Acoustic Side-Channel Attacks on a Computer Mouse

Conti, Mauro; Duroyon, Marin; Orazi, Gabriele; Tsudik, Gene

DOI
10.1007/978-3-031-64171-8_3

Publication date
2024

Document Version
Final published version

Published in
Detection of Intrusions and Malware, and Vulnerability Assessment - 21st International Conference, DIMVA 2024, Proceedings

Acoustic Side-Channel Attacks on a Computer Mouse

Mauro Conti[1,2], Marin Duroyon[2], Gabriele Orazi[1,4](✉), and Gene Tsudik[3]

[1] University of Padua, Padua, Italy
mauro.conti@unipd.it, gabriele.orazi@phd.unipd.it
[2] Delft University of Technology, Delft, The Netherlands
[3] University of California, Irvine, USA
gts@ics.uci.edu
[4] FDM Business Services, Milan, Italy

Abstract. Acoustic Side-Channel Attacks (ASCAs) extract sensitive information by using audio emitted from a computing devices and their peripherals. Attacks targeting keyboards are popular and have been explored in the literature. However, similar attacks targeting other human-interface peripherals, such as computer mice, are under-explored. To this end, this paper considers security leakage via acoustic signals emanating from normal mouse usage.

We first confirm feasibility of such attacks by showing a proof-of-concept attack that classifies four mouse movements with 97% accuracy in a controlled environment. We then evolve the attack towards discerning twelve unique mouse movements using a smartphone to record the experiment. Using Machine Learning (ML) techniques, the model is trained on an experiment with six participants to be generalizable and discern among twelve movements with 94% accuracy. In addition, we experiment with an attack that detects a user action of closing a full-screen window on a laptop. Achieving an accuracy of 91%, this experiment highlights exploiting audio leakage from computer mouse movements in a realistic scenario.

Keywords: Cybersecurity · Human Interface Devices · Computer Mouse · Acoustic Signals · Acoustic Side-channel · Input Devices · Audio Leakage · Machine Learning · Mouse Movement

1 Introduction

In general, Side-Channel Attacks (SCAs) exploit unintended leaks, such as patterns of power consumption [12,14], electromagnetic emissions [19], visual information (e.g., from screen content or flickering), or audio (e.g., noise from speakers, CPU or disk) from computers or their peripherals to extract sensitive information. Of those, ASCAs typically focus on Human Interface Devices (HIDs) – mechanical devices that produce sound when used by humans. A notable example keyboard acoustic emanations. Characteristic clicking sounds of a keyboard,

F. Maggi et al. (Eds.): DIMVA 2024, LNCS 14828, pp. 44–63, 2024.
https://doi.org/10.1007/978-3-031-64171-8_3

generally considered innocuous, can be analyzed to infer user keystrokes. Such attacks occur in the presence of an attacker-placed (or attacker-compromised) microphone near the target device. While considerable research has been conducted on keyboard-based attacks [3], ASCAs on mouse movements remain relatively unexplored.

Since computer mice are a very popular means of interaction with laptop and desktop computers, this work seems to understand potential security vulnerabilities associated with ASCAs. Our objective is to evaluate the correlation between acoustic signals and mouse movements thus assessing the feasibility of mouse-based SCAs. We begin formulating three research questions:

RQ1. *Can mouse sounds leak its motion path and/or activity?*
RQ2. *If so, with what accuracy?*
RQ3. *In which real-world scenarios do mouse-based ASCAs pose a security risk?*

This paper is organized as follows: Sect. 2 explores related work. Next, Sect. 3 details our research methodology for the experiments. Then, Sect. 4 describes the core experiments, followed by Sect. 5, which discusses security implications of the ASCA. Section 7 concludes the paper and outlines some directions for future works.

2 Related Work

SCAs exploit unintended information leakage from computing devices or their peripherals. For instance, a pattern power consumption of a USB-powered device can be used to profile its activity [18]. SCA can even be used to infer the structure of a neural network [4]. This section overviews ASCAs – the category of SCA that subsumes our work.

Passive ASCAs operate by interpreting audio emanations. The present work falls into this category since the equipment we use (microphone) captures audio without any interactions with the target computer, its peripherals or the user.

Gupta *et al.* [11] use audio Time Difference of Arrival (TDoA) at two different points (using two microphones on the same device) to infer the location of the sound source. Similar studies [15,22] explore various methods to recover keystrokes from acoustic emanations. Balagani *et al.* [2] present a novel ASCA on ATM PIN entry, called the PinDrop attack. It involves two steps: (1) an acoustic profile is created for each key on the target PIN pad, and (2) the attacker records audio emitted by each pressed key during PIN entry and compares these recordings to the acoustic profiles to identify the keys pressed. The resulting classifier is tested on a dataset comprising ten samples for all 26 English alphabet keys, achieving 90.61% accuracy.

Cecconello *et al.* [3] present an attack on popular Voice-over-IP (VoIP) software (Skype) that captures and transmits all acoustic emanations, including keyboard sounds. This attack can lead to leakage of sensitive information, such as passwords, that a victim user might type during a Skype call. Given some knowledge of the victim's typing style and the keyboard model, an attacker can achieve a top-5 accuracy of 91.7% in learning a random key pressed by the victim. We note that [3] is relevant to our work since the keyboard is typically used in tandem with the mouse and our attack scenario is comparable to that in [3].

PoKeMon [9] is a new keystroke monitoring method for smartphones using Mel-Frequency Cepstral Coefficient (MFCC) [21]. Nandakumar *et al.* [16] propose an innovative finger tracking technique by transforming a smartphone into an active sonar system. The system transmits inaudible sounds in the 18–20 kilohertz (kHz) range and tracks the echoes of the finger with its microphones. The system achieves 2-D finger tracking accuracy of 8 millimeters (mm) at 169 frames/sec with a smartphone prototype.

Cheng *et al.* [6] use device speakers to emit inaudible acoustic signals, which are then reflected off the user's fingers and recorded by the smartphone microphones. Similarly, Orthogonal Frequency-Division Multiplexing (OFDM) [8] sounds emitted from device's speakers, while the microphones on the same device are then used to record the echos of these sounds. Also, [6] demonstrates that the number of unlock patterns an attacker must try until a successful authentication can be reduced by up to 70% using this ASCA.

The two results most relevant to our work are Synesthesia [10] and Behavicker [5]. In the former, Genkin *et al.* examine how ASCAs can leak screen content via audio emanations, discovering that LCD screens emit content-dependent audio signals which can be captured by nearby microphones. Experiments conducted using both built-in and webcam microphones reveal acoustic leakage from audio recordings and video-conferencing, e.g., Google Hangouts. The authors determine that the power consumption of the monitor's circuits changes depending on the screen content, causing internal components to vibrate and emit sounds. The authors iteratively simulate the acoustic leakage during hundreds of key presses on the on-screen keyboard, which lead to a 100% accuracy rate in detecting key presses. In their attempt at text extraction they achieve an accuracy rate between 88% to 98% for most individual characters. When extracting words from the screen, the correct word is identified in the top five most probable words in 72% of the cases.

The goal of Behavicker (Chen *et al.* [5]) is to determine user activities from keyboard and mouse clicking and scrolling events. Sounds produced by keyboard typing or mouse clicking reveal information about users' behavior. Actions, such as browsing a news website or playing a video game, produce distinct keyboard and mouse usage patterns, each with its own unique acoustic footprint. Behavicker identifies six basic interaction events with a 88.3% accuracy and differentiates between seven computer-usage activities with a 82.7% accuracy. It utilizes two functional modules: Acoustic-based Interaction Event Recognition and Computer-Usage Recognition. The first uses signal processing and ML to recognize interaction events, while the second employs hierarchical classifiers via time-series analysis to distinguish between types of activities.

Research Gap: Our review of related work reveals an appreciable research gap with respect to the analysis of audio leakage from mouse activities. However, mouse usage remains widespread and, similar to keyboards, mice produce a litany of sounds. Therefore, security of mouse movements deserves and needs to be studied. This is the key motivation for our work.

3 Methodology

Our work involves multiple phases based on research questions posed in Sect. 1. The initial phase is an investigation of mouse-based acoustic leakage. The second phase is the refinement of our predictive models. The third and final phase validates these models in real-world scenarios by recording and analyzing audio from mouse activity in more complex and realistic settings, such as an external mouse used on a laptop.

We do not assume that all mouse activities reflect typical user behavior. Moreover, since this research is exploratory, the datasets of samples consisted of one participant. Since collecting these datasets take several days and do not always provide fruitful results, we conducted initial tests with a single sample size. Thus, while we initially ignored plausibility of mouse movements and sample sizes, we considered this later on in experiments exploring real-world security risks.

We emphasize that all experiments aim to infer the direction of the current mouse movement. This does not yield pixel-level precision, partly because doing so would require taking into account the resolution of the monitor, the model and sensitivity of the mouse as well as many other environmental factors.

All ML models are trained on a Dell laptop with Intel(R) Core(TM) i7-12700H 2.30 GHz, and NVIDIA GeForce RTX 3050, with 16 GB of RAM.

3.1 Mouse and Mouse Pad

In this study, the *'mouse pad'* is the mouse movement area, either an exterior mouse pad or the flat surface near the laptop, upon which the latter rests. Initially we experimented with various mouse pads, discovering that, if audio signals are "audible" by the microphone it yields the same experimental results. Similarly, the experimental mouse is a HP X500 mouse, which is a typical commodity office-style model. For the sake of consistency, the same mouse and mouse pad are used for all experiments.

We use a coordinate system on the mouse pad to categorize mouse movements. Directional movements are defined relative to specific points on the pad: 'T' for top, 'B' for bottom, 'L' for left, and 'R' for right. Also, 'M' represents the middle along the y-axis, while 'C' indicates the center along the x-axis. Therefore, a movement labelled 'TL → BR' means a diagonal trajectory from the top-left to the bottom-right corner of the pad.

3.2 Audio Recording Methods

Each experiment uses different recording methods based on the objective. Key parameters of audio recordings, such as sample rate, bit depth, channels, and file format, play a crucial role in recorded data. Sample rate, typically measured in kHz, defines the number of samples captured per second, where higher rates allow for a more accurate representation of the original sound [7]. Moreover, according to [7], bit depth determines the resolution of the amplitude of each audio sample. This influences the noise level of the recording. The number of channels affects the spatial representation of the sound: mono contains only one waveform, while stereo contains two.

Our experiments use the Waveform Audio File Format (WAV), due to being uncompressed, thus ensuring high fidelity of recordings [7]. We record in mono for experiments utilizing a single microphone since the sound comes from a single source, switching to a stereo configuration in scenarios where a second microphone is used. Bit depth selected for each experiment varies from 16 to 32 bits. Sample rate ranges from 44.1 kHz to 48 kHz in the smartphone range.

In most of our experiments, the microphone is directly connected to the computer. To record with a second microphone, using a smartphone, we use an Android application 'AudioRec' (available in the Google PlayStore[1]).

4 Acoustic Analysis of Mouse Movements

Recall that our goal is to find a correlation between audio signals and mouse movements. While existing literature studied the inference of clicking and scrolling events via acoustic signals, the potential for inferring mouse movements from these emissions remains unexplored. This section addresses the central research question:

> *Can the sounds generated by the movements of a computer mouse be used to infer its trajectory*?

We initially conduct experiments using a single microphone, focusing on identifying and categorizing distinct movement patterns. Next, we integrate a second microphone to further expand the potential of the attack while maintaining a viable attack model.

4.1 Single Microphone Analysis

Audio Capture via Processing Software. In the first phase, we assess whether an acoustic leakage model could distinguish between four basic directional movements of a mouse: up, down, left, and right. Specifically, the experiment is to determine the feasibility of detecting mouse movements along four directions using acoustic signals. A microphone is positioned on the left side of a right-handed mouse pad. The mouse is then moved repetitively along the 'X' and 'Y' axes. The movements are captured using experimental Java Processing Sketch software designed to record these events. As seen in Fig. 1, the starting point (in green) triggers the beginning of the recording process, while the

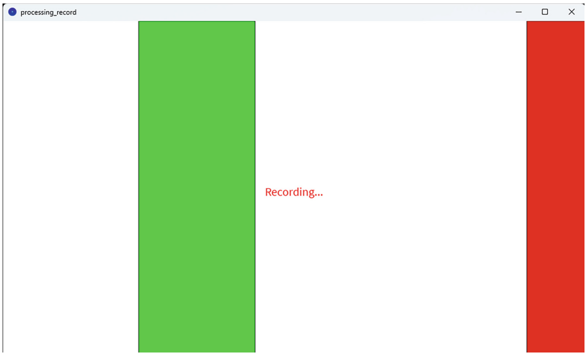


Fig. 1. Recording phase of Left to Right movement. The user has to move the cursor from the green area to the red one. (Color figure online)

red-colored end-point stops it. As a result, the Java Processing Sketch returns acoustic representations of mouse movements in various directions.

We collected 6,000 samples for each direction, yielding a total of 24,000 samples. We note that acoustic signal data-frames, due to the characteristics of the recording procedure, are not of a uniform size. Samples are pre-processed and cleaned for a Convolutional Neural Network (CNN) [17] ML model. We process the waveform of noises to obtain MFCCs using the *Sliding Windows* method with a size of 36 milliseconds (ms), which lead to the best result. At this point, we end up with two lists: (1) one of MFCCs and (2) the other – of encoded labels from 0 to 3 corresponding to each direction. The list is split into training and testing samples, with the testing dataset taking up 35% of the original dataset size. The data are entered in a CNN model, compiled with the default Adam optimizer and categorical cross-entropy as a loss function using ten epochs.

While this model may appear complex, the experiment shows no signs of over-fitting [20], as confirmed by the loss functions in Fig. 2. We also verified lack over-fitting with a second validation dataset split before any computations. After several checks, we deem the results of this experiment as reliable and representative. This model demonstrates a high level of confidence, achieving a 98% classification accuracy rate, with F1-scores ranging between 97% and 99% for each category. These performance metrics are reflected in Table 1.

This study highlights the viability of ASCA. While the current experiment is limited due to simple and constrained movements, it still serves as a proof-of-concept, motivating further research.

Continuous Audio Capture During Mouse Movements. The following experiment we remove the recording software of the previous experiment and use continuous audio capture corresponding to a more realistic setup. This means that the mouse moves on the desktop without predefined patterns and the audio

[1] https://play.google.com/store/apps/details?id=com.audioRec&hl=en_US&gl=US.

Table 1. Classification Report for classifying four movement categories using Processing Record script.

Class	Precision	Recall	F1-Score	Support
Up	0.97	0.97	0.97	2093
Left	0.99	0.99	0.99	2084
Down	0.99	0.99	0.99	2144
Right	0.97	0.97	0.97	2079
Accuracy			0.98	8400

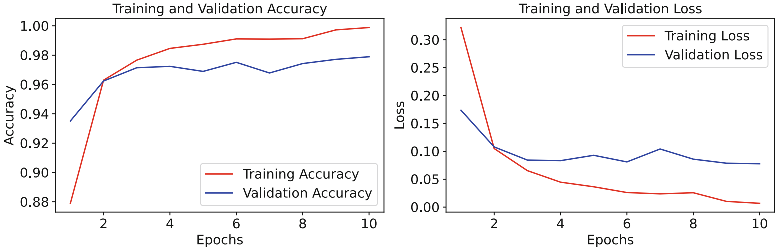


Fig. 2. Accuracy and loss graph for classifying four movement categories using Processing Record script.

is recorded simultaneously. In order to set a direction vector and a movement angle corresponding to movements in an audio frame, we synchronized the background recording with the mouse movements.

The experiment consists of repeating mouse movements in cardinal directions. The mouse coordinates are then converted to direction vectors and associated with acoustic signal frames. The size of each acoustic signal chunk is formed by 8,192 samples with a frame rate of 44.1 kHz, resulting in approximately 186 milliseconds. For classification, the angles are categorized into four (cardinal) directions using the angle representation of the direction vector, meaning that 25% of the circle spatial dimension was associated to a single label. For instance, with a clockwise circle with 0° at the top (north), the angles between −45° and 45° corresponded to the up (or north) direction category.

Computation of MFCCs and their normalization are the only pre-processing steps required and are computed using the same methods as in the previous experiment. Using ten epochs with the same CNN model, we observe a plateau of the validation loss, which leads us to assume that the ML model stops improving. Based on the batch size of 32 data points and the *"sum_over_batch_size"* reduction method during loss calculation, we obtain a loss value of 0.8, which is higher than the average loss for each data point. With this setting, we achieved the overall accuracy of 74%. The classification report can be seen in Table 2, along with the accuracy and loss function graph in Fig. 3.

Table 2. Classification Report for four area classification using background recording.

Class	Precision	Recall	F1-Score	Support
Up	0.69	0.74	0.72	755
Left	0.80	0.68	0.73	852
Down	0.77	0.79	0.78	698
Right	0.72	0.78	0.75	765
Accuracy			0.74	3070

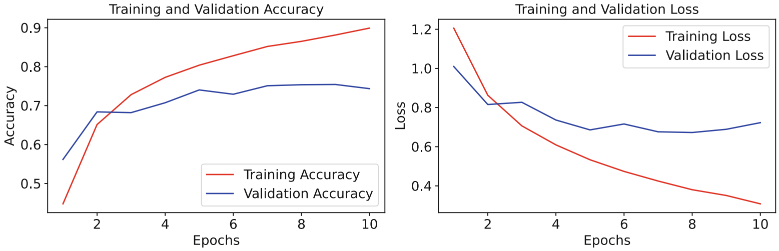


Fig. 3. Accuracy and loss graph for four area classification using background recording.

The accuracy is lower than that obtained with the previous experiment since the recording setup has a greater degree of freedom. With 74% accuracy we can state with confidence that movements in four directions leak enough audio information to determine some user activities. Using a pipeline of a Standard Scaler to normalize, Principal Component Analysis (PCA) and a Support Vector Machine (SVM), we obtain the accuracy of 83%. Consequently, this section answers the leakage model research question: There is indeed an acoustic leakage model for SCA, and it is possible to use audio to infer mouse movements.

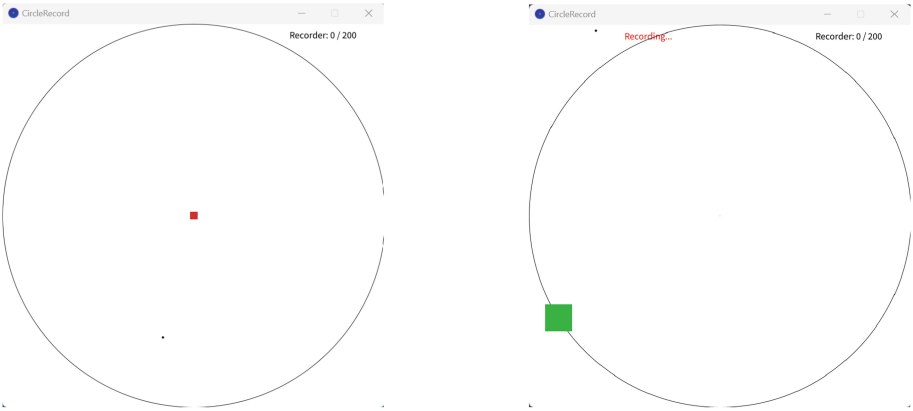
Limitation of Single Microphone Approach. Previous experiments show acoustic leakage that can be split into four categories, However, it does not represent realistic mouse movements. Of course, the idea is to increase the level of granularity in order to understand what angles are discernible with a single microphone and the ML algorithm.

We now evaluate our model to distinguish between eight directions simultaneously, doubling the bins of the previous classifier (classes from 0 to 360° with a step of 45°). The classification report shows that the model has an accuracy of at most 15%. Despite employing the ML framework similar to prior successful models, the algorithm's performance with the eight different categories is sub-optimal.

With different ML models and multiple datasets that cover various environments, the results are still falling short: from Support Vector Regression (SVR) to more complex CNNs. (These experiments are not described due to space limitations).

No model could differentiate between eight movement angle categories. It thus becomes clear that a singular microphone is a limiting factor in capturing similar mouse movements. This prompts us to explore other methods.

Rather than classifying angles into bins, we attempt to apply a regression learning algorithm to predict the angle continuously. The data points are recorded using a similar approach adopted in Sect. 4.1, though using a circular pattern. The software, shown in Fig. 4, initiates the recording when the mouse reaches the middle of the screen (i.e., a red square) and stops it when the green target box is touched by the cursor. The green boxes are always placed on top of the circumference depending on a certain angle with respect to the starting point, i.e., the circle center.



(a) Move cursor towards red center to start the recording.

(b) Move cursor towards green box to stop the recording.

Fig. 4. Circular recording method. (Color figure online)

The CNN model comprises convolutional layers and fully connected layers to map to the target angle prediction. We use a L1 loss function, which simply calculates the mean absolute difference, to approach the target angle as closely as possible. The dataset is composed of ∼21,000 samples of audio MFCCs and mouse movement angles. The results clearly show randomness in the angle predictions. Thus, we believe that predicting angles from 0 to 360 degrees is complicated due to the circular nature of angle measurements. For instance, angles close to 0 degrees and 360 degrees are nearly identical in orientation but are numerically distant. This discontinuity, also known as "Angle Periodicity" [13], can complicate regression models. To address the issue, we conduct a new experiment where angles are represented using their cosine and sine values [13]. However, the average angle difference between the model output and the real angle is 88.79°, which is close to the random angle choice.

To rule out the possibility that the size of the dataset compromises the outcome of the regression, we test data augmentation by adding white noise. However, the model clearly shows signs of over-fitting on randomized augmented data.

The series of regression experiments conducted using a single microphone yield insights into the possibility of accurately predicting continuous mouse movement angles. However, a single microphone is insufficient to regress the angle of mouse movements.

4.2 Analyzing Proximity Through Sound Amplitude Variations

The methodology for analyzing acoustic signals requires a new approach. To this end, we examine the amplitude of sound waves to infer the proximity of sound-generating movements to a microphone. Sounds produced nearer the microphone register with higher amplitudes than those originating at a greater distance. This variance in amplitude could provide a metric for determining the relative distance of the sound source along an axis.

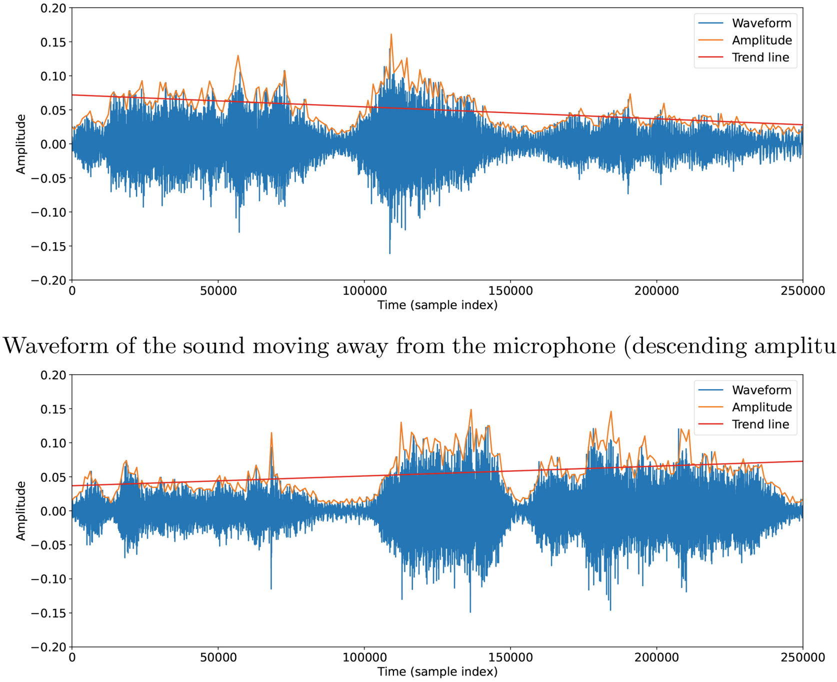
The test involves a controlled series of bidirectional movements along the microphone's directional orientation axis with ten samples. During this process, we segment the audio recordings into two categories: from left to right to represent the mouse trajectory away from the microphone and from right to left to denote its approach towards the microphone. Next, we plot the audio amplitude over time. To do so, we segment the sound into discrete windows and calculate the amplitude, which are then be compared and plotted to visualize waveform differences in time. Afterwards, we apply a linear regression analysis to amplitude data points for a better graphical interpretation. This facilitates a general trend which reveals whether there is a consistent increase or decrease in amplitude. The slope of the regression line serves as the indicator of this relationship.

To highlight the outcomes, we incorporate three distinct auditory markers, consisting of three circles done with the mouse at the start and at the end of the recording. This modification aims to extend the time frame in which the microphone captures the audio.

The experimental data in Figs. 5a and 5b illustrate the subtle, yet consistent, trends in sound amplitude corresponding to the mouse movement near the microphone. We obtain a negative slope angle value of $-0.0000099°$ decrement in amplitude when the mouse moves away from the microphone, and a positive slope angle value of $0.0000068°$ in the opposite direction. Although minimal, the differences are consistent across different tests. These outcomes demonstrate the possibility to visually determine the direction of a mouse movement, validating the hypothesis that a distinctive mouse movement along a single axis can be accurately inferred.

4.3 Dual-Microphone Approach for Two-Dimensional Amplitude Analysis

We now introduce a second microphone. This is achieved using two microphones of a smartphone to provide a two-dimensional measurement plane, as shown in Fig. 6.



(a) Waveform of the sound moving away from the microphone (descending amplitude).

(b) Waveform of the sound moving towards the microphone (ascending amplitude).

Fig. 5. Audio waveforms for the mouse moving towards and away from the microphone.

The recording movement pattern involves traversing the length of the mouse pad, moving from top to bottom and back, and then horizontally from a location near the phone to the far edge and back. Figure 7 shows the resulting graph of two out of four cardinal directions. Each graph shows a plot of the two microphone amplitudes and the amplitude difference. Also, regression lines for the microphone amplitudes are plotted to show general trends. As can be seen in Fig. 7a, representing moves up the mouse pad, a criss-cross effect is evident, indicating one microphone "hearing" louder than the other one at certain times. Furthermore, Fig. 7b shows a negative slope trend in the regression line when moving away from the microphone. Even though graphs are not included in Fig. 7 for the sake of simplicity, we observed coherent behaviour while moving the mouse towards or down with respect to the smartphone.

Since the experiment yield visual representations that clearly distinguish various directional movements, we extend it by attempting to distinguish among ten movement patterns. We record and segment approximately 250 samples for each directional shift using Audacity[2]. The ten categories are a combination of

[2] https://www.audacityteam.org/.

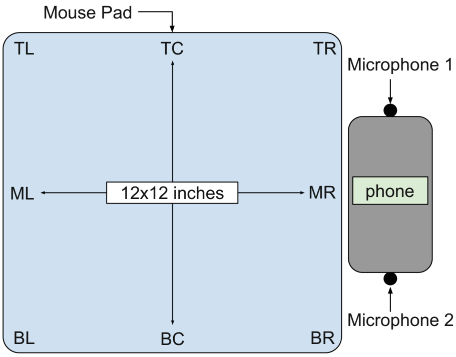


Fig. 6. Dual-Microphone experimental setup.

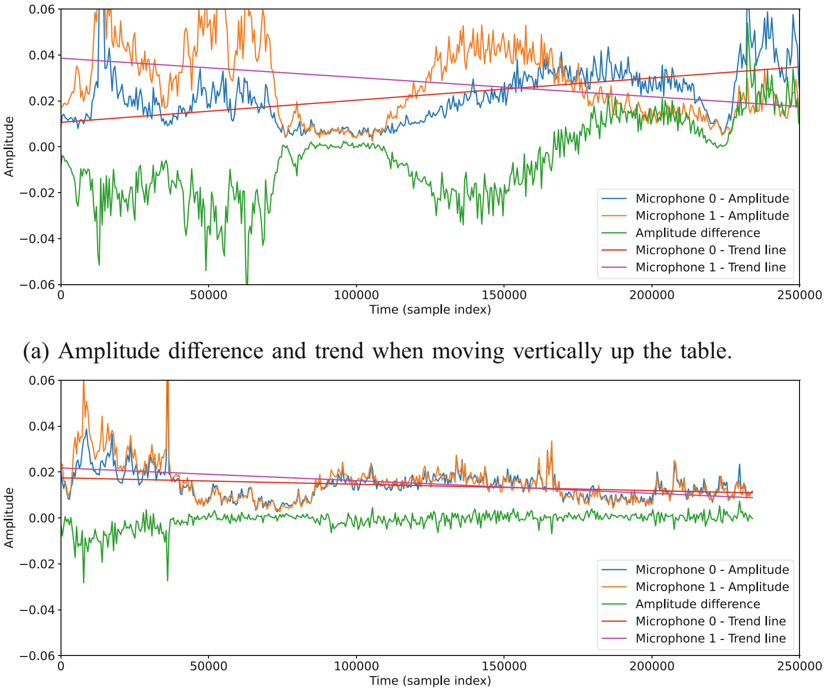
movements with specific start and ending points of the mouse pad, where each point is defined by a vertical and horizontal location. These points are shown in Fig. 6.

Table 3. Classification report on ten movements with dual-microphone setup.

Class	Precision	Recall	F1-Score	Support
BC→TC	0.91	0.94	0.92	52
BL→BR	1.00	1.00	1.00	50
BR→BL	1.00	0.96	0.98	50
BR→TR	1.00	0.97	0.99	39
ML→MR	0.83	0.88	0.85	49
MR→ML	0.88	0.86	0.87	49
TC→BC	0.98	0.98	0.98	52
TL→TR	0.98	1.00	0.99	49
TR→BR	0.97	0.92	0.95	39
TR→TL	1.00	1.00	1.00	49
Accuracy			0.95	478

Using a Random Forest (RF) Classifier with 100 estimators, the model achieves a high accuracy of 95.19%, indicating good performance across various movement categories. The classification report in Table 3 provides further insights into the model's efficacy.

In summary, this experiment demonstrates the ability to infer ten distinct two-dimensional movements, thus exposing the ability to understand the leakage model of a mouse using a smartphone.



(a) Amplitude difference and trend when moving vertically up the table.

(b) Amplitude trend with a negative slope, indicating movement away from the phone.

Fig. 7. Graphs showing amplitude differences and trends based on the mouse's movement.

5 Real-World Implications and Security Risks

We now consider practical implications of our results. A typical attack environment could be a shared office (e.g., a cubicle farm), a cafeteria, or a library. A simple smartphone represents the attack vector. Modern smartphones now feature dual microphones, mainly to reduce ambient noise during calls. The smartphone is a perfect attack vector since it can safely be placed near the victim. It requires no interaction in the recording phase and can remain with its screen off the entire time, so as not to arouse suspicion. With this scenario in mind, we now describe experiments with six participants who volunteered to be recorded while performing predefined mouse movements. Based on this, we examine real-world implications of such attacks.

5.1 Experiment with Other Participants

The next phase involves extending the experiments to include a multiple participants. This helps to determine the generalizability of the model and its applicability to various environments. The new experiment involves six participants performing a series of mouse movements on a mouse pad. This setup is identical to the one described in Sect. 4.3.

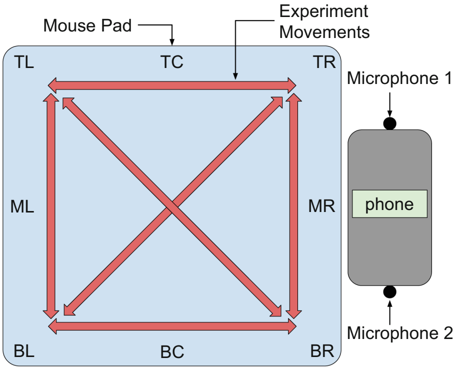


Fig. 8. Mouse pad with mouse movements in the experiment.

Participants are instructed to perform distinct back-and-forth movements along both axes within a 12-by-12 mouse pad placed on an office table. The experiment details specific movement patterns, shown in Fig. 8. Participants are asked to repeat each back-and-forth movement for one to two minutes, resulting in a total experiment time ranging from 6 to 12 minutes. To help with the segmentation during the pre-processing phase, participants are also asked to clap before and after recording each axis to delineate the start and end.

The six participants show variability in their mouse usage methods. For example, some execute the movements rapidly, resulting in a higher frequency of actions, while others move more slowly. In addition, the amount of pressure applied to the mouse differs among participants, producing distinct acoustic signatures. These speed, frequency, and pressure differences are integral to the experiment's design. They provide a diverse dataset that is ideal for testing the generalizability of the ML model, since it reflects a more realistic range of user behavior.

We specifically isolate the segments corresponding to mouse movements using a thresholding technique to distinguish them from background noise, followed by a manual verification process to ensure accuracy. The waveforms from both channels are split into 50 discrete windows and then we calculate their amplitude. Thus, we extract the same number of data points for each movement acoustic signal regardless of different recording methods used by the participants.

Our analysis also involves calculating the difference in amplitudes between the top and the bottom microphone recordings for each segment. Consequently, we generate a "difference amplitude line" of 50 data points representing the differential amplitude for each audio sample. These data points serving as learning values are the 'X' values. The 'Y' values, or target labels, are assigned based on the type of directional movement recorded. This structured dataset, with its distinct X and Y values, allows us to train and test the ML model to classify the directional movements based on acoustic signatures.

This experimental setup is designed to record six types of back-and-forth mouse movements, resulting in 12 distinct directional classes. In the course of the experiments, a total of 5,507 samples are collected. The distribution of these samples across the different classes is between 406 and 506 samples each, with a mean of 456 samples per class.

As shown in Table 4, the classification report evaluates performance of the ML model. The report indicates high accuracy in the model's predictions, with a low rate of false positives. The F1-scores, which combine precision and recall into a single measure, consistently reflect high performance, predominantly in the range of 0.93 to 1.00. This underscores the model's balanced accuracy in both precision and recall dimensions. The overall test accuracy of the model is 96%, indicating the model's effectiveness in accurately classifying the directional movements.

Table 4. Classification Report for a Random Forest Classifier with 100 estimators.

Direction	Precision	Recall	F1-Score	Support
BL→BR	0.98	0.99	0.98	81
BL→TL	0.91	0.98	0.95	97
BL→TR	0.96	0.90	0.93	91
BR→BL	0.99	0.93	0.96	82
BR→TL	0.94	0.99	0.96	91
BR→TR	1.00	0.99	0.99	89
TL→BL	0.92	0.97	0.94	96
TL→BR	0.98	0.93	0.96	92
TL→TR	0.99	1.00	1.00	101
TR→BL	0.93	0.95	0.94	91
TR→BR	0.98	0.93	0.95	90
TR→TL	0.99	0.99	0.99	101
Accuracy			0.96	1102

5.2 Inferring Realistic Mouse Movements

To demonstrate the feasibility of such attacks and help in assessing their potential impact on real-world scenarios, we shift towards a more practical approach. The next experiment aims to determine whether it is possible to detect when a user clicks the 'close' button, typically found at the top right of a Windows laptop screen. While closing a window may not directly reveal sensitive security information, the ability to discern precise user interactions like button clicks might have broader implications for user privacy.

The experiment is structured around two recording sessions of five minutes with a single participant. It is performed on a traditional mouse setting using a Windows laptop and an entry level office mouse pad. In the first session, the participants are instructed to move the cursor from random points on the screen to the top-right corner and click the red 'X', a standard action for closing a window (positive class for the classification). The second session involves recording mouse movements followed by clicks at various random points on the screen – a pattern different from the first recording session (negative class).

We focus on a time window around click events in the analysis phase. This approach allows us to compare acoustic characteristics of mouse movements and clicking sounds associated with both positive and negative classes.

The recordings are first analyzed to determine the clicking events through quantile thresholding methods. To do so, the waveforms are converted to a mono channel to calculate the absolute value of the waveform. This threshold can be modified based on the visual representation and manually adjusted.

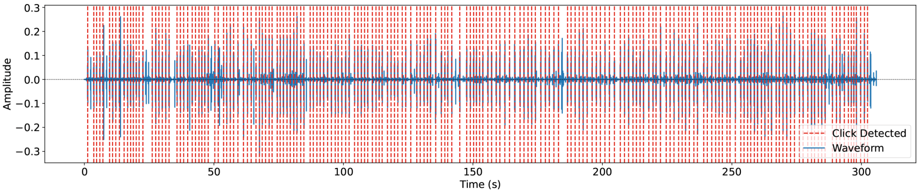


Fig. 9. Detected clicks out of mouse events. (Color figure online)

With these clicking events, the goal is to find the two consecutive peaks corresponding to the mouse press and release events during a mouse click. To do so, we use a minimum and maximum distances between detected peaks to isolate the events that seem to correspond to mouse-clicking events. These distances are again estimated via empirical values, in this case, from 10 ms to 200 ms. Figure 9 shows the waveform with red vertical lines representing detected clicks.

As we iterate through each identified mouse click event, a specific time window surrounding each event is extracted to prepare the data for the ML pipeline. Next, we apply an MFCC transformation to each window. These transformed data points are fed into a ML pipeline utilizing a binary RF Classifier with 50 estimators.

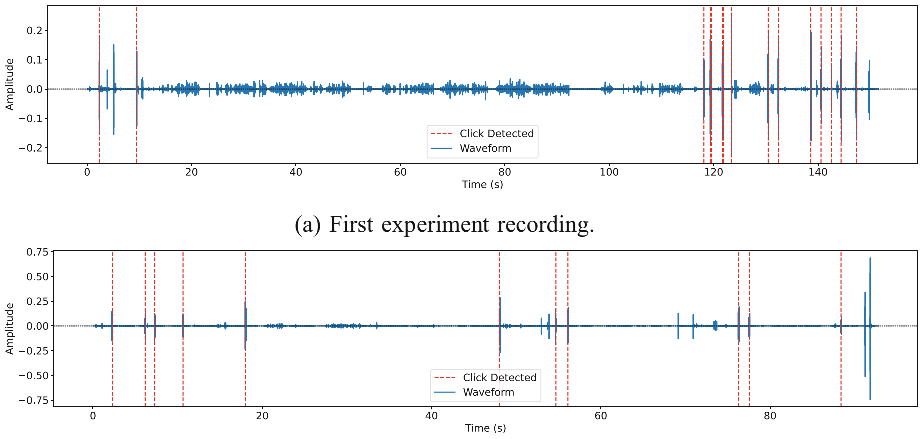
The classification report shown in Table 5 reflects the performance of the binary classification model. The model demonstrates a high level F1-score, with class *'General Click'* reaching 0.92 and class *'Closing Click'* at 0.89. These scores indicate a strong balance between precision and recall across both classes. The overall accuracy of the model stands at 91% for the 136 samples tested, further reinforcing the model's effectiveness.

This experiment successfully demonstrates the ability to distinguish between various mouse movements and their associated clicks. Using controlled recordings, we train a RF Classifier to recognize these specific activities.

Table 5. Classification Report for the Binary Classification Model.

Class	Precision	Recall	F1-Score	Support
General Click	0.89	0.96	0.92	77
Closing Click	0.94	0.85	0.89	59
Accuracy			0.91	136

The goal of the next phase is to extend this approach to more natural computer usage scenarios. We record the authors' typical computer activities, including working on a project in Overleaf, ending with the action of closing a window. These recordings are then processed using the steps previously described. Next, we apply the pre-trained RF Classifier to the isolated mouse click events and infer the type and nature of mouse movements and clicks. This step marks a significant advance in applying our model to real-world scenarios, thus bridging the gap between controlled experimental conditions and everyday computer usage.



(a) First experiment recording.

(b) Second experiment recording.

Fig. 10. Comparison of waveform detection in two iterations of the experiment. (Color figure online)

We experiment by dividing it into two iterations. In the first iteration, the red 'X' is clicked to close the window, together with other two additional clicks. However, due to the thresholding method employed, the final click is not detected, as per Fig. 10a. In the second iteration, the red 'X' to close the window is clicked and followed with two hand claps. This means that the final detected mouse click is the one of interest, as showin in Fig. 10b. Looking at the waveforms, we observe background audio that includes random clicks and typing, creating a consistent hum of ambient noise.

In the first recording, among 15 clicks detected, the next-to-last sample is accurately identified as corresponding to the 'clicking red X' event. Similarly, in the second recording, out of 11 notable events, the model correctly infers the last one as the 'click red X' event. These results demonstrate the trained model's ability to accurately identify window-closing events, pointing at the feasibility of employing such a method in an ASCA.

5.3 Security Risks of ASCAs Based on Mouse Movement Inference

Ourfindings illustrate the feasibility of developing a generalizable model applicable to various users and the capability to accurately identify specific mouse movements in realistic settings. Looking ahead, these insights open the door to potential future threats where ASCAs might be combined with keyboard inference techniques, further reconstructing a victim's computer activities. For instance, one plausible scenario involves targeting users filling out sensitive online forms, such as tax documents. In such cases, the model could be fine-tuned to recognize distinctive patterns associated with clicking specific buttons or entering information into text fields. The attack does not aim to recognize by the number of pixels across which the user moved the moue and whether, at any given time, the cursor is precisely over a specific button. However, pattern recognition in mouse movements could lead to a comparable result. This highlights the importance of further research into ASCAs.

6 Ethical Concerns

Our institutions do not require any formal IRB approval to perform the experiments described above. Nonetheless, all experiments and corresponding evaluations were performed in accordance with the guidelines of the Menlo report [1]. We preemptively informed all voluntary participants about intended usage of their data. We obtained their informed consent before the recording process. We anonymously recorded only audio samples produced by a mouse and explicitly asked all participants to avoid speaking during recording sessions. We used audio samples for research purposes only.

7 Conclusions

This paper explored mouse-based ASCAs. It analyzed whether audio emanations from mouse movements reveal any sensitive information. The initial experiments show that it is possible to differentiate among four basic mouse movements. We then looked into granular measurements in order to predict the direction angles of a moving mouse, which shows some predictive complexities.

Next, we switched to a stereo recording method using two microphones on a smartphone, This allowed us to distinguish between ten two-dimensional movements on a mouse pad.

We then considered potential real-world scenarios where mouse-based ASCAs pose a security risk. To this end, we trained a model – using six participants – that predicts 12 distinct two-dimensional movements. This shows the ability to generalize the model and attack multiple users in one recording environment. Furthermore, we experimented with detection of a specific user actions, such as closing a Window, which demonstrated the efficacy of our approach in a realistic experimental setup. Consequently, we showed that the acoustic leakage model of mouse activity poses a real security risk.

Since this is only the first attempt to experiment with mouse-based ASCAs we identify directions for future work. Applying regression algorithms to stereo acoustic data would be interesting, since all regression tests so far are performed using one microphone. Moreover, the experiment in Sect. 5.2 could be extended by finding other situations where mouse movements reveal sensitive information. Finally, mouse-based ASCAs can be conducted in tandem with keyboard-based ones. The combination of the two might yield even more leakage.

References

1. Bailey, M., et al.: The Menlo report. IEEE Secur. Priv. **10**(2), 71–75 (2012)
2. Balagani, K., et al.: We can hear your PIN drop: an acoustic sidechannel attack on ATM PIN pads. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds.) ESORICS 2022. LNCS, vol. 13554, pp. 633–652. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17140-6_31
3. Cecconello, S., et al.: Skype & type: keyboard eavesdropping in voiceover- IP. ACM Trans. Priv. Secur. (TOPS) **22**(4), 1–34 (2019)
4. Chabanne, H., et al.: Side channel attacks for architecture extraction of neural networks. CAAI Trans. Intell. Technol. **6**(1), 3–16 (2021)
5. Chen, M., et al.: Behavicker: eavesdropping computer-usage activities through acoustic side channel. Wirel. Commun. Mob. Comput. **2022** (2022)
6. Cheng, P., et al.: SonarSnoop: active acoustic side-channel attacks. Int. J. Inf. Secur. **19**, 213–228 (2020)
7. Ciesla, R.: Bits, sample rates, and other fundamentals of digital audio. In: Sound and Music for Games: The Basics of Digital Audio for Video Games, pp. 1–24. Apress, Berkeley (2022)
8. Edfors, O., et al.: An introduction to orthogonal frequency-division multiplexing (1996)
9. Fang, Y., et al.: Eavesdrop with PoKeMon: position free keystroke monitoring using acoustic data. Futur. Gener. Comput. Syst. **87**, 704–711 (2018)
10. Genkin, D., Pattani, M., Schuster, R., Tromer, E.: Synesthesia: detecting screen content via remote acoustic side channels. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 853–869. IEEE (2019)
11. Gupta, H., Sural, S., Atluri, V., Vaidya, J.: Deciphering text from touchscreen key taps. In: Ranise, S., Swarup, V. (eds.) DBSec 2016. LNCS, vol. 9766, pp. 3–18. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41483-6_1
12. Kogler, A., et al.: Collide+power: leaking inaccessible data with software-based power side channels. In: 32nd USENIX Security Symposium (USENIX Security 2023), pp. 7285–7302 (2023)

13. Li, H., et al.: Deep learning methods for protein torsion angle prediction. BMC Bioinform. **18**(1), 1–13 (2017)
14. Lipp, M., et al.: PLATYPUS: software-based power side-channel attacks on x86. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 355–371. IEEE (2021)
15. Liu, J., et al.: Snooping keystrokes with mm-level audio ranging on a single phone. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, pp. 142–154 (2015)
16. Nandakumar, R., et al.: FingerIO: using active sonar for finegrained finger tracking. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1515–1525 (2016)
17. O'Shea, K., Nash, R.: An introduction to convolutional neural networks. In: arXiv preprint arXiv:1511.08458 (2015)
18. Spolaor, R., et al.: Plug and power: fingerprinting USB powered peripherals via power side-channel. In: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, pp. 1–10. IEEE (2023)
19. Yilmaz, B.B., Prvulovic, M., Zajić, A.: Electromagnetic side channel information leakage created by execution of series of instructions in a computer processor. IEEE Trans. Inf. Forensics Secur. **15**, 776–789 (2019)
20. Ying, X.: An overview of overfitting and its solutions. J. Phys.: Conf. Ser. **1168**, 022022 (2019)
21. Zeng, Y., et al.: Spectrogram based multi-task audio classification. Multimed. Tools Appl. **78**, 3705–3722 (2019)
22. Zhu, T., et al.: Context-free attacks using keyboard acoustic emanations. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 453–464 (2014)