# OptiVibe: Keystroke Inference Attacks Through A New Optical-Vibration Side Channel

Youngtak Cho[1], Sanket Suresh Badgujar[1], Srinivasan Murali[2], Xuhao Xie[1], Ming Li[1]
[1] University of Texas at Arlington, Arlington, TX, USA
Emails: youngtak.cho@mavs.uta.edu, sxb3360@mavs.uta.edu, xxx9206@mavs.uta.edu, ming.li@uta.edu
[2] Texas A&M University–San Antonio, San Antonio, TX, USA
Email: srinivasan.murali@tamusa.edu

***Abstract*—Webcam-based video conferencing and live streaming have become pervasive in daily life. During such sessions, users often multitask by typing messages, notes, or emails while keeping the camera active. Many users believe that their typed information remains private as long as the webcam does not capture the keyboard or hands. This paper reveals that this assumption does not always hold. We present a new attack surface for keystroke snooping in video streaming scenarios. In particular, we introduce OptiVibe, an optical–vibration side-channel attack that infers keystrokes directly from webcam video, even when no part of the victim's body or input device appears in the camera's field of view. Our key insight is that typing-induced mechanical vibrations propagate from the keyboard to the webcam and introduce subtle pixel displacement in the streamed video. OptiVibe exploits this effect to extract vibration signatures from webcam footage. It leverages the rolling-shutter mechanism of CMOS sensors to recover high-frequency features despite low frame rates. It further reconstructs missing information caused by network-induced video quality degradation. Besides, the attack does not need any prior knowledge of the victim user. Comprehensive experiments validate the feasibility of this new attack surface.**

***Index Terms*—Optical-vibration side channel, keystroke inference attack, video calls, live video streaming**

## I. Introduction

Webcam-based video calls and live streaming have become a routine part of daily communication. Users rely on these platforms for work meetings, online classes, and social interactions [1], [2]. During such sessions, it is common for users to multitask, for example by typing messages, notes, or emails while keeping the camera on. In these scenarios, users often assume that typing is safe when their hands, keyboard, screen, and even their body are outside the camera's field of view.

This paper shows that those beliefs do not always hold; keystrokes can be inferred from ordinary webcam video alone, even when the keyboard, hands, and the user themselves do not appear in the video. As long as the webcam is active, typing activity can still leak through an unexpected channel. We investigate a novel optical–vibration side channel that arises during typing. Each keystroke generates mechanical vibrations that propagate through the device structure and eventually reach the webcam. Although these vibrations are invisible to the human eye, they induce subtle camera motion. This motion is captured by the webcam and embedded into video frames as tiny pixel displacements in the background scene.

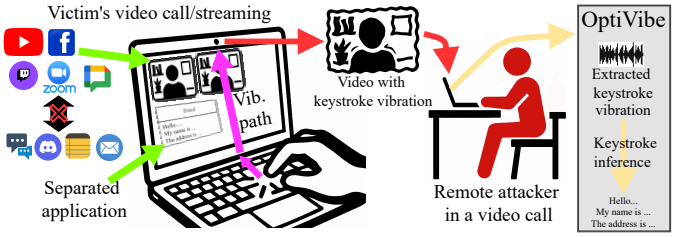



Fig. 1. In a video call, as a victim types, keystroke-induced mechanical vibrations propagate to the victim's webcam and cause subtle pixel displacement in video frames streamed to the attacker's PC. The pixel displacement bears rich information of victim's typing content. The attacker, as a normal participant in the same video call, can record the streamed video and infer victim's keystrokes from video frames.

An adversary who records the streamed video can later analyze these signals to infer the typed text.

While the underlying idea is conceptually simple, constructing a practical attack based on this side channel is faced with several obstacles. First, keystroke-induced vibrations contain important high-frequency components. These components are critical for distinguishing between different keys. However, commodity webcams typically operate at around 30 frames per second, which severely limits the keystroke recognition capability. Second, video calls are often carried over dynamic network conditions. When bandwidth fluctuates, streaming platforms adapt by lowering video resolution. This process would lose fine-grained motion details in the video. Third, the attacker has no prior knowledge of the victim user, including typing behavior, the keyboard type, and device configuration. Pre-training a user-specific model for keystroke inference is infeasible.

To overcome these challenges, we design OptiVibe, an end-to-end keystroke inference attack. OptiVibe extracts vibration-induced motion from webcam video by analyzing subtle pixel displacement in the background scene. It exploits the rolling-shutter mechanism of CMOS webcams to perform temporal up-sampling within each video frame. It thus recovers valuable high-frequency features for keystroke inference. OptiVibe further incorporates depth-aware processing to normalize vibration signals captured from objects at different distances. To cope with information loss due to network dynamics, OptiVibe reconstructs missing high-frequency features via a

novel feature synthesization approach. Finally, it performs keystroke sequence inference using an unsupervised Hidden Markov Model. The model learns directly from the observed video stream and does not rely on any prior knowledge of the victim's typing behavior, device configuration, or keyboard type.

To sum up, this paper makes the following contributions:

- We identify a novel optical–vibration side channel for keystroke inference attacks via webcam video. The attack is feasible even when no body parts or keyboards are visible.
- We design OptiVibe, an end-to-end attack pipeline that addresses practical challenges to launch the attack in real-world video streaming environments.
- We conduct a comprehensive evaluation across diverse devices, keyboards, environments, and video conferencing platforms to validate the feasibility of OptiVibe.

The rest of the paper is organized as follows. Section II presents the threat model. Section III gives necessary background to interpret the proposed optical-vibration side channel. Section IV discusses a unique challenge of OptiVibe under network dynamics and the corresponding solution. It is followed by Section V that describes the end-to-end pipeline design of OptiVibe. The evaluation is discussed in Section VI, which is followed by the discussion of the proposed attack in Section VIII. Section IX concludes the paper.

## II. Threat Model

**Attack Scenarios.** We consider a common video call/conferencing scenario in which the victim participates in an online meeting and shares their webcam feed with other participants. The attacker is a legitimate participant in the same video call. The victim turns on the webcam during the call. To preserve privacy, the victim may intentionally position the camera away from sensitive areas, such as hands or keyboards. The victim's body and head may or may not exist in the camera's field of view, up to the victim's preference. The setting is flexible. During the video session, the victim also types textual content in a separate application. This content may include messages, emails, notes and addresses. We assume that the victim does not intentionally enable embedded background blurring functions provided by the conferencing platform. The video stream therefore preserves natural background details, which is the default configuration in many platforms and widely used in practice.

**Attacker's Capability.** Throughout the session, the attacker continuously records the victim's streamed video on their own machine using widely available commercial recording tools (e.g., [3]). The attacker records only the content that is legitimately visible to a normal participant in the session. The attacker's goal is to perform offline analysis on the recorded footage to recover the typed content from the victim during the video call.

The adversary cannot deploy any external sensing equipment around the victim. The attacker does not compromise the victim's system. They do not hack the webcam, modify the operating system, or interfere with the video conferencing application. No malware is pre-installed on the victim's device. Furthermore, the adversary has no prior knowledge about the victim or the victim's devices. This includes the victim's typing behavior, keyboard model, webcam type, or laptop configuration. The attacker does not possess labeled training data of the victim's keystrokes.

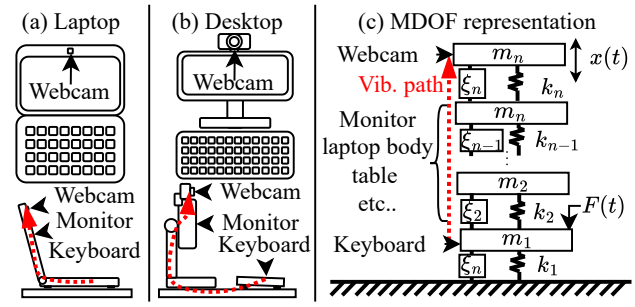



Fig. 2. (a) Laptop configuration, (b) desktop configuration with an external webcam, and (c) an MDOF representation of keystroke-induced vibration from the keyboard to the webcam. Red arrows indicate the vibration path.

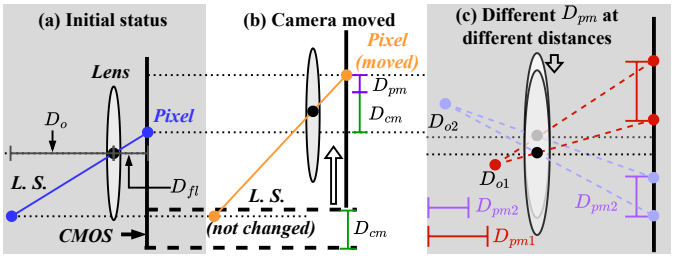



Fig. 3. Illustration of how webcam motion is mapped to pixel displacement under imaging geometry. (a) Initial configuration, where $D_o$ denotes the distance between the light source (L.S.) and the lens, and $D_{fl}$ denotes the focal length. (b) Camera motion induces apparent pixel displacement $D_{pm}$ on the CMOS sensor while the light source remains stationary. (c) For the same camera motion, objects at different depths $D_{o1}$ and $D_{o2}$ exhibit different pixel displacement magnitudes due to depth-dependent geometric scaling.

## III. Background: Optical–Vibration Side Channel in Video Streaming

This section establishes the foundations of the optical–vibration side channel exploited by OptiVibe. We first describe how keystroke-induced mechanical vibrations propagate through the device structure and result in subtle webcam motion that appears as pixel displacement in video frames. We then explain how the rolling-shutter operation of CMOS sensors converts this motion into high-frequency-rich samples. Finally, we present a feasibility study that empirically demonstrates the existence of the proposed new side channel under controlled environments.

### A. A New Optical-Vibration Side Channel

When a user types, keystrokes induce mechanical vibrations that propagate from the keyboard to the webcam through mechanically coupled components (e.g., the device body, hinge, and webcam mount) as shown in Figure 2. This transmission

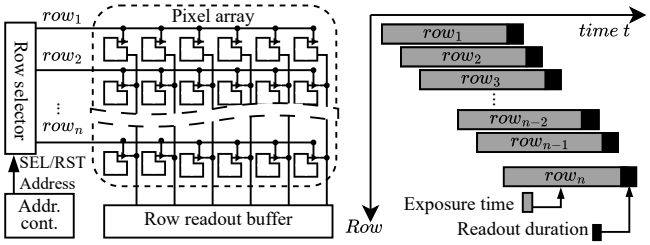



Fig. 4. Left: Pixel array structure of a CMOS sensor; Right: rolling shutter operation in CMOS.

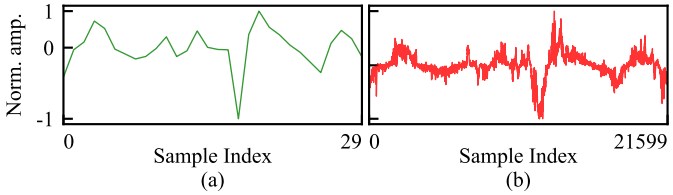



Fig. 5. Extracted one-dimension optical flow (a) before and (b) after rolling-shutter-based temporal up-sampling. One second of video originally sampled at 30 fps (30 samples) is upsampled to an effective 720×30 = 21,600 temporal samples by exploiting row-wise exposure offsets.

path can be modeled as a *Multi-Degree-Of-Freedom (MDOF) mass–spring–damper system* [4].

In this model, the unique mechanical properties of the path (e.g., mass, stiffness, and damping) act as a physical transfer function. While the specific parameters of these intermediate components are unknown to a remote adversary, the MDOF structure ensures that the resulting webcam displacement, denoted by $x(t)$, is a deterministic function of the input force $F(t)$. Consequently, the characteristics of $x(t)$ can serve as an indicator of each specific keystroke.

Under a pinhole camera model, the webcam displacement $x(t)$ translates into pixel-level object displacement, denoted by $D_{pm}(t)$, in the captured video as illustrated in Figure 3. The relation between $x(t)$ and $D_{pm}(t)$ can be expressed as

$$x(t) \simeq -K\, D_{pm}(t). \tag{1}$$

Here $K = D_o/D_{fl}$ represents a scaling factor based on the camera-to-object distance $D_o$ and *focal length* $D_{fl}$. The negative sign indicates that camera motion appears as object motion in the opposite direction.

**To sum up**, the novel optical-vibration side channel for keystroke inference attacks can be described as follows: When the victim presses a key, the keystroke force $F(t)$ is propagated through the device's mechanical MDOF structure to produce a physical webcam displacement $x(t)$. This displacement is then mapped to the pixel displacement $D_{pm}(t)$ in video frames available at the attacker.

### B. Up-sampling the Side Channel via Rolling Shutter

We next describe how to estimate the pixel displacement $D_{pm}(t)$ from webcam video. It is estimated by computing *optical flow* between consecutive video frames using the *calcOpticalFlowFarneback ()* implemented in OpenCV [5]. This function takes two adjacent frames as input and outputs a dense field of per-pixel motion vectors. We aggregate the magnitude of these motion vectors to obtain a one-dimensional temporal signal. This signal approximates the vibration-induced pixel displacement and is denoted as $\hat{D}_{pm}(t)$. In typical online video conferencing or live-streaming settings, the effective frame rate is often at or below 30 fps. Under frame-level sampling, the *Nyquist sampling theorem* limits the maximum recoverable frequency to half the sampling rate, i.e., 15 Hz. As a result, frame-level sampling alone loses most high-frequency vibration components that are critical for distinguishing individual keystrokes.

To overcome this limitation, we exploit the *rolling shutter effect* inherent to CMOS image sensors following prior works [6], [7]. Instead of exposing the entire sensor simultaneously, rolling shutter exposes the pixel array row by row, where each sensor row begins its exposure with a small temporal offset. These row-wise exposure delays introduce additional intra-frame temporal samples that effectively increase the sampling rate beyond the frame rate. Let $\Delta t$ denote the time delay between the exposure of two adjacent sensor rows. The row-wise exposure delay and the corresponding maximum resolvable vibration frequency can be approximated as

$$\Delta t \approx \frac{1}{\text{fps} \times N_{\text{rows}}}, \qquad f_{\max} \approx \frac{1}{2\Delta t}. \tag{2}$$

Here, $N_{\text{rows}}$ is the number of sensor rows in the captured video frame. For example, with a frame rate of 30 fps and 720 sensor rows, $\Delta t \simeq 46.3\,\mu$s. Then the maximum recoverable frequency significantly increases to 10.8 kHz (from 30 Hz). Figure 5 shows the effect of applying rolling shutter to up-sample the one-dimension optical flow signal. The up-sampled signal bears much richer information regarding the keystroke.

### C. Feasibility Study

A feasibility study is conducted to validate the proposed optical–vibration side channel. All experiments are performed under a controlled setup. To eliminate the impact of network dynamics, participants do not initiate real video calls. Instead, they turn on the built-in webcam and record the video stream locally using standard recording software. This allows us to analyze raw webcam output without compression or resolution changes caused by network conditions. Each participant presses the keys `'a'`, `'b'`, and `'y'` 20 times each. All keystrokes are performed under identical environmental conditions. For each recorded video, we extract a one-dimensional vibration signal following the method described in Section III-B. For each keystroke segment, we compute its frequency-domain representation using the Fast Fourier Transform (FFT). The spectra are derived by averaging all the trials for each key.

Figure 6 presents the result. The three keys exhibit distinct frequency patterns. Differences are especially visible in higher frequency bands. Lower frequency components appear more similar across keys. This is because high-frequency components are more sensitive to key location and local structural

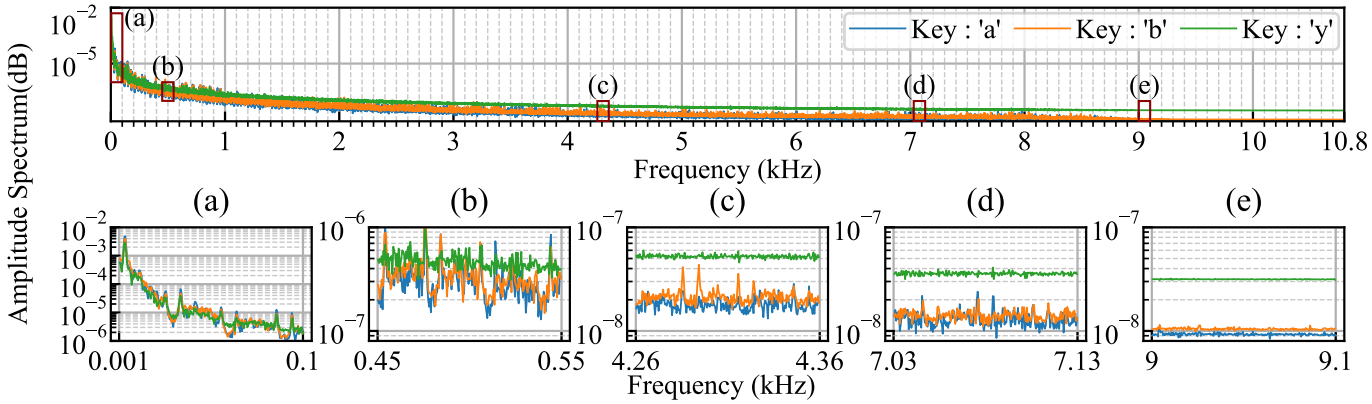



Fig. 6. Top: FFT spectrum of optical flow induced by pressing keys `'a'`, `'b'`, and `'y'`. Bottom: Zoomed-in frequency ranges illustrating distinct high-frequency characteristics across the keys: 450–550 Hz (a), 1–100 Hz (b), 4.26–4.36 kHz (c), 7.03–7.13 kHz (d), and 9.0–9.1 kHz (e).

resonances. Low-frequency components mainly reflect global device motion. As a result, they carry less discriminative information.

**Conclusion.** This study confirms the existence of the proposed optical–vibration side channel. Different keys produce distinguishable vibration patterns in webcam video. The results also validate effectiveness of rolling shutter. Row-wise exposure effectively increases the temporal sampling rate. This allows the recovery of high-frequency vibration features, which contain critical information for keystroke inference.

## IV. Dealing with Video Resolution Changes Under Poor Network Conditions

### A. Impact of Poor Network Conditions

In the previous section, we validate the optical–vibration side channel under ideal conditions, where webcam videos are recorded locally without network-induced distortion. In practice, however, video quality is dynamically adapted in response to network conditions; spatial resolution is often reduced when bandwidth becomes limited. We therefore examine how resolution degradation affects the proposed side channel.

Recall that our approach exploits the rolling-shutter effect to up-sample the optical flow. Rolling-shutter sampling relies on per-row temporal offsets: each image row is exposed at a slightly different time, so as to provide multiple temporal samples within a single frame. When video resolution is reduced, the delivered video undergoes spatial down-sampling, which aggregates multiple sensor rows into fewer output rows. As a result, the number of rows contributing distinct rolling-shutter samples decreases. This reduction directly lowers the effective temporal sampling rate of the rolling-shutter signal and thus reduces the maximum recoverable vibration frequency. This effect follows directly from Eq. (2): as $N_{\text{rows}}$ decreases, $f_{\max}$ decreases proportionally. For example, given a frame rate at 30 fps, as the resolution is reduced from 720p to 480p, the maximum recoverable frequency drops from 10.8 kHz to 7.2 kHz. Figure 7 shows the impact of video resolution changes. It complies with what is discussed above.

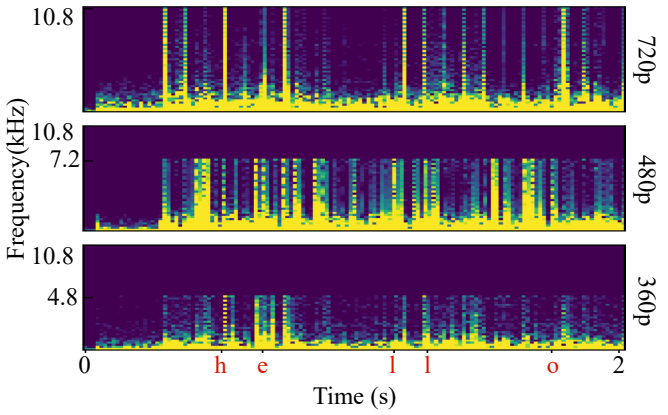



Fig. 7. Spectrograms of the recovered optical flow at different resolutions while typing "hello". The maximum recoverable frequency drops as the resolution degrades. Therefore, useful information to for keystroke inference is lost too.

Since keystroke discrimination relies primarily on these high-frequency components, resolution degradation leads to the loss of information that is critical for accurate keystroke inference.

### B. Reconstructing Missing High-Frequency Features

This part focuses on reconstructing the missing high-frequency features in the optical flow caused by network fluctuations.

Let $\mathbf{F} = [\mathbf{F}_L, \mathbf{F}_H]$ denote the frequency-domain feature vector extracted from a keystroke's optical flow under normal (high-resolution) conditions. Here $\mathbf{F}_L$ and $\mathbf{F}_H$ represent the low- and high-frequency components, respectively. When resolution is degraded, high-frequency components become unavailable, and the observed feature reduces to $\hat{\mathbf{F}} = [\hat{\mathbf{F}}_L, \varnothing]$, with $\mathbf{F}_L$ remaining unchanged.

Our key idea is to reconstruct the missing $\mathbf{F}_H$ of a degraded keystroke by leveraging previously observed keystrokes with similar low-frequency structure that were captured without resolution degradation. Intuitively, if two keystrokes share similar low-frequency vibration characteristics, they are likely

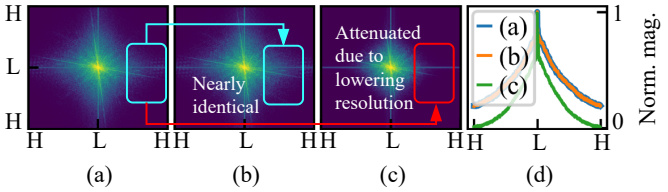



Fig. 8. FFT magnitude spectra of consecutive video frames: (a), (b) from 720p and (c) from a degraded 480p stream. Figure (d) shows the magnitude spectrum of frames. High-frequency energy diminishes after resolution reduction.

to correspond to the same key and therefore share similar high-frequency spectra.

**Step 1: Neighbor Identification in the Low-frequency Components.** Given a degraded keystroke $\hat{\mathbf{F}}$, we first identify a set of neighboring keystrokes whose low-frequency components are similar to $\hat{\mathbf{F}}_L$. Specifically, we compute the *Euclidean distance* $d(\hat{\mathbf{F}}_L, \mathbf{F}_L)$ between $\hat{\mathbf{F}}_L$ and $\mathbf{F}_L$ from previously observed keystrokes. Keystrokes with distances below a threshold are selected as neighbors, forming a set $\mathcal{D}$.

**Step 2: Statistical Modeling of High-frequency Components.** We aggregate the high-frequency components $\mathbf{F}_H$ from the neighbor set $\mathcal{D}$ and estimate their mean and covariance:

$$\boldsymbol{\mu}_H = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{F}\in\mathcal{D}} \mathbf{F}_H, \quad \sigma_H = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{F}\in\mathcal{D}} (\mathbf{F}_H - \boldsymbol{\mu}_H)(\mathbf{F}_H - \boldsymbol{\mu}_H)^\top.$$

These statistics define a multivariate Gaussian distribution that models the high-frequency components of $\mathcal{D}$.

**Step 3: High-frequency Feature Synthesis.** We assume that the missing high-frequency component of the degraded keystroke also follows this distribution. We therefore sample from $\hat{\mathbf{F}}_H \sim \mathcal{N}(\boldsymbol{\mu}_H, \sigma_H)$ to synthesize the missing high-frequency features. The reconstructed keystroke feature is then formed as $[\hat{\mathbf{F}}_L, \hat{\mathbf{F}}_H]$, which restores full-spectrum components.

This reconstruction process does not require labeled data or victim-specific training.

### C. Detecting Resolution Degradation

Now the remaining question is how the attacker determines when resolution degradation occurs. Under our threat model, the attacker does not have access to video metadata. Detection therefore relies entirely on the observed video frames.

The key observation is that changes in frame resolution produce a distinct signature in the frequency domain. When resolution is reduced, high-frequency components diminish due to spatial down-sampling as shown in Figure 8. This effect appears as a sudden and sustained loss of high-frequency energy in the frame spectra. This behavior differs fundamentally from spectral changes caused by scene motion. Motion-induced variations affect a broad range of frequencies. It typically appears as gradual or transient changes across the spectrum, rather than an abrupt suppression confined to high-frequency bands. To capture this effect, we analyze raw

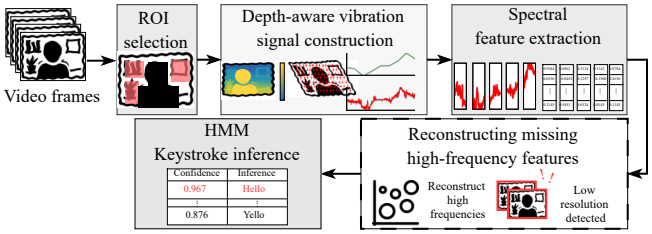



Fig. 9. The design overview of OptiVibe.


Fig. 10. Two example regions considered during ROI selection. The left image shows the video frame. (a) A selected region. (b) Edge pixels extracted within the selected region in (a). (c) A region rejected. (d) Edge pixels extracted within the rejected region in (c).

video frames using FFT and compute the magnitude spectrum for each frame. By monitoring changes in high-frequency energy across consecutive frames, we can identify resolution degradation events without accessing video metadata.

## V. End-to-End Design of OptiVibe

### A. Overview of OptiVibe

We now present the end-to-end design of OptiVibe, as illustrated in Figure 9. The system takes raw webcam video as input. It first selects regions of interest (ROI) from video frames and extracts vibration-induced motion signals using depth-aware optical flow analysis. These signals are then segmented to identify individual keystroke events and transformed into spectral features. To address quality degradation caused by dynamic network conditions, missing high-frequency components are reconstructed. Finally, the reconstructed features are fed into a Hidden Markov Model (HMM) to recover the keystroke sequence. We describe each step in detail below.

### B. Technical Design for Each Module

**ROI Selection.** We focus exclusively on the background scene for vibration extraction, as it provides stable visual cues to extract pixel displacement. Foreground elements related to the victim, such as the body or face, may introduce independent motion that interferes with vibration signals. We therefore remove them from each frame. Google MediaPipe [8] is employed to isolate background regions.

Not all background regions provide reliable measurements of camera motion. Texture-poor, reflective, or blurry surfaces tend to produce noisy optical flow estimates. We therefore select textured and geometrically stable background regions, called ROI regions, for vibration extraction. For each candidate region, we compute its *edge density*, defined as the fraction of pixels classified as edges by a Canny detector from OpenCV [9]. Regions with low edge density lack sufficient

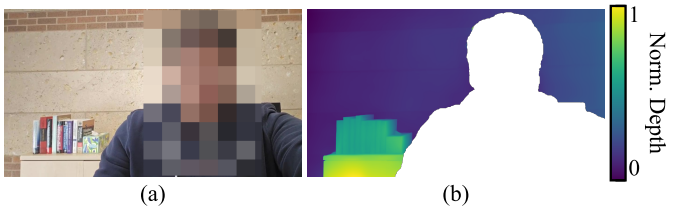



Fig. 11. Illustration of depth estimation: (a) original image and (b) the corresponding relative depth map.

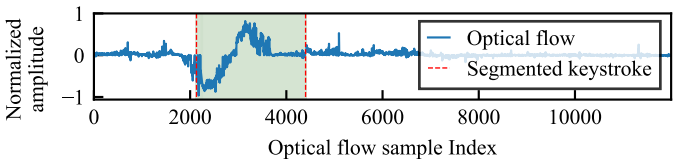



Fig. 12. An example of keystroke segmentation from one-dimension optical flow.

texture and are excluded, as they yield unreliable optical flow estimates. Figure 10 illustrates examples of selected and rejected regions.

**Depth-aware Vibration Signal Construction.** The camera motion produces different pixel displacement magnitudes for ROIs at different depths. Consequently, it makes vibration signals extracted from near and far regions not directly comparable. To compensate for this effect, we estimate a per-pixel relative depth map from a single RGB frame using a monocular depth estimator. We then normalize pixel displacement by the estimated inverse depth. This step mitigates depth-dependent amplitude distortion up to a constant scale factor. We then compute optical flow between consecutive frames and extract displacement magnitudes from the selected ROIs. Leveraging the rolling-shutter mechanism described in Section III, vertically aligned pixels are treated as temporally ordered samples within each frame. For each frame, the normalized displacement values are averaged row by row to form a per-frame vibration profile. Finally, concatenating these profiles across frames yields a continuous one-dimensional time-domain signal in which keystrokes appear as impulse-like events as shown in Figure 12.

**Spectral Feature Extraction.** As the time-domain signal is continuous, we first separate them based on keystroke events. This is done using a short-term average over long-term average (STA/LTA) algorithm [10], [11] by identifying impulse-like peaks (Figure 12). To extract latent spectral features from each keystroke, we compute Gammatone Frequency Cepstral Coefficients (GFCCs) [12] over each extracted time-domain segment. GFCCs are widely used in auditory analysis [12]. They are capable of capturing fine-grained spectral structure and transient energy patterns across a wide frequency range. This makes them well suited for our case.

**Reconstructing Missing High-frequency Features.** Recall that we discussed in Section IV how to recover high-frequency components in the optical flow under dynamic network conditions. The module is integrated into the OptiVibe pipeline

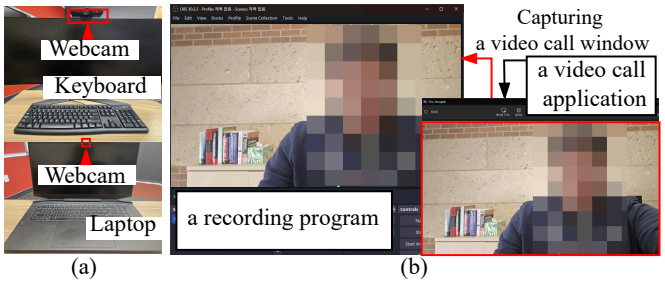



Fig. 13. Experimental setup. (a) Victim-side configurations: external webcam with separate keyboard (top) and integrated laptop (bottom). (b) Adversary-side view.

right after the step of spectral feature extraction. Please refer to Section IV for technical details.

**HMM-based Keystroke Inference.** As the final step of the attack pipeline, we infer the keystroke sequence from the extracted spectral features using a Hidden Markov Model (HMM). Each keystroke is represented by its spectral feature vector and treated as an observation, while the underlying keystroke sequence is modeled using hidden states.

The HMM *transition model* captures dependencies between characters in natural language. Since our attack targets naturally written text such as messages or notes, we adopt a character-level bigram language model as a lightweight linguistic prior. The transition probabilities are estimated offline from 60,000 sentences in the Microsoft News Dataset [13], which provides broad coverage of common character transitions in English text.

The *emission model* describes how spectral features are generated by the hidden states. Because the attacker has no access to labeled keystroke data or prior knowledge of the victim, the emission parameters are learned directly from the observed spectral feature sequence in an unsupervised manner using the Baum–Welch algorithm [14]. This allows the model to adapt to the vibration characteristics observed during the attack without requiring any pretraining or victim-specific information.

Given the derived emission model and the transition model, we apply Viterbi decoding to recover the most likely keystroke sequence.

## VI. Evaluation

### A. Experimental Setup and Metrics

We evaluate video conferencing scenarios where a passive adversary monitors a victim's webcam feed (Fig. 13). We varied devices, lighting, and backgrounds (Table I), with a baseline using a Prometheus XVII laptop under 500 lux illumination [15] via Microsoft Teams. On the adversary side, a remote participant recorded calls using OBS Studio. We recruited 25 participants to type text from the Hillary Clinton email dataset [16] under an IRB-approved protocol (#2025-0547).

We evaluate the following metrics, *Precision*, *Recall*, *Character Error Rate (CER)*, and *Top-k accuracy*. Specifically,

TABLE I
EXPERIMENTAL CONFIGURATIONS USED FOR EVALUATION. **AL**: AMBIENT LIGHT LEVEL, **BG**: BACKGROUND OBJECT OCCUPANCY, **BT**: BACKGROUND ROI TEXTURE, **PL**: VIDEO CALL PLATFORM, **N**: NUMBER OF PARTICIPANTS.

| Category | Configuration |
|---|---|
| Webcams | Nexigo N930W, Logitech Brio 101, TRAUSI TW-05 |
| Laptops | MacBook Pro 2018, Prometheus XVII, ThinkPad T480 |
| Keyboards | Logitech K120, Apple Magic Keyboard, Razer BlackWidow |
| AL | Low ($<150$), Medium (150–449), High ($>=450$) |
| BG | 0%, 50%, 100% occupancy |
| BT | White wall, Office door, Textured wall, Cubicle partition, Office ceiling |
| PL | Teams, Zoom, Google Meet, Discord, Twitch, YouTube |
| N | 25 |

TABLE II
KEYSTROKE INFERENCE PERFORMANCE ACROSS DIFFERENT WEBCAMS. **CAT.** DENOTES THE DEVICE CATEGORY, WHERE **L** INDICATES LAPTOP-INTEGRATED WEBCAMS AND **W** INDICATES EXTERNAL WEBCAMS.

| Cat. | Device | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|---|
| L | MacBook Pro 2018 | 9.9% | 86.3% | 84.3% | 67.0% | 88.8% | 99.3% |
| L | Prometheus XVII | 11.6% | 85.7% | 83.8% | 63.6% | 83.5% | 96.3% |
| L | ThinkPad T480 | 12.9% | 85.2% | 82.9% | 59.4% | 78.8% | 92.2% |
| W | Nexigo N930W | 10.6% | 85.8% | 83.5% | 67.4% | 87.6% | 98.6% |
| W | Logitech Brio 101 | 11.8% | 85.5% | 83.6% | 61.5% | 82.8% | 96.1% |
| W | TRAUSI TW-05 | 14.9% | 84.4% | 82.0% | 52.8% | 71.2% | 82.9% |

*Precision* = (number of correctly inferred keystrokes) / (total number of inferred keystrokes), and *Recall* = (number of correctly inferred keystrokes) / (total number of ground-truth keystrokes). *Character Error Rate (CER)* is defined as the normalized distance between the reconstructed and ground-truth character sequences. *Top-$k$ accuracy* measures whether the ground-truth sequence appears among the Top-$k$ HMM-decoded candidates.

### *B. Attack Performance of OptiVibe*

**Webcam Devices.** Table II shows that keystroke inference performance varies across different webcams. Take Top-10 accuracy as an example: the best-performing devices are the MacBook Pro 2018 (67.0%) and the Nexigo N930W (67.4%), while the TRAUSI TW-05 has the lowest accuracy of 52.8%. The variance is due to the differences in sensor quality and mechanical coupling: webcams with lower noise floors and more stable mounting structures preserve vibration cues more effectively, which leads to high attack success rate. A comparison between laptop-integrated and external webcams shows no consistent advantage for either category. Certain integrated webcams, such as the MacBook Pro, provide strong performance because their camera modules are firmly coupled to the laptop chassis, so that keystroke-induced vibrations to transfer cleanly. Conversely, some external webcams, like the Nexigo N930W, also perform well due to better hardware.

**Impact of Keyboards.** We evaluate keystroke inference performance across three representative keyboards: Logitech K120, Apple Magic Keyboard, and Razer BlackWidow. As shown in Table III, the Logitech K120 and Apple Magic Keyboard achieve similar performance, with low CER and high Top-$k$ accuracy. The performance degrades a bit for Razer BlackWidow. This difference is due to the mechanical design difference in keyboards that impact the keystroke-induced vibration propagation.

TABLE III
AVERAGED KEYSTROKE INFERENCE PERFORMANCE ACROSS DIFFERENT KEYBOARD TYPES

| Keyboard Type | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|
| Logitech K120 | 10.6% | 85.8% | 83.5% | 67.4% | 87.6% | 98.6% |
| Apple Magic Keyboard | 10.6% | 85.7% | 83.9% | 66.3% | 87.2% | 98.5% |
| Razer BlackWidow | 12.6% | 85.2% | 83.2% | 59.7% | 80.7% | 93.5% |

TABLE IV
PARTICIPANT-SPECIFIC KEYSTROKE INFERENCE PERFORMANCE. WPM (WORDS PER MINUTE) DENOTES TYPING SPEED.

| | Participant | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Metric** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| CER | 10.2% | 10.3% | 10.3% | 10.3% | 10.5% | 10.5% | 10.5% | 10.6% | 10.6% |
| Prec. | 85.6% | 85.2% | 88.0% | 85.8% | 86.0% | 86.8% | 84.4% | 84.6% | 86.7% |
| Rec. | 82.5% | 84.5% | 82.1% | 84.5% | 83.3% | 81.7% | 83.3% | 84.9% | 84.2% |
| WPM | 36 | 34 | 36 | 35 | 43 | 26 | 27 | 38 | 43 |
| **Metric** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |
| CER | 10.7% | 10.7% | 10.7% | 10.7% | 10.8% | 10.8% | 10.8% | 10.9% | 10.9% |
| Prec. | 86.6% | 85.9% | 84.3% | 85.4% | 84.8% | 87.2% | 87.4% | 84.9% | 85.9% |
| Rec. | 84.1% | 84.5% | 84.8% | 82.4% | 84.7% | 85.4% | 82.2% | 82.1% | 82.4% |
| WPM | 34 | 37 | 33 | 31 | 36 | 33 | 53 | 36 | 30 |
| **Metric** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | | |
| CER | 11.0% | 11.1% | 12.8% | 13.1% | 13.4% | 13.6% | 13.7% | | |
| Prec. | 86.7% | 84.6% | 56.2% | 52.7% | 51.7% | 47.8% | 47.1% | | |
| Rec. | 84.0% | 83.3% | 49.2% | 47.4% | 43.1% | 43.8% | 48.1% | | |
| WPM | 32 | 60 | 72 | 70 | 92 | 98 | 102 | | |

**Impact of Victims.** Table IV and Figure 14 present participant-specific inference performance. Across the 25 participants, accuracy remains stable, with CER around 10–11% and Precision in the 84–87% range. However, performance drops for the fastest typists. Participants with typing speeds above 70 WPM (e.g., Participants 21–25) exhibit higher CER values and lower Precision and Recall. This pattern indicates that rapid keystrokes distort vibration signatures. This renders keystroke-induced patterns less distinguishable. Despite this degradation, the attack remains feasible for all individuals.

**Video Call/Streaming Platform.** The experiments are performed on various video call and live-streaming platforms, including Google Meet, Microsoft Teams, Zoom, Discord, Twitch, and YouTube. All experiments are under a unified setting to assess the impact of platform-specific video processing on the attack performance. As shown in Table V and Figure 15, inference performance remains highly consistent across platforms, with nearly identical CER and only minor variations in Precision, Recall, and Top-$k$ accuracy. Occa-

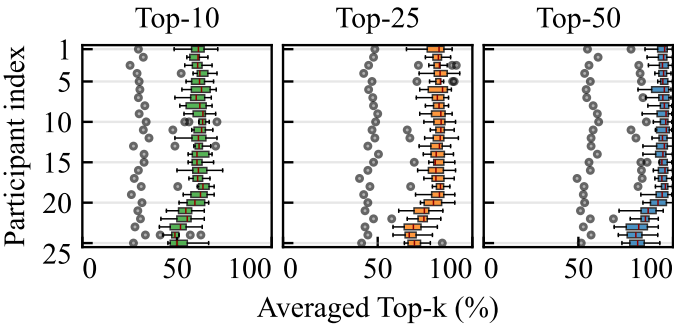



Fig. 14. Top-$k$ inference performance across participants, where each bar represents the Top-10, Top-25, and Top-50 accuracy.

TABLE V
KEYSTROKE INFERENCE PERFORMANCE ACROSS DIFFERENT VIDEO CALL/STREAMING PLATFORMS.

| Platform | CER | Precision | Recall |
|---|---|---|---|
| G.Meet | 11.6% | 85.6% | 83.7% |
| Teams | 11.6% | 85.7% | 83.8% |
| Zoom | 11.8% | 85.4% | 83.5% |
| Discord | 11.6% | 85.0% | 83.7% |
| Twitch | 11.7% | 85.4% | 83.4% |
| YouTube | 11.7% | 85.5% | 83.6% |

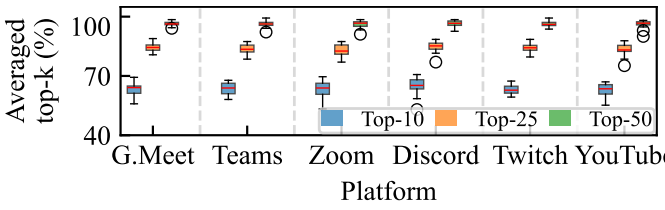



Fig. 15. Top-$k$ keystroke inference accuracy under different video call/streaming platforms.

TABLE VI
KEYSTROKE INFERENCE PERFORMANCE UNDER DIFFERENT NETWORK BANDWIDTHS

| Platform | Bandwidth | Top-$k$ | | |
|---|---|---|---|---|
| | | $k$=10 | $k$=25 | $k$=50 |
| **Discord** | 150 kB/s | 55.5% | 69.5% | 77.8% |
| | 100 kB/s | 49.5% | 59.8% | 65.8% |
| | 50 kB/s | 49.3% | 58.4% | 63.7% |
| **G.Meet** | 150 kB/s | 55.3% | 70.5% | 79.4% |
| | 100 kB/s | 49.3% | 60.5% | 67.0% |
| | 50 kB/s | 49.2% | 59.0% | 64.7% |
| **Teams** | 150 kB/s | 52.0% | 66.1% | 74.7% |
| | 100 kB/s | 46.8% | 57.3% | 63.6% |
| | 50 kB/s | 47.1% | 56.2% | 61.7% |

sional outliers are observed, likely due to transient network conditions, but no systematic platform-dependent effects. This is probably because these platforms employ similar real-time video capture and streaming pipelines.

**Network Conditions.** We further evaluate keystroke inference performance under different network bandwidths. Three bandwidth settings are considered, 150 kB/s, 100 kB/s, and 50 kB/s. The platforms then adjust video quality following their adaptive streaming mechanisms. Table VI summarizes the results. We notice that reduced bandwidth degrades inference accuracy a bit, due to the high-frequency information loss in the observed optical flow. However, such impact is negligible; the attack's success rate is non-negligible still. This is due to the high-frequency feature recovery module developed in the OptiVibe pipeline. The results meet our expectations.

**Background ROI Occupancy.** Table VII summarizes keystroke inference performance under different background ROI occupancy levels. Recall that OptiVibe extracts features from selected background ROIs. Only pixels within these ROIs contribute to the optical flow used for keystroke recovery. As the ROI occupancy increases from 0% to 100%, inference performance improves consistently. Higher ROI occupancy implies that a larger portion of the video frame contains textured background regions suitable for vibration extraction. This provides more samples of camera motion. Consequently, dense ROI conditions achieve substantially higher Top-$k$ accuracy than sparse ROI settings. These results indicate that sufficient background ROI coverage is critical for reliable optical flow estimation and for preserving discriminative vibration patterns across keystrokes.

**Background ROI Textures.** Table VIII shows keystroke inference performance under different background scene textures. Recall that the edge density $\rho$ is defined as the ratio of edge pixels to the total number of pixels in the selected region. This metric varies substantially across texture types and directly affects the quality of optical flow estimation. Texture-poor regions produce higher CER and lower Top-$k$ accuracy. These surfaces lack sufficient visual features, leading to noisy and unstable optical flow that weakens vibration signal extraction. In contrast, texture-rich regions provide abundant visual anchors, which facilitate motion estimation. As a result, they consistently achieve better inference performance.

Among all tested background textures, the textured wall delivers the best performance. The white wall exhibits the lowest performance due to its minimal texture and low edge density. Nevertheless, even under this unfavorable condition, OptiVibe achieves a Top-10 accuracy of 45.7%, which is non-negligible for keystroke inference attacks.

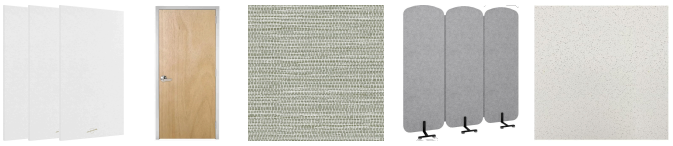
**Impact of Ambient Light Conditions.** Table IX reports keystroke inference performance under different ambient illumination levels. Following OSHA workplace lighting guidelines [17], [18], we simulate office lighting conditions by modeling typical typing intensive office environments at illumination levels exceeding 500 lux, meeting room conditions between 150 and 449 lux, and other indoor spaces at illumination levels below 150 lux.

Low-light conditions impact performance, with CER increasing to 24.5% and sequence reconstruction accuracy dropping (Top-25: 46.4%, Top-50: 55.9%). Increasing illumination to moderate levels yields substantial improvements across all metrics. CER decreases to 12.0%, while Top-25 and Top-50 accuracy increase to 83.3% and 95.2%, respectively. Further increasing illumination provides only marginal gains, with

TABLE VII
KEYSTROKE INFERENCE PERFORMANCE UNDER DIFFERENT BACKGROUND ROI OCCUPANCY LEVELS.

| Background | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|
| 0% | 15.3% | 84.2% | 81.9% | 50.1% | 69.4% | 80.5% |
| 50% | 12.9% | 84.3% | 82.9% | 59.1% | 79.3% | 91.2% |
| 100% | 11.7% | 85.6% | 83.3% | 63.7% | 84.1% | 96.1% |

TABLE VIII
KEYSTROKE INFERENCE PERFORMANCE UNDER DIFFERENT BACKGROUND ROI TYPES (TEXTURES) ($\rho$: THE EDGE DENSITY OF THE ROI). (A)-(E) ILLUSTRATION OF ROI TYPES TESTED IN THE EXPERIMENT.



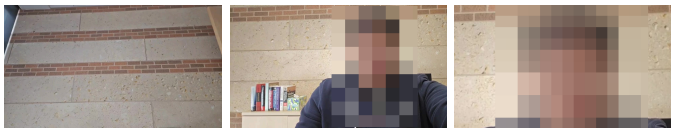
(a) (b) (c) (d) (e)

| ROI types | $\rho$ | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|---|
| (a) White wall | 5% | 16.2% | 82.8% | 81.5% | 45.7% | 64.9% | 76.7% |
| (b) Office door | 8% | 15.8% | 83.6% | 81.8% | 47.8% | 66.6% | 77.8% |
| (c) Textured wall | 47% | 12.2% | 85.9% | 83.7% | 61.4% | 81.4% | 94.7% |
| (d) Cubicle partition | 44% | 12.8% | 85.3% | 82.8% | 60.0% | 79.4% | 92.1% |
| (e) Office ceiling | 37% | 12.6% | 85.0% | 83.6% | 60.4% | 81.1% | 93.3% |

TABLE IX
KEYSTROKE INFERENCE PERFORMANCE UNDER DIFFERENT AMBIENT LIGHT CONDITIONS.

| Ambient Light (lux) | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|
| Low ($<$ 149) | 24.5% | 80.5% | 78.3% | 30.4% | 46.4% | 55.9% |
| Medium (150 – 449) | 12.0% | 85.6% | 83.2% | 63.2% | 83.3% | 95.2% |
| High ($>$ 450) | 11.6% | 85.7% | 83.8% | 63.6% | 83.5% | 96.3% |

TABLE X
KEYSTROKE INFERENCE PERFORMANCE UNDER DIFFERENT BODY VISIBILITY IN THE CAMERA VIEW. (A)-(C) ILLUSTRATION OF DIFFERENT BODY VISIBILITY CASES TESTED IN THE EXPERIMENT.



(a) (b) (c)

| Visibility | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|
| (a) No body | 11.5% | 86.1% | 83.6% | 64.1% | 82.8% | 94.4% |
| (b) Upper body | 11.9% | 85.5% | 83.9% | 62.9% | 81.4% | 92.8% |
| (c) Face only | 12.6% | 83.9% | 82.5% | 61.1% | 78.9% | 91.0% |

performance under high lighting remaining comparable (Top-25: 83.5%, Top-50: 96.3%). These results indicate that ambient lighting primarily impacts inference under insufficient illumination, while typical indoor lighting is sufficient to achieve reasonable reconstruction accuracy.

**Impact of Body Visibility in the Camera View.** We evaluate whether the visibility of the victim's body in the camera view affects keystroke inference performance. We consider three common camera configurations: (i) background-only, where the camera captures only the surrounding scene and no part of the victim; (ii) upper-body visible, where the victim's upper-body appears in the frame; and (iii) face-only, where only the victim's head is visible. Across all three settings, inference performance remains largely consistent. The background-only configuration achieves the best results, followed closely by the upper-body-visible setting. The face-only configuration shows a small degradation across metrics, but the difference is marginal. Overall, these results indicate that body visibility is not a key factor in enabling the attack.

### C. Evaluation of OptiVibe: Deep Dive

**Effectiveness of High Frequency Recovery Module.** Recall that OptiVibe is employed with a high frequency recovery module (Section IV-B) that reconstructs the missing high-frequency features due to poor network conditions. This part is to evaluate the effectiveness of this module. Table XI reports the attack performance under degraded resolutions, and more importantly, the performance with or without the recovery module. Compared to the 720p baseline, resolution degradation substantially increases CER and reduces Top-$k$ accuracy due to the loss of high-frequency information in the optical flow. Applying high-frequency feature reconstruction reduces CER by up to 5.5% at 240p and improves Top-10, Top-25, and Top-50 accuracy across all degraded resolutions. It demonstrates its effectiveness under severe resolution loss.

**Effectiveness of Depth Compensation and ROI Selector Modules.** Table XII reports the ablation results for two key components of OptiVibe: the depth compensation module and the ROI selector module. Disabling either module leads to a clear degradation in attack performance across all metrics. When depth compensation is removed, CER increases from 11.6% to 14.1%, while Top-10 accuracy drops from 63.6% to 53.9%. This decline occurs because vibration-induced pixel displacement scales with object depth; without normalization, signals extracted from objects at different depths become misaligned in amplitude, which weakens feature consistency across keystrokes. Similarly, disabling the ROI selector increases CER to 13.0% and reduces Top-10 accuracy to 59.8%. Without selective region filtering, the vibration signal is diluted by pixels from low-texture or unstable areas that contribute noise but carry little vibration information.

## VII. RELATED WORKS

Side-channel-based keystroke inference attacks have been extensively studied in prior works. They can be broadly classified into four categories: IMU-based [19]–[22], RF-based [23]–[29], audio-based [30]–[36], and visual-based [37]–[43].

**IMU-based attacks** exploit motion sensors in smartphones, wearable devices, or other personal devices to infer keystrokes from body movement [19]–[22]. They represent the earliest effort on side channel keystroke inference attacks. The key

TABLE XI
KEYSTROKE INFERENCE PERFORMANCE ACROSS DIFFERENT VIDEO RESOLUTIONS (BL: BASELINE AT 720P; B: BEFORE RECONSTRUCTION; A: AFTER RECONSTRUCTION).

| Resolution | CER | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|
| 720p (BL) | 11.6% | 63.6% | 83.5% | 96.3% |
| 480p (B) | 16.7% | 43.1% | 62.9% | 75.6% |
| 480p (A) | 14.1% | 49.5% | 69.3% | 81.7% |
| 360p (B) | 21.3% | 28.5% | 48.1% | 61.4% |
| 360p (A) | 17.2% | 39.2% | 59.2% | 71.9% |
| 240p (B) | 26.6% | 27.6% | 48.9% | 60.8% |
| 240p (A) | 20.8% | 40.9% | 62.9% | 74.2% |

TABLE XII
ABLATION STUDY ON KEY SYSTEM COMPONENTS.

| Component | Setting | CER | Prec. | Rec. | Top-10 | Top-25 | Top-50 |
|---|---|---|---|---|---|---|---|
| Depth Comp. | Off | 14.1% | 83.9% | 81.6% | 53.9% | 73.4% | 85.7% |
| | On | 11.6% | 85.4% | 83.4% | 63.6% | 83.5% | 96.3% |
| Selector | Off | 13.0% | 84.2% | 80.9% | 59.8% | 78.9% | 90.9% |
| | On | 11.6% | 85.5% | 83.9% | 63.6% | 83.5% | 96.3% |

idea is that when victim users wear or hold such devices, their typing activity induces subtle movements that can be captured by onboard inertial sensors. Their readings thus reflect the victim's keystrokes. Most IMU-based attacks require pre-installation of malware or privileged applications to access motion sensor data, which limits their practicality in realistic attack settings.

**RF-based attacks** infer keystrokes by exploiting RF signals. Some attacks exploit unintended electromagnetic (EM) or RF emissions generated by end devices or keyboards during user input. The signals are distinctive across the keystrokes [28], [29], [44], [45]. Another class of RF-based attacks exploits wireless communication signals, such as WiFi [23]–[27]. When a user types on a terminal device, finger motion alters the propagation of surrounding RF signals, characterized by, for example, channel state information (CSI). By analyzing the correlation between channel state fluctuations and typing activity, adversaries can infer keystrokes without direct access to the device. One limitation of RF-based attacks is their sensitivity to environmental dynamics. RF signals are easily affected by background changes, such as nearby movement or variations in the victim's body posture. These factors can significantly degrade keystroke inference accuracy.

**Audio-based attacks** exploit sound patterns generated by keystrokes to recover typed input. Such attacks require a sound receiver (i.e., typically microphones) to place near the victim [30], [32]–[34]. As acoustic signals attenuate rapidly with distance, it often requires the receiver to be close to the victim device. It thus significantly restricts their practical deployment. Even though advanced multi-microphone and ultrasonic techniques improve the attack distance [31], [35], [36], the deployment constraint is still there.

**Visual-based attacks** infer keystrokes by analyzing unintended visual cues captured by videos/cameras, such as screen reflections, hand motion, gaze trajectories, or device movement [37]–[43]. Among these works, the most closely related study is [40], which infers keystrokes from video-conferencing footage by analyzing the victim's upper-body movement. That approach relies on the assumption that typing induces observable body movements that correlate with specific key presses. This attack requires the victim's upper body to be captured by the camera. This assumption does not always hold. In many video calls, only the user's face is visible, or the camera faces to the side and captures no part of the body at all. Under such conditions, the attack [40] does not work. OptiVibe removes this restriction entirely. It infers keystrokes by analyzing subtle pixel displacements in the background scene caused by keystroke-induced mechanical vibrations. This optical–vibration side channel does not require the victim to appear in the video. More importantly, it investigates a new optical-vibration side channel that has never been studied before.

## VIII. DISCUSSION

**Generality.** While our evaluation targets QWERTY keyboards, the OptiVibe pipeline is not inherently tied to any specific keyboard layout. The vibration extraction and feature construction stages operate on keystroke-induced mechanical responses captured by the webcam. These stages learn discriminative spectral patterns associated with individual key presses, without assumptions about key positions or layout geometry. Hence, OptiVibe can be readily applied to other layouts, such as AZERTY or numeric keypads, without making any changes.

**Mitigations.** End users can diminish the risk of information leakage caused by OptiVibe by using a virtual background or background blur that replaces the original background scene during the video call. This removes the essential visual cues needed to observe subtle motion caused by typing activity. This functionality is already supported by most modern video conferencing applications and can be easily enabled by users without requiring any additional system changes.

## IX. CONCLUSIONS

This paper presents a novel study demonstrating that keystroke information can be inferred from webcam recordings of video calls or live streams by exploiting an optical-vibration side channel induced by typing activity. We develop an end-to-end keystroke inference system called OptiVibe that leverages typing-induced vibrations propagating from the keyboard to the camera and temporally sampled by CMOS rolling-shutter operation to extract discriminative motion cues, regardless of whether the victim appears in the camera view. We implement OptiVibe and conduct a comprehensive evaluation. The results validate the feasibility of this new attack surface.

## ACKNOWLEDGMENT

The work of Ming Li was supported in part by the National Science Foundation (NSF) under grants CNS-1943509 and CNS-2343618.

## REFERENCES


[1] SearchLogistics, "Zoom user statistics in 2025," https://www.searchlogistics.com/learn/statistics/zoom-user-statistics/, 2025.

[2] Pew Research Center, "How COVID-19 impacted Americans' relationship with technology," https://www.pewresearch.org/, 2025.

[3] OBS Project, "OBS studio," https://obsproject.com/, 2024.

[4] E. Kausel, *Multiple Degree of Freedom Systems*. Cambridge, UK: Cambridge Univ. Press, 2017, pp. 131–250.

[5] MathWorks, "opticalflowfarneback," https://www.mathworks.com/help/vision/ref/opticalflowfarneback.html, 2025.

[6] Y. Long, P. Naghavi, B. Kojusner, K. Butler, S. Rampazzi, and K. Fu, "Side eye: Characterizing the limits of POV acoustic eavesdropping from smartphone cameras with rolling shutters and movable lenses," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, San Francisco, CA, USA, 2023, pp. 1857–1874.

[7] B. Nassi, E. Iluz, O. Cohen, O. Vayner, D. Nassi, B. Zadov, and Y. Elovici, "Video-based cryptanalysis: Extracting cryptographic keys from video footage of a device's power LED captured by standard video cameras," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, San Francisco, CA, USA, 2024, pp. 2422–2440.

[8] Google AI, "MediaPipe image segmenter," https://ai.google.dev/edge/mediapipe/, 2025.

[9] OpenCV Contributors, "OpenCV image processing: Feature detection," https://docs.opencv.org/, 2025.

[10] R. V. Allen, "Automatic earthquake recognition and timing from single traces," *Bull. Seismol. Soc. Amer.*, vol. 68, no. 5, pp. 1521–1532, 1978.

[11] A. Trnkoczy, "Understanding and parameter setting of STA/LTA trigger algorithm," in *New Manual of Seismological Observatory Practice 2 (NMSOP-2)*. Deutsches GeoForschungsZentrum GFZ, 2012, pp. 1–20.

[12] M. Tami, S. Masri, A. Hasasneh, and C. Tadj, "Transformer-based approach to pathology diagnosis using audio spectrogram," *Information*, vol. 15, no. 5, p. 253, 2024.

[13] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu, and M. Zhou, "MIND: A large-scale dataset for news recommendation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Online, 2020, pp. 3597–3606.

[14] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," in *Proc. 3rd Symp. Inequalities*, Los Angeles, CA, USA, 1972, pp. 1–8.

[15] Occupational Safety and Health Administration (OSHA), "etools: Computer workstations - workstation environment," https://www.osha.gov/etools/computer-workstations/workstation-environment, 2024.

[16] WikiLeaks, "Hillary clinton email archive," https://wikileaks.org/clinton-emails/, 2016.

[17] Occupational Safety and Health Administration (OSHA), "29 CFR 1915.82 - lighting," https://www.osha.gov/laws-regs/regulations/standardnumber/1915/1915.82, 2026.

[18] Avetta, "A guide to OSHA workplace lighting requirements," https://www.avetta.com/blog/guide-osha-workplace-lighting-requirements, 2026.

[19] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: Your finger taps have fingerprints," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv. (MobiSys)*, Low Wood Bay, UK, 2012, pp. 323–336.

[20] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: Password inference using accelerometers on smartphones," in *Proc. 12th Workshop Mobile Comput. Syst. Appl. (HotMobile)*, San Diego, CA, USA, 2012, pp. 9:1–9:6.

[21] A. Maiti, O. Armbruster, M. Jadliwala, and J. He, "Smartwatch-based keystroke inference attacks and context-aware protection mechanisms," in *Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, Xi'an, China, 2016, pp. 795–806.

[22] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, 2011, pp. 551–562.

[23] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using WiFi signals," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Paris, France, 2015, pp. 90–102.

[24] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Vienna, Austria, 2016, pp. 1068–1079.

[25] S. Fang, I. Markwood, Y. Liu, S. Zhao, Z. Lu, and H. Zhu, "No training hurdles: Fast training-agnostic attacks to infer your typing," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Toronto, Canada, 2018, pp. 1747–1760.

[26] J. Hu, H. Wang, T. Zheng, J. Hu, Z. Chen, H. Jiang, and J. Luo, "Password-stealing without hacking: Wi-Fi enabled practical keystroke eavesdropping," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Copenhagen, Denmark, 2023, pp. 239–252.

[27] S. Chen, H. Jiang, J. Hu, T. Zheng, M. Wang, Z. Xiao, D. Liu, and J. Luo, "Echoes of fingertip: Unveiling POS terminal passwords through Wi-Fi beamforming feedback," *IEEE Trans. Mobile Comput.*, vol. 24, no. 2, pp. 662–676, 2025.

[28] Z. Zhan, Z. Zhang, S. Liang, F. Yao, and X. Koutsoukos, "Graphics peeping unit: Exploiting EM side-channel information of GPUs to eavesdrop on your neighbors," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, San Francisco, CA, USA, 2022, pp. 1440–1457.

[29] W. Jin, S. Murali, H. Zhu, and M. Li, "Periscope: A keystroke inference attack using human coupled electromagnetic emanations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Seoul, South Korea, 2021, pp. 700–714.

[30] J.-X. Bai, B. Liu, and L. Song, "I know your keyboard input: A robust keystroke eavesdropper based-on acoustic signals," in *Proc. ACM Int. Conf. Multimedia (MM)*, Virtual Event, China, 2021, pp. 1239–1247.

[31] P. Wang, J. Hu, C. Liu, and J. Luo, "RefleXnoop: Passwords snooping on NLoS laptops leveraging screen-induced sound reflection," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Salt Lake City, UT, USA, 2024, pp. 3361–3375.

[32] D. F. Kune and Y. Kim, "Timing attacks on PIN input devices," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, 2010, pp. 678–680.

[33] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *Proc. Annu. Int. Conf. Mob. Comput. Netw. (MobiCom)*, Paris, France, 2015, pp. 142–154.

[34] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Scottsdale, AZ, USA, 2014, pp. 453–464.

[35] L. Lu, J. Yu, Y. Chen, Y. Zhu, X. Xu, G. Xue, and M. Li, "KeyListener: Inferring keystrokes on QWERTY keyboard of touch screen through acoustic signals," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, 2019, pp. 775–783.

[36] Y. Zhao, Y. Zhao, S. Li, H. Han, and L. Xie, "UltraSnoop: Placement-agnostic keystroke snooping via smartphone-based ultrasonic sonar," *ACM Trans. Internet Things*, vol. 4, no. 4, p. 22, 2023.

[37] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, "Beware, your hands reveal your secrets!" in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Scottsdale, AZ, USA, 2014, pp. 904–917.

[38] J. Sun, X. Jin, Y. Chen, J. Zhang, R. Zhang, and Y. Zhang, "VISIBLE: Video-assisted keystroke inference from tablet backside motion," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, 2016.

[39] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "EyeTell: Video-assisted touchscreen keystroke inference from eye movements," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, San Francisco, CA, USA, 2018, pp. 144–160.

[40] M. Sabra, A. Maiti, and M. Jadliwala, "Zoom on the keystrokes: Exploiting video calls for keystroke inference attacks," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, 2021.

[41] Z. Yang, Y. Chen, Z. Sarwar, H. Schwartz, B. Y. Zhao, and H. Zheng, "Towards a general video-based keystroke inference attack," in *Proc. 32nd USENIX Secur. Symp.*, Anaheim, CA, USA, 2023, pp. 141–158.

[42] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Scottsdale, AZ, USA, 2014, pp. 1403–1414.

[43] Y. Wang, W. Cai, T. Gu, and W. Shao, "Your eyes reveal your secrets: An eye movement based password inference on smartphone," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2714–2730, 2020.

[44] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *Proc. 18th USENIX Secur. Symp.*, Montreal, Canada, 2009, pp. 1–16.

[45] L. Wang and B. Yu, "Analysis and measurement on the electromagnetic compromising emanations of computer keyboards," in *Proc. 7th Int. Conf. Comput. Intell. Secur. (CIS)*, Sanya, China, 2011, pp. 640–643.