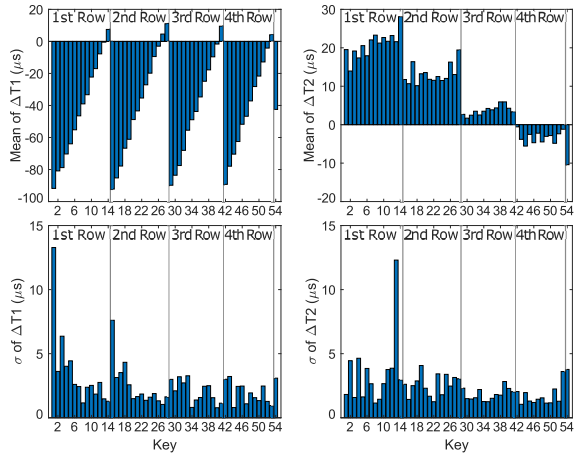

Figure 8: B-round localization statistics. The average $\sigma(\Delta T_1)$ is 2.5428 μs. The average $\sigma(\Delta T_2)$ is 2.4495 μs.

However, there are still overlapping and errors in the range of microseconds and tens of microseconds (Fig. 6). The means of ΔT in different rows still overlap (Fig. 8, Top-right), and the standard deviations for some keys are relatively large.

4.4 T-Round: Focusing on Transients

We observe that the transient parts at the beginning of keystroke sounds (including K1, N1, K2) are very short (within one or a few milliseconds) but can provide more precise localization measurements. A challenge in utilizing specific parts of keystroke signals is that all keystrokes are unique, and the ranges of transient parts vary for different keystrokes.

In the third round (T-Round), we align all keystroke signals based on each of their time delays derived in B-Round. Because the transient parts of keystrokes are short, aligning based on B-Round measurements not only helps to limit the cross-correlation window but also reduces the asymmetry on the front and end parts of two truncated signals.

Signal Dissection. To reliably and automatically select the transient parts, we conduct convolution computation based on specific windows and band-pass filters.

We first delay-and-sum the keystroke signals based on the B-Round time delay results. This operation beamforms the keystroke sound and can improve the signal-to-noise ratio. We then compute the convolution based on Hanning windows using band-pass filtered high and low-frequency signal parts.

We observe that the peak indexes of the filtered signals usually become more stable when the Hanning window size reaches 5,000 for the interpolated keystroke signals (Fig. 9, a). We observe that this window size (5,000) works well for different keystroke instances. The noisy tails can contain more interference and have low SNR. We also observe that the distribution of low (<4 kHz) and high-frequency (> 4kHz) energy can be different for different keystrokes. Using the

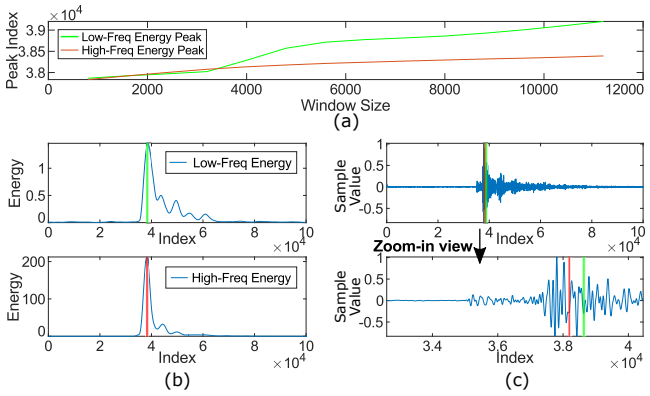

Figure 9: Extracting the transient parts of keystroke signals. (a) The peak indexes become more stable as the window size increases and reaches around 5000 samples. (b) Energy calculated based on convolution with a window size of 5000. (c) The range of signal containing transient parts can be found with the energy peak indexes.

minimum value of the low-frequency and high-frequency peak indexes (Fig. 9, c), we can reliably select the signal with transient parts of the keystroke and remove the noisy tail after the transient part K2.

After aligning the keystroke signals based on B-Round results, we calculate the time delays by cross-correlation on the transient parts. By recalculating using the selected signal consisting of transient parts, the μs-scale errors can be mitigated. As shown in Fig. 10, the standard deviations become very low. This means that the localization measurements for each key become highly concentrated. We observe that this same process can be applied to different keyboards, such as the Razor keyboard to extract the transient parts of the keystroke signals.

4.5 Calibration Rounds (C-Rounds)

C-Rounds are additional, extendable rounds following similar ideas to further select keystroke signals and deal with smaller ranges of errors. We implement C-Rounds to address small quantization errors in computing digitized keystroke sounds. We first select a closer range of transient signals, such as a point with (1/10) peak energy in front of the peak. Since the signal is very short, we can interpolate the signals with a high sample rate, such as 88,200 kHz. These steps will make the localized measurements more continuous. They may slightly concentrate the measurements, but such changes will not be significant in statistics because the quantization error is small.

4.6 Evaluation

We evaluate the attack on an Apple Magic keyboard and a Razor Blackwidow mechanical keyboard. We illustrate the settings and microphone indexes (1 to 6) in Fig. 11. We collect

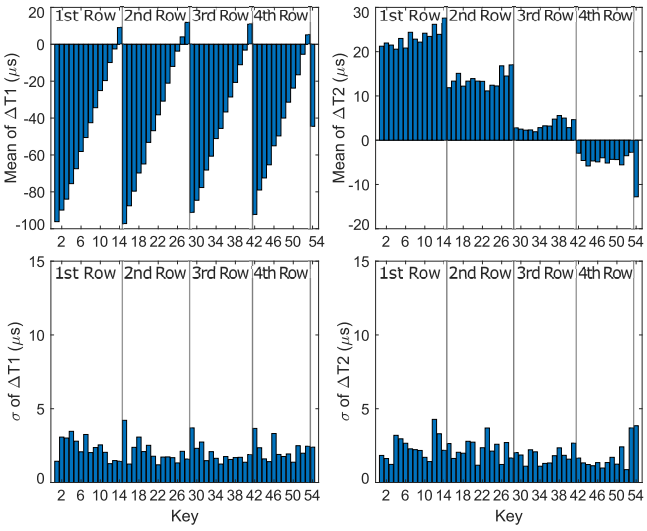

Figure 10: T-round localization results' statistics. The average $\sigma(\Delta T_1)$ is 2.1278 μs. The average $\sigma(\Delta T_2)$ is 2.0326 μs.

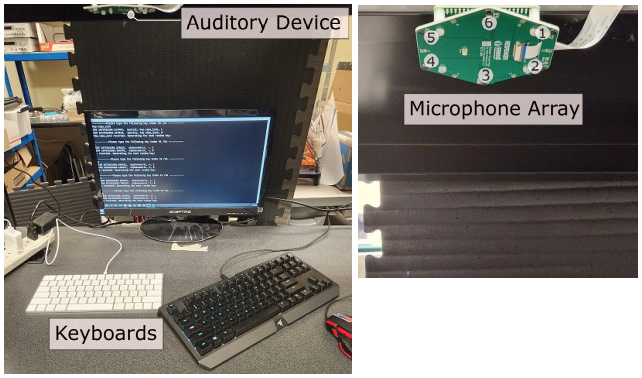

Figure 11: Left: Settings. Right: Zoom-in view of the microphone array on the auditory device. The vertical distance between the microphones and the keyboards is about 0.5m.

more than 11 keystrokes for each key in the main input block of the keyboard. There are 54 commonly used keys for inputting information, including the four rows of the main input block and space keys. These keys include alphabetic, numeric, editing, punctuation, and symbols in all QWERTY keyboards. In total, we collect 594 keystrokes and 595 keystrokes on the Apple and Razor keyboard, respectively. The adversary auditory device is a ReSpeaker circular microphone array [15] connected to a Raspberry Pi. We use two pairs of microphones to derive time delay measurements (ΔT_1 and ΔT_2). The device supports a standard recording sample rate of 44.1 kHz. Devices with similar hardware are common in various voice applications [4, 10, 16].

As shown in Fig. 12, we observe that in the I-Round results on the Razor keyboard, there are large-scale errors. The B-round addresses the large-scale errors but still shows μs-scale

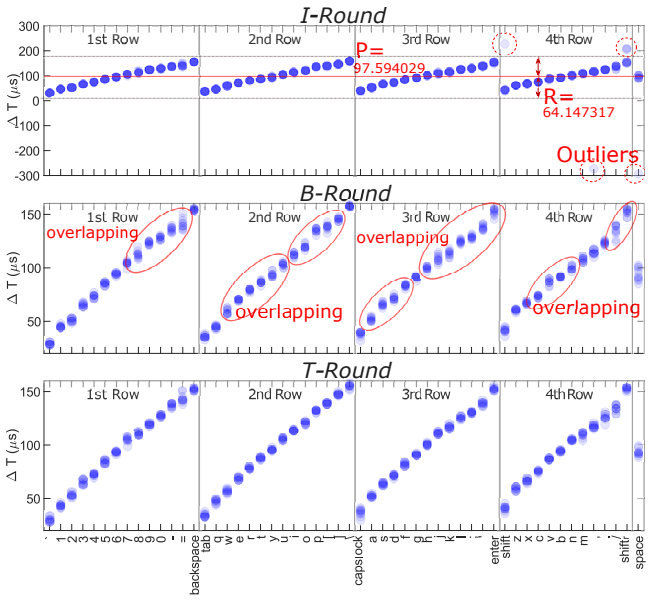

Figure 12: We apply the same approach to keystrokes on a Razor mechanical keyboard. Top: I-round localization results include large-scale errors. Middle: B-round results reduced the very large errors but still show overlapping between keys. Bottom: T-round results show that our approach significantly reduced the errors and overlapping.

Table 1: *n*th-attempt accuracy of 594 keystrokes on an Apple keyboard and 595 keystrokes on a Razor keyboard

Keyboard	1st	2nd	3rd	4th
Apple	90.64%	98.16%	99.50%	100.00%
Razor	96.47%	99.16%	99.50%	99.83%

errors causing overlapping between keys. The final results show that our approach also significantly reduces the errors on keystrokes from the Razor keyboard. We calculate the recovered keys based on the Euclidean distance between the sample and the mean value of measurements of a key. The 1st-attempt is the key closest to the sample localization result. The nth-attempt is the key nth-closest to the result. The attack achieves high accuracy on both keyboards (Tab. 1). The average standard deviation of the localization reaches around 1 or 2 μs (Tab. 2). The localization results in each round are illustrated in Appendix.

Our approach is based on analyzing the physical properties and measuring of keystroke sounds. There is no need to train or learn features for a specific keystroke or keyboard. We apply the same approach to both keyboards.

5 Recovering Unconstrained Information

We collect keystrokes from different users (Fig. 13). The vertical distance between the auditory device and the keyboard is about 50 cm. The users do not need to type with a spe-

Table 2: Average standard deviation (unit: μs), indexes of the utilized microphones to derive localization measurements, and the separation distance D_s between the two microphones

	Apple ΔT_1	Apple ΔT_2	Razer ΔT_1	Razer ΔT_2
σ	2.1339	2.0272	1.6274	1.3890
Mic Indexes	4-2	6-3	5-2	1-3
D_s	8.1 cm	9.3 cm	9.3 cm	8.1 cm

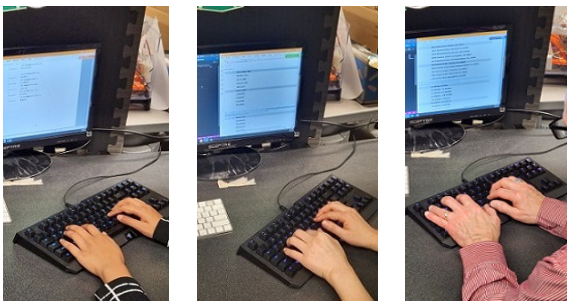

Figure 13: Experiment settings. The users type with their natural typing styles (touch typing). They can make typos and are free to adjust typing styles/speeds based on the typed contents or their own willingness, statuses, and habits.

cific, fixed style or speed. The typed information covers dates, IDs, addresses, GPS coordinates, real-world texts that include punctuation and capital letters, usernames, passwords, and SSH credentials.

We observe that users' touch typing usually generates more noises in keystroke sounds. The movements of different users are also different to reach for the keys. For the same user, his/her typing speed and style vary based on the typed content. For example, their typing speeds are relatively low when inputting numbers and complex passwords. When inputting texts, their typing speeds can reach higher. The users watch the screen most of the time while typing. Further, the users press the Shift keys frequently when typing capital letters in texts or symbols in strong passwords. When reaching out for the Shift keys, the typing movements are very different from typing other keys. Different users or scenarios may use Shift keys on different sides. In the experiments, the users are free to make or correct typos. They are free to adjust typing styles/speeds based on the typed contents or their own statuses, feelings (such as the heaviness of hands), and typing habits. The three users include 1 male and 2 female users. Their ages are distributed across three ranges: 20-30, 40-50, and 50-60. Each of these users speaks a different first language and originates from a different country.

Research Ethics. The studies involving human participants were reviewed and approved by our institution (IRB). We obtained informed consent from the participants before the data collection.

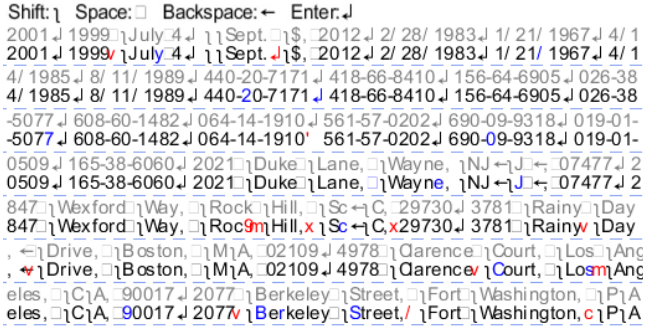

Figure 14: A snippet of the typed and recovered dates, social security numbers, addresses, and GPS coordinates. Characters in gray are the ground truth. Characters in black, blue are correctly identified results in first, second attempts, respectively. Characters in red are errors.

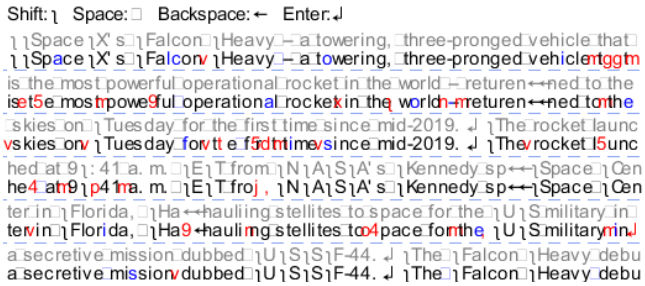

Figure 15: A snippet of recovered unconstrained text inputs.

5.1 Data Recovering

ID numbers, dates, addresses, GPS coordinates. Fig. 14 shows the typed and recovered information. We can observe that a real-world user often types backspace keys to correct typos, and types Shift keys to input capital letters. There are hyphens, commas, and period keys when typing ID numbers (e.g., social security) and GPS coordinates. Most ID numbers are identified correctly with the first and second-attempt results. The dates are correctly identified from the recovered information. There can be one or two errors in the characters of each address, but all addresses can be recognized. Most of the digits in the GPS coordinates are correctly identified.

Real-world Texts. Real-world texts are not purely alphabetic. They include punctuation, capital letters, numbers, and many other keys. Further, the input sequences may not be fixed due to the use of the Shift and Backspace keys. The user may correct a typo and may frequently type the left/right Shift keys to input capital letters.

Because most enter, backspace, and shift keys are correctly identified, the structure of the text is clear in the recovered data (Fig. 15). Sentences in the recovered results are readable. Details such as the numbers and abbreviation codes are mostly correct. There can be more errors in the characters when recovering real-world texts compared to other inputs because

Figure 16: Recovering usernames and 7/8-more-digit passwords. We can observe that a real-world user may make typos and corrections when inputting passwords.

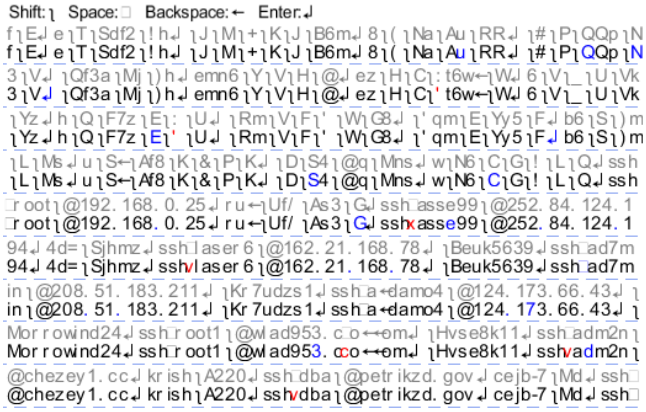

Figure 17: Recovering strong passwords and SSH credential inputs, which include a lot of symbols, capital letters, numbers, and punctuation.

the users are typing very fast. In this scenario, a keystroke signal is more frequently affected by the tail of the previous signal. We also observe a lower accuracy on user B, who types faster than users A and C.

User names and passwords. We randomly select usernames and 7/8-more-passwords [6]. The 7-more-passwords consist of passwords with 7 characters or more. The 8-more-passwords contain more than 8 characters and exclude all-numeric passwords, consecutive (3 or more) characters, all-lowercase passwords, and strings without both a capital letter and a number [6]. Most of the usernames and passwords are completely recovered within two attempts (Fig. 16). The inputs, including capital letters, shift keys, underlines, and numbers, are correctly identified. Most of the enter keys indicating the end or submission of the passwords are also recovered.

Strong passwords and SSH credentials. We then recover randomly generated strong passwords that contain lowercase and uppercase letters, numbers, and special characters simultaneously. To demonstrate the threats, we recover SSH credentials that include numbers, special characters, etc. Most of the characters in strong passwords and credentials can be recovered within two attempts (Fig. 17). Most Shift keys and special characters (such as @) are correctly identified in the first attempt.

We collect and evaluate a total number of 7,719 keystrokes from users typing unconstrained information (Tab. 3). We calculate the recovered keys based on the Euclidean distance between the sample and the reference mean value of measurements of a key. We derive the reference mean value of the measured samples of a key excluding outliers that are more than three scaled median absolute deviations (MAD [38]) from the median of all measurements and outliers outside the percentile range of [10%, 90%] of remaining measurements. We calculate the accuracy by comparing the recovered keys and the actual typed keys (ground truth); we run a Python program on the computer connected to the keyboard to capture the key press and release events. We record the actual typed sequences that include typos and Shift keys. This setting allows us to better understand and study the threats. A real-world attacker can always carefully examine, filter, slice [21], and listen to the recorded sounds to identify signals containing the keystrokes, and make use of the recovered keystroke localization information to gain a substantial advantage in revealing the unconstrained input contents.

Table 3: *n*th-attempt accuracy, correctly identified keystrokes and the total number of keystrokes of user inputs.

User	1st Attempt		2nd Attempt		Total
	Accuracy	Correct	Accuracy	Correct	Keystrokes
A	90.6%	2,635	95.3%	2,773	2,909
B	83.8%	2,018	92.5%	2,228	2,408
C	89.3%	2,145	93.8%	2,253	2,402

6 Exploring Bendable Eyesight

While cameras are usually considered as more invasive sensors compared to microphones, auditory devices can pose less noticeable threats to approach the users. Moreover, we find that in certain scenarios, even when the line-of-sight view is blocked, our attacks can still reveal the typed keys.

6.1 Attack Scenario 1: Covert Typing

Security-aware individuals may try to cover the typed keys to avoid surveillance. A camera's view in such non-line-of-sight (NLOS) scenarios will be blocked (Fig. 18). Since sounds do not travel in straight lines, the refracted sound waves can still be measured and analyzed with Auditory Eyesight. We find that the localization information is not completely lost in the refracted keystroke sounds after multi-path transmission in NLOS settings. For instance, we recover the inputs from two users type individual keys, GPS numbers, user names, and passwords while using another hand to cover the typed area. Our results (Fig. 19) show that the typed information can still be recovered. With accuracy lower than line-of-sight scenarios, the attack can still correctly identify most of the keys in less than 3 attempts (Table 4).

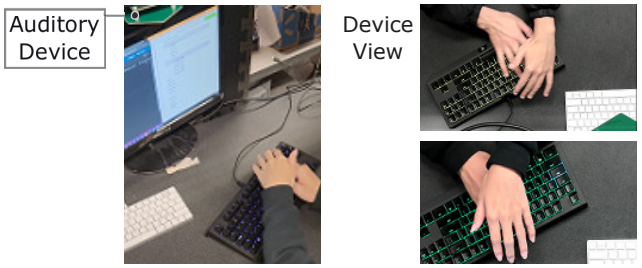

Figure 18: Non-line-of-sight (NLOS) attacks on covert typists who block the keys while typing.

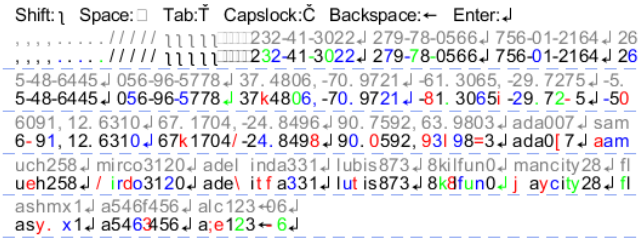

Figure 19: Recovering unconstrained inputs in NLOS covert typist scenario. Gray: actual inputs; Black: 1st-attempt results; Blue: 2nd-attempt results; Green: 3rd-attempt results.

Table 4: *n*th-attempt accuracy, correctly identified keys, and total number of keystrokes of covert user inputs.

User	1st Attempt		2nd Attempt		3rd Attempt		Total
	Accu.	Corr.	Accu.	Corr.	Accu.	Corr.	Keystrokes
N1	74.3%	378	88.4%	450	93.5%	476	509
N2	56.8%	269	75.3%	357	84.4%	400	474

6.2 Attack Scenario 2: Laptop Attack

We find that adversaries can recover the victim's inputs without pointing any sensors toward the victim. Specifically, adversaries can inconspicuously conduct the attack by putting a laptop on the table next to the victim. As shown in Fig. 20, the microphones are on the screen side of the adversary's laptop. The horizontal distance from the adversary's laptop to the upper edge of the victim keyboard is about 40 cm. The laptop uses a standard recording sample rate of 44.1 kHz. The microphone separation distance D_s is 9.6 cm. In a common and inconspicuous scenario, the adversary won't see the victim's inputs because the devices have blocked the direct line-of-sight (Fig. 20) while the victim is typing. This setting is common in libraries, meeting rooms, or office scenarios. Usually, it can be hard for the victim to notice the threats because there is no camera or other sensors on the back of the adversary's laptop.

Although the localization errors are much larger because of the NLOS transmission of keystroke sounds, they can be mitigated with our multi-round keystroke localization approach (Fig. 21). We collect 601 keystrokes (more than 11 keystrokes for each key) in the NLOS setting. In the I-round results, there

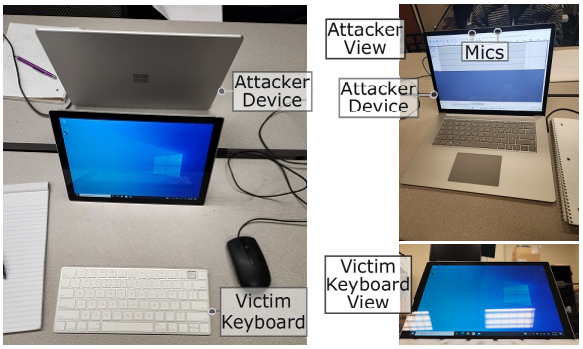

Figure 20: NLOS attack settings with a laptop. The adversary uses a laptop to collect and analyze the keystrokes without pointing any camera or other sensors to the victim's keyboard.

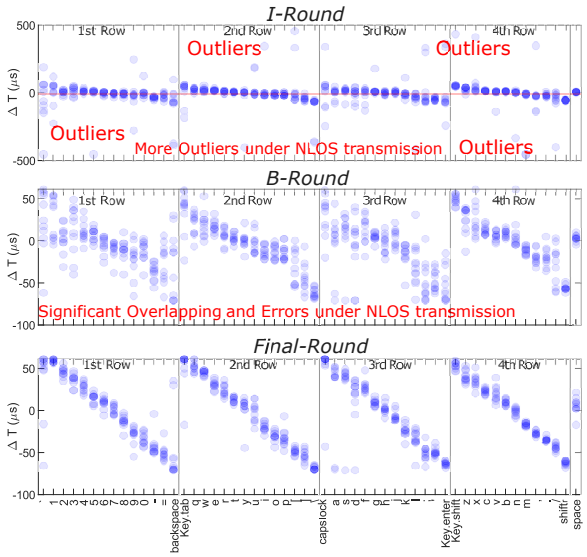

Figure 21: Results of NLOS laptop-based attacks. Our multi-round approach can effectively reduce the excessive errors caused by NLOS keystroke sound transmissions.

are large-scale errors (outliers). In the B-Round, after conducting the computation in the bounded range, there are still significant overlapping and errors. The causality of these errors is that when the line-of-path transmission is blocked, the keystroke sounds have to diffract around the objects to reach the microphones. This induces significant multi-path effects, resulting in more noises and artifacts in the recorded signals. By multi-round customized signal processing and focusing on the transient parts of keystroke signals, we show that our approach reduces the errors caused by NLOS transmissions (Fig. 21). There will be significant overlapping between keys with conventional methods [28, 39, 46] (Fig. 32).

7 Discussion

Distance. We evaluate the localization at 1m and 2m distances from the keyboard. We measure the horizontal distance be-

Table 5: nth-attempt accuracy (in percentage) on 601 keystrokes in the laptop-based NLOS attack.

1st	2nd	3rd	4th	5th	6th	7th	8th	9th
21.96	42.10	54.91	68.05	75.54	82.36	85.19	89.35	91.18

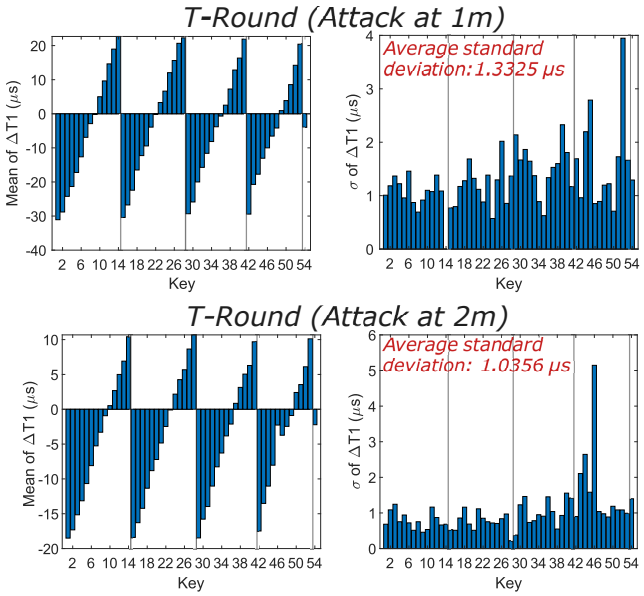

Figure 22: Localization statistics on an Apple Magic keyboard at 1 and 2-m attack distances.

tween the laptop microphones and the keyboard's edge. The horizontal distance between the microphone center to the keyboard center is about 106.6 cm, 206.5 cm, respectively. We collect over 594 keystrokes (at least 11 keystrokes from each key) in each scenario. The average standard deviations are 1.3325 μs at 1m, and 1.0356 μs at 2m (Fig. 22). Fig. 23 illustrates the localization results in different rounds at 2m.

Angles. We evaluate the attack at different angles (Fig. 24). Test cases 4 and 5 use three microphones to form the two microphone pairs. The separation distance D_s of the microphone pairs m5-1 (microphones 5 and 1, Fig. 11) and m1-3 is 8.1cm. D_s of m6-3 is 9.3cm. D_s of m6-1 and m6-5 is 4.8cm. The microphone indexes are labeled in Fig. 11. The setting is similar to Sec. 4.6 except that the device is positioned differently. We categorize the accuracy using the T-round localization

Table 6: nth-attempt accuracy, correctly identified keys, and the total number of keystrokes with different attack angles.

Test Case	1st Attempt		2nd Attempt		3rd Attempt		Total
	Accu.	Corr.	Accu.	Corr.	Accu.	Corr.	Keystr.
1 (m5-1, m6-3)	94.1%	560	99.0%	589	99.8%	594	595
2 (m5-1, m6-3)	69.4%	412	86.4%	513	92.1%	547	594
3 (m5-1, m6-3)	89.2%	530	97.1%	577	98.7%	586	594
4 (m6-1, m1-3)	86.5%	514	96.1%	571	99.2%	589	594
5 (m6-2, m6-5)	85.7%	509	98.2%	583	99.3%	590	594

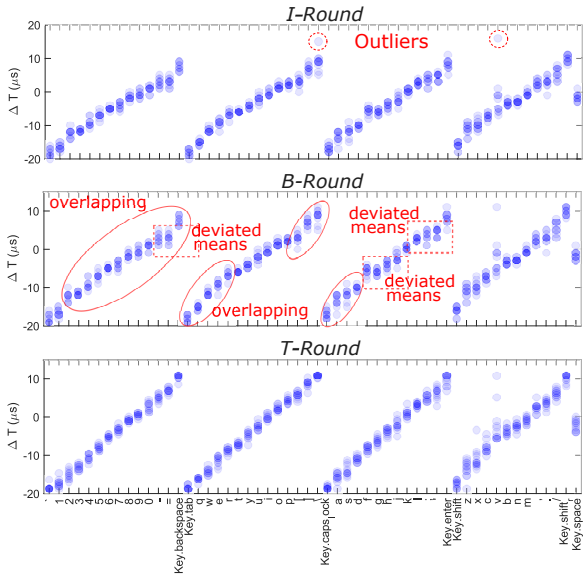

Figure 23: Top: I-Round localization results on an Apple Magic keyboard at 2-m attack distance. Middle: B-round results show significant deviated means and overlapping between keys. Bottom: T-Round reduces the μs-scale errors and the overlapping. The range of the time delay has become quite small ($[-19, 11]$ μs) at 2-m attack distance.

results. Our experimental results (Tab. 6) show that the attack is effective with different angles and microphone separation distances.

Noise. We have conducted our experiments in an ordinary office (42-44 dBA) adjacent to the entrance of a large building. We then evaluate the attack performance under different noise levels. We place two full-range speakers at a distance of 0.5 m from the keyboard and play noises. The noises are audios consisting of noisy human conversation and activities from publicly available high-quality recordings [7]. We play the audio repetitively while recording the keystroke sounds. The noise sound level is measured with a sound level meter at 10 cm in front of the speakers. Results (Tab. 7) show that the attack is robust under low-to-moderate noises (*e.g.*, 55 dBA).

Keyboard Displacement. When the user momentarily ceases typing to adjust the keyboard's position, there is usually a break in the typing sounds. This adjusting movement can cause the keyboard to produce sounds of friction against the desk. The friction sounds are distinguishable from keystroke sounds. By listening to the recorded audio and examining the spectrogram, an attacker could discern these moments of adjustment. Assuming the user does not frequently move the keyboard while typing, the adversary may divide the recorded keystrokes into segments after the keyboard's displacement and recover the results separately. Additionally, if the user slightly adjusts the keyboard, the attack will still work; however, it may localize an adjacent key due to the displacement. In the future, it might be possible to combine other

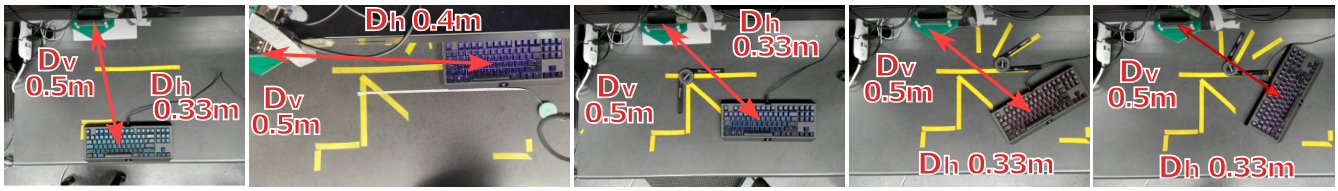

Figure 24: Test cases 1 to 5 (left to right) to evaluate the attack with different angles. D_v and D_h are the vertical and horizontal distances between the microphone array center to the keyboard center.

Table 7: *n*th-attempt accuracy, correctly identified keys, and the total number of keystrokes under environmental noises.

Keyboard (Noise Level)	1st Attempt Accu.	1st Attempt Corr.	2nd Attempt Accu.	2nd Attempt Corr.	3rd Attempt Accu.	3rd Attempt Corr.	Total Keystr.
Apple (55dBA)	95.5%	579	97.8%	593	98.0%	594	606
Apple (65dBA)	58.2%	351	69.7%	420	74.8%	451	603
Razor (55dBA)	95.3%	570	99.5%	595	100.0%	598	598
Razor (65dBA)	94.7%	571	99.3%	599	99.8%	602	603

side-channel information to improve the robustness of the attack when the keyboard moves.

8 Related Work

Typing on a keyboard is a physical process that can emit and perturb physical signals in the environment. Such side-channel [42] signals could allow adversaries to recover the victim's input information without connecting to or visually monitoring (*e.g.*, via cameras, shoulder surfing) the keyboard.

Acoustic-based keylogging attacks. Prior works utilized Fast Fourier Transform (FFT) and Mel-frequency cepstral coefficients (MFCC) features of keystrokes to classify the typed keys with supervised learning [21, 27]. Recent papers leveraged deep learning to improve the effectiveness of the attacks [30, 43]. Training-based methods require collecting a substantial amount of labeled data from the victim. Moreover, it can be challenging to ensure that the victim types with consistent or fixed style/pattern for stable classification performance. Giallanza *et al*. observed that the recurrent network takes advantage of common phrases and letter sequences, but such sequences are lacking in strong passwords [30]. Further, users' actual typing sequences in unconstrained settings can be different when typing capital letters, symbols, punctuation, and making/correcting typos; it can be hard for an adversary to predict and include the sequences in a training dataset.

Dictionary-based approaches [23, 47, 48] associate acoustic features to the keys with clustering. However, the attacks assume that the inputs are texts (lower-case words separated by spaces) only and should not contain other keys or any sequence that is not included in the dictionary.

Zhu *et al*. leveraged smartphones close to the keyboard (10 cm) to recover texts containing lower-case alphabetic words and backspaces [46]. Liu *et al*. [39] studied close-proximity attacks to classify the 26 alphabetic keys. These approaches [39, 46] are based on general techniques [24, 28, 37] to localize an entire, larger-scale target. Directly applying general sound localization techniques on keystroke sounds can lead to undesired distinguishability (Fig. 25).

Prior acoustic keylogging attacks [21, 27, 30, 39, 43, 46–48] focused on constrained settings and inputs such as alphabetic keys/words [21, 27, 39, 46–48] and known sequences [27, 30, 43, 48] in a given dictionary or training dataset. The FFT/MFCC features were universally extracted from the keystroke sound signals [21, 27, 30, 30, 39, 43, 47, 48] (Table 8). Similarly, the time delay information was based on one-round cross-correlation (CC) calculation [24, 28, 37] directly using the entire recorded keystroke sounds [23, 39, 46].

Wireless signal perturbation, vibration, and heat-based keylogging attacks. Many studies utilized supervised learning [20, 26] and dictionary-based [29] methods to classify single-letter-case alphabetic keys/words via wireless signals. Researchers explored vibration and motion sensors to recover alphabetic keys/words [40, 41, 45]. Recently, Kaczmarek *et al*. showed that thermal residues measured with a thermal camera can hold keystroke information for 0.5 to 1 minute [33].

In summary, prior works studied mapping side-channel signals to constrained keyboard inputs (e.g., single-letter-case alphabetic keys/words) [20, 21, 27, 29, 39, 45–48] and known sequences [27, 29, 30, 43, 48]. However, these methods 1) exclude most real-world text inputs or passwords that may not strictly follow previously specified sequences in a dictionary or training dataset, 2) require substantial labeled training data or assume single-letter-case alphabetic keys/words, and 3) may require constrained/fixed user-specific typing style. We develop the first keystroke side-channel attack study on unconstrained inputs, including non-alphabetic keys and sequences that are not necessarily in a known dictionary or training dataset, such as real-world text inputs (with punctuation, typos, numbers, and capital letters) and strong passwords.

Further, prior works [21, 23, 39, 46–48] focused on processing keystroke sounds without distinguishing the internal sound components, using the entire keystroke sounds to extract information. We discover that the sound components and the underlying physics study allows the attacker to extract more targeted and accurate information.

Table 8: Comparisons of our paper to prior keyboard acoustic side-channel attack studies.

Methodology	**Signal handling process**†	**Specific analysis domain(s) and target level**	**Focused inputs and recovered info**‡
Supervised learning [21, 27]	Universal processing	FFT, MFCC	Keys/words (Alph. Sing.)
Deep learning [30, 43]	Universal processing	FFT, MFCC	Texts
Clustering [23, 47, 48]	Universal processing	FFT, MFCC, CC (entire keystroke)	Words (Alph. Sing.)
General localization [39, 46]	Universal processing	CC (entire keystroke)	Keys/words (Alph. Sing.)
AuditoryEyesight: Customized signal processing chain inspired by imperfect measuring and keystroke sound physics analysis	Internal sound component and multi-round processing	Temporal analysis and frequency-energy analysis (on internal transient and noisy parts), interp., align and recalc.(within keystrokes to μs-range)	Unconstrained (with unknown sequences and non-Alph. keys)

† **Universal processing** handles all components of the signals equally in a relatively coarse-grained, large-scale manner, which tends to bury the detailed and accurate information due to imperfect physics and measuring of keystroke sounds.

‡ **Alph.**: Alphabetic. **Sing.**: single-letter-case. **Unconstrained inputs**: inputs that include non-alphabetic keys and sequences that are not necessarily in a known dictionary or training dataset, such as real-world text inputs and passwords.

9 Conclusion

Keyboard inputs contain highly sensitive data including SSH confidentials, social security numbers, and real-world texts such as emails. The confidentiality of such information is critical to privacy and also to the security of systems authenticated with keyboard inputs. This work explored the leakage from such unconstrained keyboard inputs via increasingly ubiquitous voice sensing devices. We also observed that although diffraction of sounds waves was usually considered as an undesired property for high-precision localization measurement, it brings unintended benefits over light sensors to allow attacks on keystrokes in certain NLOS scenarios.

References

[1] Alexa can now pay for gas at over 11,500 Exxon and Mobil stations in the US. https://techcrunch.com/2020/09/01/alexa-can-now-pay-for-gas-at-over-11500-exxon-and-mobil-stations-in-the-u-s/. Sep. 1, 2020. Last accessed: 2021-08-10.

[2] Alexa for Business: Cheat sheet. https://www.techrepublic.com/article/cheat-sheet-alexa-for-business/. May 14, 2021. Last accessed: 2021-08-11.

[3] Alexa for Business. Use Alexa for work. https://aws.amazon.com/alexaforbusiness/. Last accessed: 2021-09-01.

[4] Alexa on Sonos. https://www.sonos.com/en-us/alexa-on-sonos. Last accessed: 2021-09-04.

[5] BlueBorne Vulnerability Also Affects 20Mil Amazon Echo and Google Home Devices. https://www.bleepingcomputer.com/news/security/blueborne-vulnerability-also-affects-20mil-amazon-echo-and-google-home-devices/. Accessed at 2022-04-17.

[6] Bruteforce Database - Password dictionaries . https://github.com/duyet/bruteforce-database. Accessed at 2022-06-02.

[7] Conversation noises (FreeSound). https://freesound.org/people/rampartian/sounds/236786/. Last accessed: 2023-05-02.

[8] Global Smart Speaker market shipments hit 154 million in 2020. https://omdia.tech.informa.com/pr/2021-feb/global-smart-speaker-market-shipments-hit-154-million-in-2020. Last accessed: 2021-09-30.

[9] How Amazon Alexa can be a helpful office cubicle companion. https://www.gearbrain.com/using-alexa-in-the-office-2639004111.html. June 27 2019. Last accessed: 2021-08-10.

[10] How to use Amazon Alexa with Bose. https://www.bose.com/en_us/better_with_bose/amazon-alexa.html. Last accessed: 2021-07-02.

[11] How voice-activated devices listen to you (and how to turn them off) . https://www.usatoday.com/story/tech/news/2016/03/02/how-voice-activated-devices-listen-you-and-how-turn-them-off/81187578/. Accessed at 2022-03-02.

[12] Marriott Puts Alexa in Hotel Rooms . https://voicebot.ai/2018/10/30/marriott-puts-alexa-in-hotel-rooms/. Oct. 30, 2018. Last accessed: 2021-08-10.

[13] Researcher finds Google Home speaker vulnerable of getting hacked and snooping on conversations . https://www.businessinsider.in/tech/news/google-home-can-be-hacked-and-snooped-on-listening-conversations/articleshow/96704649.cms. Accessed at 2023-01-05.

[14] Researchers Turn Amazon Echo Into an Eavesdropping Device. https://www.bleepingcomputer.com/news/security/researchers-turn-amazon-echo-into-an-eavesdropping-device/. Accessed at 2022-02-05.

[15] ReSpeaker 6-Mic Circular Array Kit for Raspberry Pi. https://www.seeedstudio.com/ReSpeaker-6-Mic-Circular-Array-Kit-for-Raspberry-Pi.html. Last accessed: 2021-06-25.

[16] ReSpeaker Core v2.0 - Alexa Demo. https://youtu.be/q7b8iLqRiPY. May 31, 2018. Last accessed: 2021-07-02.

[17] See How Amazon Echo is taken over by the "BlueBorne" Bluetooth Hack. https://www.wpxbox.com/see-amazon-echo-taken-blueborne-bluetooth-hack/. Accessed at 2022-04-17.

[18] Smart Speaker Market Size In 2021 with Top Countries Data, Leading Manufacturers and Key Insights to 2025. https://www.rfdtv.com/story/44783867/smart-speaker-market-size-in-2021. Last accessed: 2021-09-10.

[19] Smart Speakers Global Market Report 2021: COVID-19 Growth And Change To 2030. https://finance.yahoo.com/news/smart-speakers-global-market-report-114500729.html. Last accessed: 2021-09-10.

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

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

[22] Md Abdul Awal, Sheikh Shanawaz Mostafa, Mohiuddin Ahmad, and Mohd Abdur Rashid. An adaptive level dependent wavelet thresholding for ecg denoising. *Biocybernetics and biomedical engineering*, 34(4):238–249, 2014.

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

[24] Peter Bona. Precision, cross correlation, and time correlation of gps phase and code observations. *GPS solutions*, 4:3–13, 2000.

[25] Alessio Brutti, Maurizio Omologo, and Piergiorgio Svaizer. Speaker localization based on oriented global coherence field. In *Ninth International Conference on Spoken Language Processing*, 2006.

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

[27] Alberto Compagno, Mauro Conti, Daniele Lain, and Gene Tsudik. Don't skype & type! acoustic eavesdropping in voice-over-ip. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 703–715, 2017.

[28] Sanjeev Dhull, Sandeep Arya, and OP Sahu. Comparison of time-delay estimation techniques in acoustic environment. *International Journal of Computer Applications*, 8(9):29–31, 2010.

[29] Song Fang, Ian Markwood, Yao Liu, Shangqing Zhao, Zhuo Lu, and Haojin Zhu. No training hurdles: Fast training-agnostic attacks to infer your typing. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1747–1760, 2018.

[30] Tyler Giallanza, Travis Siems, Elena Smith, Erik Gabrielsen, Ian Johnson, Mitchell A Thornton, and Eric C Larson. Keyboard snooping from mobile phone arrays with mixed convolutional and recurrent neural networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–22, 2019.

[31] Stacey Gray. Always on: privacy implications of microphone-enabled devices. In *Future of privacy forum*, pages 1–10, 2016.

[32] Fabian Höflinger, Rui Zhang, Joachim Hoppe, Amir Bannoura, Leonhard M Reindl, Johannes Wendeberg, Manuel Bührer, and Christian Schindelhauer. Acoustic self-calibrating system for indoor smartphone tracking (assist). In *2012 international conference on indoor positioning and indoor navigation (IPIN)*, pages 1–9. IEEE, 2012.

[33] Tyler Kaczmarek, Ercan Ozturk, and Gene Tsudik. Thermanator: Thermal residue-based post factum attacks on keyboard data entry. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 586–593, 2019.

[34] Luka Kraljević, Mladen Russo, Maja Stella, and Marjan Sikora. Free-field tdoa-aoa sound source localization using three soundfield microphones. *IEEE Access*, 8:87749–87761, 2020.

[35] Kristine M Larson, Andria Bilich, and Penina Axelrad. Improving the precision of high-rate gps. *Journal of Geophysical Research: Solid Earth*, 112(B5), 2007.

[36] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–31, 2018.

[37] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 99–112, 2012.

[38] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of experimental social psychology*, 49(4):764–766, 2013.

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

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

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

[42] John V Monaco. Sok: Keylogging side channels. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 211–228. IEEE, 2018.

[43] David Slater, Scott Novotney, Jessica Moore, Sean Morgan, and Scott Tenaglia. Robust keystroke transcription from the acoustic side-channel. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 776–787, 2019.

[44] Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.

[45] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015.

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

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

[48] Li Zhuang, Feng Zhou, and JD Tygar. Keyboard acoustic emanations revisited. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 373–382, 2005.

Appendix

Implementation, Benchmark, and Dataset. We use a computer with two Xeon E5-2683(v3) CPUs and 32-GB RAM. We use Matlab parallel pools to run multiple threads for offline computation. We have made our dataset, benchmark results, and code available (https://github.com/auditoryeye/auditoryeyesight).

Supplementary Figures. Fig. 25 shows the direct conventional localization results of 594 keystrokes on the Apple Magic keyboard from 0.5 m without our signal processing and multi-round approach. Fig. 28 shows the 2D final-round localization results of experiments on the Apple Magic keyboard (Sec. 4). Fig. 26 and Fig. 27 illustrate the first and second-round results. Fig. 31 shows the 2D final-round localization results of experiments on the Razor Blackwidow Mechanical keyboard (Sec. 4). Fig. 29 and Fig. 30 illustrate the first and second-round results.

Fig. 32 shows outlier errors (up to 10 to 20 ms) and significant entire-keyboard-scale overlapping in the time delay measurements of NLOS laptop-based attacks using conventional methods [28, 39, 46].

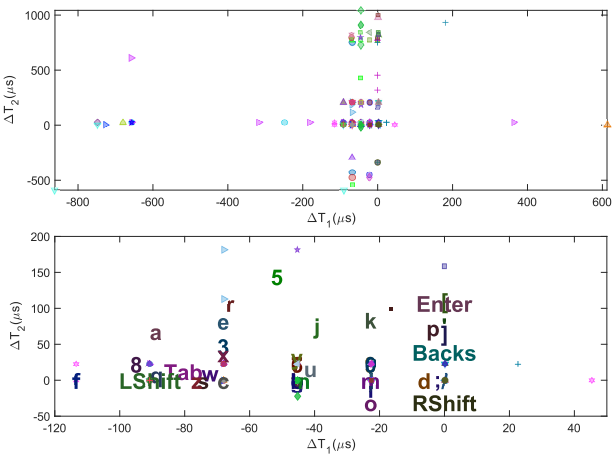

Figure 25: Direct localization results of 594 keystrokes on the Apple Magic keyboard from 0.5 m. The typed keys are not distinguishable using conventional methods.

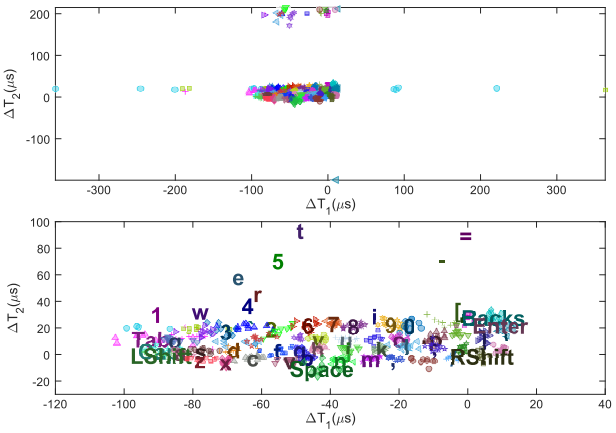

Figure 26: Our initial-Round localization results of 594 keystrokes on the Apple Magic keyboard from 0.5 m.

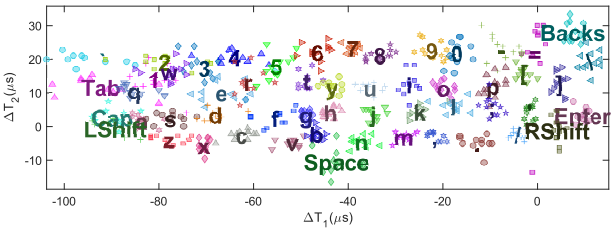

Figure 27: Our second-round localization results of 594 keystrokes on the Apple Magic keyboard from 0.5 m.

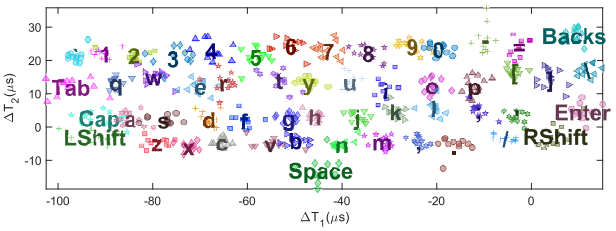

Figure 28: Our final-round localization results of 594 keystrokes on the Apple Magic keyboard from 0.5 m.

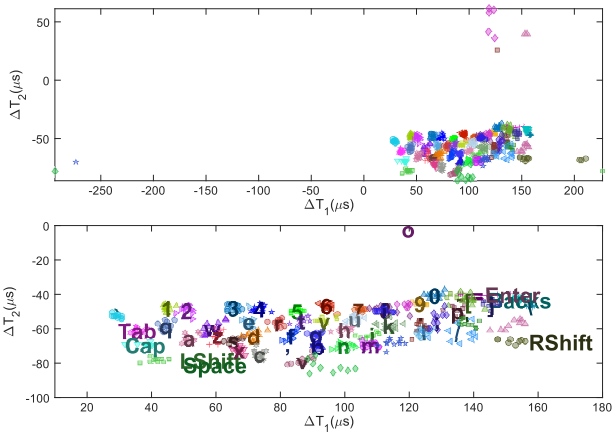

Figure 29: Initial-Round localization results of 595 keystrokes on the Razer Blackwidow mechanical keyboard.

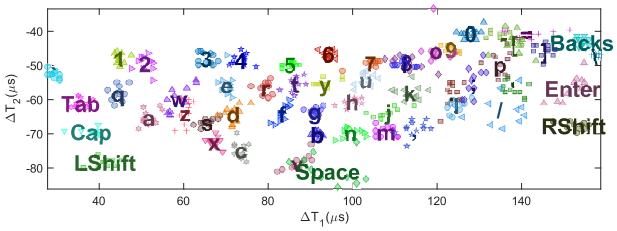

Figure 30: Second-round localization results of 595 keystrokes on the Razer Blackwidow mechanical keyboard.

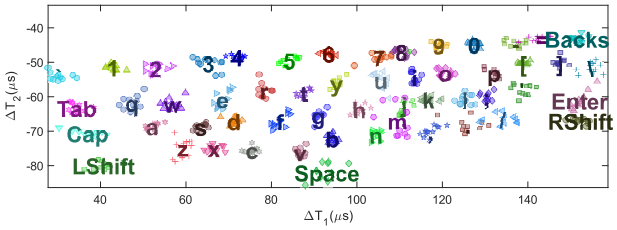

Figure 31: Final-round localization results of 595 keystrokes on the Razer Blackwidow mechanical keyboard.

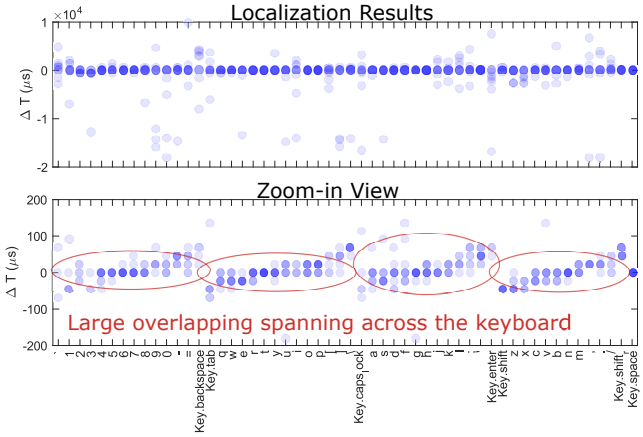

Figure 32: Results of our NLOS laptop-based attacks with conventional methods [28, 39, 46]. The results show outlier errors (up to 10 to 20 milliseconds) and significant entire-keyboard-scale overlapping.