AcouListener: An Inaudible Acoustic Side-channel Attack on AR/VR Systems

Fengliang He[1], Hong-Ning Dai[1(✉)], Hanyang Guo[2], Xiapu Luo[3], and Jiadi Yu[4]

[1] Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong
hndai@ieee.org
[2] Sun Yat-sen University, Zhuhai, Guangdong, China
[3] The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[4] Shanghai Jiao Tong University, Minhang District, Shanghai, China

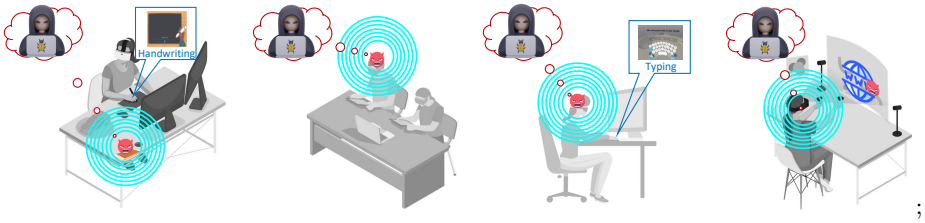
Abstract. Although augmented reality (AR) and virtual reality (VR) systems have garnered extensive attention from both industry and academia, their built-in sensors continuously collect sensitive user data, making them potential targets for malicious attacks. To assess the threat of inaudible acoustic channels in AR/VR, we propose `AcouListener`, a novel side-channel attack that uses inaudible acoustic signals emitted and received by off-the-shelf VR headsets or mobile phones. Variations in the acoustic channel caused by hand movements allow attackers to reconstruct user input (e.g., passwords). `AcouListener` is implemented as a camouflaged mobile app that runs on AR/VR or mobile platforms. We evaluate it across three common VR attack scenarios: (1) inferring victims' unlocking patterns, (2) handwriting patterns and (3) typing words and passwords on virtual keyboards. `AcouListener` achieves an average F1-score of 84%, 95% and 80%, respectively. Furthermore, we present countermeasures against this inaudible acoustic attack.

Keywords: Side-channel Attacks · Augmented Reality · Virtual Reality.

1 Introduction

Recent advances in augmented reality (AR) and virtual reality (VR) technologies provide users with immersive interactions and experiences in seamless physical-virtual worlds. These technologies are proliferating across diverse industrial sectors, such as gaming [30], education [14], smart manufacturing, and healthcare [27]. The commercially available off-the-shelf (COTS) VR headsets (such as the Meta Quest Series, HTC Vive and Bytedance Pico) typically feature Head-Mounted Displays (HMDs) for immersive visuals and controllers for interactive tracking, facilitated by various sensors such as speakers and microphones.

However, while these multi-channel sensors provide users with an immersive experience, they also increase the potential side-channel attack surface for privacy and security on AR/VR systems [30]. For example, an attacker can use cameras (video channels) to monitor a VR user's hand movement, allowing them to infer input on the virtual keyboard (such as login passwords, browsing



(a) Adversary's mobile phone emitting (b) Adversary's HMD emitting (c) Malware of victim's HMD emitting (d) Malicious website emitting

Fig. 1: Four types of side-channel acoustic attacks in AR/VR scenarios.

URLs, etc.) [12,37]. Similarly, other side channels, such as VR device motion sensors [21,29,36], connected Wi-Fi [4], or external IR [25] or mmWave [23], can also compromise user privacy via the tracked controller. These attacks pose serious security risks, including sensitive information leakage, spoofed accounts, stolen identities, and social network leakage. Therefore, it is crucial to comprehensively understand the side-channel threats to AR/VR systems so as to design effective countermeasures to promptly remedy the vulnerabilities.

In this paper, we investigate emerging threats in AR/VR systems by conducting a concealed active side-channel attack through inaudible acoustic signals. We refer to this attack system as `AcouListener`, which can be easily launched using built-in microphones and speakers in COTS VR headsets or mobile phones, requiring minimal hardware and software support. In this attack, a speaker emits inaudible acoustic signals that are imperceptible to human ears. When a victim moves their hands in an AR/VR scenario, the resulting variations in the acoustic channel can be captured by a nearby concealed mobile phone or AR/VR device. Notably, different hand movements produce distinct acoustic responses, each with unique characteristics. By processing the received signals, the adversary can recover gestures for privacy intrusion. Fig. 1 depicts four typical attack scenarios of `AcouListener` (more details given in § 3). For example, when a VR user holds controllers to input information (such as handwriting or virtual keyboard typing), local or remote attackers emitting inaudible sounds can capture variants in the acoustic channel caused by hand movements, thus inferring sensitive information (such as the device's unlock password). However, it is *non-trivial* to implement `AcouListener`, which requires overcoming the following challenges.

Challenge 1: Attack Concealment. Existing methods often rely on external devices (e.g., cameras [12], radar [23], Leap Motion [25]) that are visible and easily detected. Designing a stealthy attack channel using only built-in COTS hardware requires careful design, particularly for broad applicability.

Challenge 2: Capturing Motion Patterns. In-air gestures in AR/VR are unconstrained and motion-induced acoustic disturbances are weak and easily masked by environmental noise, making passive and low-frequency active sensing unreliable. Accurately capturing these subtle channel variations remains technically challenging.

Challenge 3: Lightweight and Deployable Design. While high performance devices (e.g., high-power audio systems) may offer better signal quality, they are typically less accessible or portable. Building a low-power, efficient sensing and inference pipeline that runs on COTS devices without dedicated modifications is essential for practical deployment.

To address these challenges, we design `AcouListener` with three key components (detailed in § 4): (1) *Acoustic Transceiver*. We emit high-frequency acoustic signals (above 18 kHz), which are inaudible to humans, ensuring attack concealment. (2) *Channel Estimator*. We employ the channel impulse response (CIR) method to capture fine-grained motion patterns. Compared to Doppler and Frequency Modulated Continuous Wave (FMCW) techniques, the CIR has demonstrated state-of-the-art performance in gesture recognition [19]. (3) *Hand-Movement Recognizer*. We use well-trained deep convolutional neural networks (CNNs) that require minimal computational and storage resources, enabling lightweight and portable deployment. We develop the whole system as a camouflaged mobile application (app), which can be deployed to mobile phones or VR devices with a compact size (around 70 MB). To further verify the effectiveness of this easily overlooked attack, we conduct extensive experiments in three attack scenarios (detailed in § 5). Results show that `AcouListener` achieves an average F1-score of 84% for unlocking pattern recognition, 95% for handwriting, and 80% for word typing. In addition, we propose countermeasures to mitigate such an attack (discussed in § 7). Moreover, we also provide countermeasures to mitigate such an attack (detailed in § 7). In summary, we highlight the following contributions of this paper.

- **Novel Side-Channel Attack Vector.** We demonstrate that adversaries can actively launch a side-channel attack in AR/VR using built-in microphones and speakers to infer user behaviors. This covert attack is largely imperceptible to victims, as it operates primarily through an inaudible acoustic channel.
- **Customized Attack Design.** We develop an attack system as a camouflaged mobile application that can be seamlessly deployed on various AR/VR devices or nearby mobile phones. By leveraging a lightweight CNN model trained on CIR graphs that capture human gesture patterns, the app can accurately recognize a wide range of victim gestures from the received acoustic signals.
- **Comprehensive Evaluations.** We conduct extensive experiments across three representative attack scenarios commonly found in AR/VR applications. Experimental results confirm the effectiveness of this concealed acoustic side-channel attack, for example, achieving a 96% F1-score in handwritten numbers recognition. In addition, we propose countermeasures to mitigate this attack.

2 Background, Motivation, and Related Work

Acoustic Side-Channel Leakage. Modern AR/VR devices and mobile phones are typically equipped with speakers and microphones to enable immersive human-computer interaction. However, improper permission configurations can intro-

duce a hidden risk of *acoustic side-channel leakage* [11]. Notably, COTS speakers can emit high-frequency acoustic signals (e.g., above 20 kHz) that are inaudible to most adults (typically below 17 kHz [17]), yet highly sensitive to ambient disturbances caused by human movement. When such signals interact with moving objects like hands or fingers, the resulting variations can be captured by microphones and analyzed, enabling accurate gesture recognition using deep CNNs [19,35]. Since the CIR characterizes signal fading, scattering, and delay in the channel [6], the differential CIR (dCIR), defined as $\mathrm{dCIR}(t_1) = \mathrm{CIR}(t_1) - \mathrm{CIR}(t_0)$ (between t_0 and t_1), can be used to model motion-induced channel variations for gesture recognition tasks [19].

Motivation. Most COTS AR/VR devices and mobile phones have microphones and speakers, enabling gesture inference from acoustic channels. These facts naturally raise the question: *Can inaudible acoustic signals be leveraged for a concealed side-channel attack?* To explore this, we investigate (1) how to accurately infer common gestures and (2) which scenarios are vulnerable. To the best of our knowledge, this is the first study to report an *active attack* on the inaudible acoustic side-channel of AR/VR systems to infer private inputs (e.g., unlocking patterns, handwriting, and typing). This attack is distinct from prior mobile-based approaches, as it targets immersive AR/VR settings where users wear HMDs and draw in the air with controllers—scenarios not replicable on standard mobile devices. Compared to earlier acoustic gesture recognition on phones [34,38], `AcouListener` expands the motion range from 10 cm to 30–60 cm, enabling large-scale motion tracking with power-limited devices. We hope this study raises awareness of such hidden risks in the AR/VR community to prevent potential privacy breaches.

Related Work We survey related work on side-channel attacks targeting AR/VR devices through various approaches.

(1) *Acoustic-Based Activity Recognition.* Content inference via acoustics has been studied [7]. Passive methods such as Keylistener [20], Acoustictype [24], and WordRecorder [8] rely on microphones to capture typing or writing sounds, making it inapplicable to in-air AR/VR typing. Active approaches emit sound to track fingers or short-range hand gestures [34,35,38]. However, they mainly focus on mobile interactions and neglect large-scale hand movements in AR/VR. In contrast, `AcouListener` reveals that acoustic channels are also capable of capturing large-scale gestures and exposing sensitive information.

(2) *Side-Channel Attacks in AR/VR.* External cameras [12,26] and virtual avatar tracking [32,37] have been used for privacy inference. Motion, optical, and eye-tracking sensors can also leak private data, such as keystrokes and voice content [29,36]. Other side channels, including performance counters [39] and power traces [18], can reveal user activities. Active-signal-based systems reconstruct hand motions via Leap Motion [25], WiFi [4,1], or mmWave [23]. In contrast, `AcouListener` systematically explores novel inaudible acoustic side-channel risks in AR/VR. Although weak and easily masked by noise, passively captured mechanical button sounds have been used for in-air keystroke inference [22]. In

contrast, `AcouListener` distinguishes itself as an *active* and *inaudible* attack, offering greater robustness to low-frequency noise.

3 Threat Model

Acoustic System. We utilize inaudible sound emitted from AR/VR devices or mobile phones to capture the victim's hand trajectory and infer their inputs. The victim's hand movement may affect the received acoustic signals. The fluctuations of received acoustic signals at the microphones can represent the victim's unique gesture features [16]. Considering signal attenuation, our attack operates within an effective range (detailed in § 6). Moreover, we also assume that the input process is one-time without any input errors or modifications.

Threats. According to different signal sources received by the adversary, we consider two types of side-channel attacks: local attacks and remote attacks. Regarding local attacks, the adversary may sit near the victim (e.g., a public coffee shop) and "hear" the victim's gesture by his/her mobile phone or HMD. In contrast to local attacks, the remote attack can be launched by an infectious malware installed on the victim's HMD or mobile phone, as assumed in many previous studies (e.g., [28,40,41]). Fig. 1 summarizes four types of side-channel acoustic attacks in AR/VR scenarios.

Attack 1. As shown in Fig. 1(a), the adversary deliberately places his mobile phone on the victim's desk. This phone emits and receives inaudible acoustic signals, enabling the adversary to track the victim's hand movements, such as when she unlocks her HMD.

Attack 2. As shown in Fig. 1(b), the adversary sits near the victim in a public setting (e.g., a coffee shop) and uses their own HMD to emit and capture inaudible signals. This allows the adversary to monitor the victim's hand movements, such as typing on a virtual keyboard.

Attack 3. As shown in Fig. 1(c), malware installed on the victim's HMD carries out the attack. The malware uses inaudible acoustic signals to monitor fine-grained hand movements, such as writing or drawing in the air.

Attack 4. As shown in Fig. 1(d), the attack can also be launched by a malicious WebVR website that can access sensors on the HMD for interaction purposes [36]. In this case, the malicious website infects the victim's HMD, emitting and collecting inaudible acoustic sounds to track the victim's hand movements, such as entering a URL.

Attack Scenarios. Adversaries can launch the above four types of attacks (either locally or remotely) in the following three scenarios.

Scenario 1 (Inferring Unlocking Patterns). Many COTS HMDs utilize unlocking patterns for device access [30], e.g., Meta Quest 2 (MQ-2). In AR/VR, users draw the unlocking pattern in the air using a controller, in contrast to touchscreen input on mobile devices [31]. In this proposed side-channel attack, adversaries capture acoustic fluctuations caused by the victim's hand movements via microphones and infer the unlocking pattern by analyzing the resulting dCIR patterns using deep CNNs.

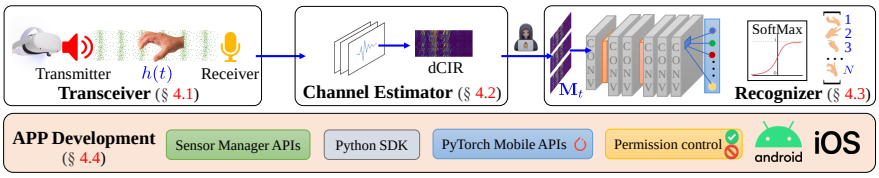


Fig. 2: System overview of AcouListener.

Scenario 2 (Inferring Handwritten Inputs). In AR/VR collaborative applications, such as virtual meetings (e.g., Meta Horizon Workrooms), victims may use an MQ-2 controller to write or draw on a virtual screen. During this process, the resulting acoustic signal variations produce unique dCIR patterns, which can be captured and analyzed (either locally or remotely) by adversaries to infer the written content.

Scenario 3 (Inferring Hand Typing on Keyboards). Typing is another common hand movement. Current VR HMDs provide various input methods [12,36]. For instance, the MQ-2 supports both beam-style and drum-style virtual keyboards. This opens up several attack opportunities: (1) monitoring web activity by capturing inputted URLs, which can be used for profiling or targeted attacks; (2) inferring passwords, including credentials for meetings or games; and (3) recovering typed content such as meeting notes or other sensitive information entered during VR sessions.

4 System Framework

This section details the design of `AcouListener`, as illustrated in Fig. 2.

4.1 Acoustic Transceiver

Our acoustic transceiver consists of two main functions: transmitting the target signal frames and receiving them for subsequent acoustic feature extraction.

Generation. Fig. 3 illustrates five steps for generating the acoustic signal used to detect channel information affected by moving objects (e.g., hand movements via VR controllers). We utilize a 26-bit GSM training sequence, which is widely used in single carrier communication due to its high efficiency in channel estimation and synchronization [19]. Furthermore, a 24-bit zero padding is added to prevent inter-frame interference. Subsequently, the frame is upsampled using replication interpolation, with each element repeated 12 times, resulting in a total of 600 symbols within the signal frame. A low-pass filter with a cut-off frequency of 2 kHz is then applied to eliminate discontinuities, ensuring a smooth signal. Each signal frame takes 12.5 ms to play at a 48 kHz sample rate, allowing for the transmission of 80 frames within 1 second. From these received sequences, 80 dCIR feature graphs can be extracted to characterize the acoustic channel variations associated with object movements. This intensity is sufficient to distinguish between different hand input movements.

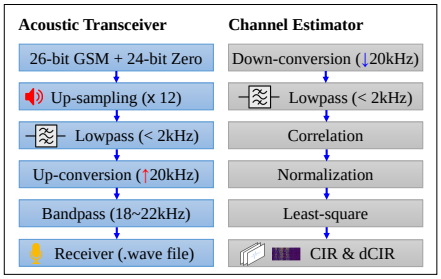
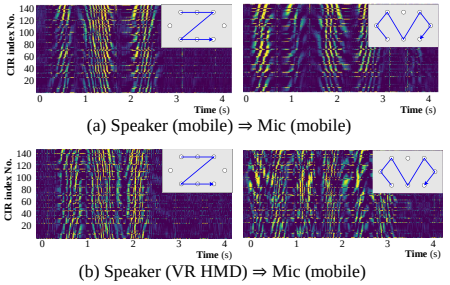


Fig. 3: Audio signals generation, transmission and estimation.



(a) Speaker (mobile) ⇒ Mic (mobile)

(b) Speaker (VR HMD) ⇒ Mic (mobile)

Fig. 4: dCIR patterns under different transceiver settings.

Up-conversion. Speakers integrated into VR devices or mobile phones typically operate at a sampling rate of 48 kHz [17]. According to the Nyquist sampling theorem, the maximum playable frequency can theoretically reach 24 kHz. To enhance the concealment of sound during detection, we up-convert the signal frame to an inaudible band by modulating with $\sqrt{2}\cos(2\pi f_c t)$, where f_c is set to 20 kHz. This process shifts the signal's center frequency to f_c, ensuring it remains within an inaudible frequency range that is imperceptible to adults [17]. Next, we apply an 18-22 kHz band-pass filter (at a 48 kHz sample rate) to the signal to eliminate interference and noise. Finally, we save the sequence S as a Wave file in 16-bit PCM format, preserving high-quality audio data while ensuring compatibility and versatility for playback and processing across various audio applications and devices.

4.2 Channel Estimator

After receiving the reflected signal sequences containing key features of the VR user's hand movements, we segment them into frames and extract channel variations using dCIR graphs.

Demodulation. The signal R is transmitted at a high frequency, with a center frequency of 20 kHz. We down-convert it by modulating with the carrier $\sqrt{2}\cos(2\pi f_c t)$. A low-pass filter with a cutoff frequency of 2 kHz processes the signal. Environmental noises (e.g., voices, music, footsteps, and air-conditioning sounds), typically fall within the low-frequency band [35]. These noises are naturally eliminated through down-conversion and the low-pass filter, enhancing the robustness of our recognition system. To identify the start index and segment the signal into frames containing the 600 training symbols for channel estimation, we calculate the Pearson Correlation Coefficients between the received 1200-bit signal and a 600-bit transmitted signal frame S. The peak of this correlation indicates the start index. The signal is then segmented into target frames, starting from this initial point and cut every 600 data points.

Normalization. The signal received by the microphone is influenced by factors such as the distance between the user's hand and the receiver, obstructions

of the sound source, and the level of environmental noise. To address this issue, the keystroke data is normalized by dividing each data point by the maximum value in the dataset. This process compresses the data to a range between 0 and 1, ensuring uniformity and comparability across the dataset.

CIR Estimation. We employ the least squares (LS) channel estimation method, which offers lower computational complexity, to estimate the CIR of the collected signals reflected by the target object (i.e., the hand). The core idea of LS estimation is to derive the CIR, denoted as H, by solving the equation $R = S * H$, where $*$ represents convolution, and S and R are the transmitted and received signals, respectively. To obtain an accurate H, it is essential to have a sufficiently large S, although this may increase computational costs. Specifically, we define the length of the training sequence as $T = P + L$, where L represents the memory length needed to account for multipath effects and P is the reference length used for channel information calculation. Therefore, the training sequence is represented as $S = \{s_1, s_2, \ldots, s_{L+P}\}$, while the calculated CIR sequence is $H = \{h_1, h_2, \ldots, h_L\}$. The received signal sequence is denoted by $R = \{r_1, r_2, \ldots, r_{L+P}\}$. Based on LS channel estimation, the CIR can be obtained using the following equation:

$$\underbrace{\begin{pmatrix} s_1 & s_2 & \cdots & s_L \\ s_2 & s_3 & \cdots & s_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_P & s_{P+1} & \cdots & s_{P+L-1} \end{pmatrix}}_{\text{Training Matrix } \mathbf{M}_t} \cdot \underbrace{\begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_L \end{pmatrix}}_{\text{CIR } H} = \underbrace{\begin{pmatrix} r_{L+1} \\ r_{L+2} \\ \vdots \\ r_{L+P} \end{pmatrix}}_{\text{Received Sequence } R} . \tag{1}$$

We then obtain the estimated CIR denoted by $\hat{H}$ as follows,

$$\hat{H} = (\mathbf{M}_t^T \mathbf{M_t})^{-1} \mathbf{M}_t^T R. \tag{2}$$

In our system, we set $L = 140$ and $P = 172$ to achieve adequate accuracy while maintaining low computational complexity. Each signal frame produces 140 channel taps, reflecting the channel status over a duration of 12.5 ms, resulting in 80 updates per second. The value of dCIR is calculated by subtracting the CIR value of the previous frame from that of the current frame: $\text{dCIR}(t_1) = \text{CIR}(t_1) - \text{CIR}(t_0)$. This value represents the extent of change in the CIR, characterizing channel fluctuations caused by object motion (i.e., hand movements).

Transceiver Setting. Attacks can be launched using various combinations of acoustic transceivers: (1) a VR device as the speaker and a mobile device as the receiver (microphone); (2) a VR device as both the speaker and receiver; (3) a mobile device as the speaker and a VR device as the receiver; and (4) a mobile device as both the speaker and receiver. We conduct preliminary experiments using the first and fourth combinations of experimental settings. The results are presented in Fig. 4. It is evident that different gesture trajectories exhibit unique patterns on the dCIR graph in each scenario. Each gesture corresponds to a distinct trajectory pattern in terms of shape and distribution, which provides a crucial foundation for gesture recognition. By analyzing these patterns, we

can extract key features to differentiate between various gestures. Furthermore, the results indicate that different transceiver combinations do not affect the unique dCIR patterns generated by hand movements. Therefore, for simplicity, we primarily consider the fourth combination as the basic setting for our system.

4.3 Hand-Movement Recognizer

The dCIR sequences of the acoustic channel are obtained from the sound signals reflected by the hands of the VR user. To identify and analyze hand movements, we employ a CNN, training it with a series of dCIR sequences (images).

CNN-based Recognizer. Due to their exceptional performance, deep CNNs have been widely employed in various computer vision tasks [5,9]. After processing the CIR signals, we obtain dCIR sequences, which can effectively be considered as images. We then train a deep CNNs model on these dCIR images to accurately recognize different types of hand movements.

Data Augmentation. One of the challenges in using CNNs in our system is the limited availability of dCIR data for training. To tackle this challenge, we employ data augmentation. Specifically, we apply various image transformation operations, including random rotations, translations, scaling, cropping, and flipping. By applying these random transformations to the obtained dCIR images, we expand the training dataset, increasing both the diversity and quantity of samples. This strategy enables the model to learn from a more varied dataset during training, thereby enhancing its generalization capability and robustness. The system can effectively handle various factors such as gesture speed, distance, and high-frequency noise by using this enriched data to train the final model.

4.4 Camouflaged App Development

To implement the aforementioned processes on mobile devices (e.g., AR/VR HMDs), we develop a camouflaged app that encapsulates the above functional modules, thus enabling its deployment on mobile or AR/VR devices (e.g., Meta Quest Series). Since speakers require no permissions, this app only requests access to the `MICROPHONE` for the collection of inaudible sound data while this permission can be easily obtained by masquerading as a benign application (e.g., video conferencing software) or injection attacks [13]. The recognizer model is developed by using `PyTorch` and saved as a script model by `torch.jit.trace` so that it can perform without relying on the original Python interpreter. We adopt `pytorch_android` API to load the trained model on the Android platform for dCIR images recognition. While our primary focus is on Android development, a similar approach can be applied to iOS or other platforms.

5 Experiment

We conducted extensive experiments to validate the feasibility of the proposed side-channel attack across different scenarios.

(a) Unlocking (b) Hand writing (c) Hand typing

Fig. 5: Experimental Setup.

5.1 Experimental Setup

Hardware Devices. In our experiments, we selected the Honor X10 and iPhone 12 Pro as mobile platforms, with the MQ-2 serving as the default VR device, currently one of the most popular VR hardware options available [36]. Generally, `AcouListener` can be adapted for use with other AR/VR devices by developing similar mobile applications.

Experiment Design. In the attack scenarios, a volunteer sits in a chair at a desk, wears the MQ-2 HMD, and holds controllers in both hands while performing designated hand movements. Simultaneously, a mobile phone (i.e., Honor X10) emits and receives corresponding inaudible audio signals and captures the volunteers' hand movements, as illustrated in Fig. 5. The speed, amplitude, and distance of the volunteers' inputs are tailored to their usage habits (on average, participants complete scenarios 1 and 2 within 3-5 seconds, and scenario 3 within 6-10 seconds). The experiments were conducted in a public office room (11 m x 15 m), where there was audible noise interference from the environment (approximately 55 dB), but no moving objects.

Data Collection. We collect a total of 9,400 samples across three attack scenarios. We recruited 10 volunteers in the experiments[5]. They vary in age (20-30 years), height (158-185 cm), and gender (70% males and 30% females). Among them, two participants are familiar with VR devices, while the others had no prior exposure. Before the experiments, we provided the volunteers with essential training to ensure they were comfortable with basic operations. Consistent with prior studies [2,33,10], we employed a training-test split, allocating 80% for training and validation, and 20% for testing. Theoretically, a well-trained model with sufficient data can generalize effectively and perform consistently on new users, just as it does on the test set.

Training CNN Models. We developed a mobile CNN model based on MobileNet V2 for gesture classification tasks (refer to Appendix A for more details). This lightweight and portable model delivers excellent performance on mobile devices while maintaining efficiency. After being resized to a specified dimension of 224x224 pixels with 3 color channels, a dCIR image is then normalized using a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225]. To enhance the model's robustness and generalization ability, we apply data augmentation techniques on dCIR images. The model is initialized with a

[5] Ethical approval has been obtained (SCI-COMP-2024-25_002).

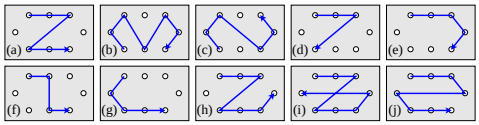


Fig. 6: Top-10 unlocking patterns.

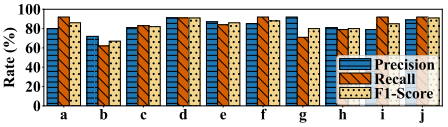


Fig. 7: 10 unlock patterns results.

learning rate of 0.001, a batch size of 32, and is trained for 100 epochs using stochastic gradient descent (SGD) as the optimization algorithm.

The trained model with a size of approximately 70 MB, has been deployed on a mobile device as described in § 4.4. Experimental measurements on the Honor X10 smartphone indicate that the model takes approximately 80 ms to predict a dCIR image, thereby meeting practical application requirements.

5.2 Attack Scenario 1: Unlocking Pattern Inference

Unlocking Action on VR Device. The MQ-2 supports pattern-based unlocking, allowing users to select and connect points on a virtual screen using the controller, as shown in Fig. 6. Each pattern typically includes four to eight points, which can be seen as a direct extension of similar features on mobile phones [31]. During startup or standby, once the controller is paired and activated, a virtual unlocking interface appears. The user moves a cursor using the controller, presses a button to select a point, and continues connecting at least four points. If the drawn pattern matches the preset configuration, the device is successfully unlocked.

Attack Scenario. In this scenario, we assume that the victim, wearing a VR HMD, is seated at a desk attempting to unlock the VR device. Simultaneously, a mobile phone, casually placed on the table, runs a camouflaged (spy) application that continuously sends and receives inaudible sounds. This application utilizes the trained mobile CNN model to recognize the victim's hand gestures. Our objective is to accurately infer the victim's unlock patterns by analyzing the signal changes in the surrounding sound channel.

Experimental Description. As shown in Fig. 6, we invited volunteers to enter 10 different unlocking patterns, which were primarily designed based on the most common unlocking patterns found in the Android system [3]. Each pattern was repeated 50 times, resulting in a total of 1,400 data samples collected.

Attack Results. The result is shown in Fig. 7. The average precision, recall, and F1-score all exceed 84%, indicating an acceptable inference performance for the adversary from the perspective of side-channel attacks. It can be found that pattern `j` has the highest F1-score (91%). The reason is that the lines in the pattern `j` are complex enough and require connecting all eight points, making it significantly different from other patterns observed in the trajectory. While pattern `i` (F1-score: 85%) also involves eight points, the intertwining of its trajectory increases the similarity between different segments.

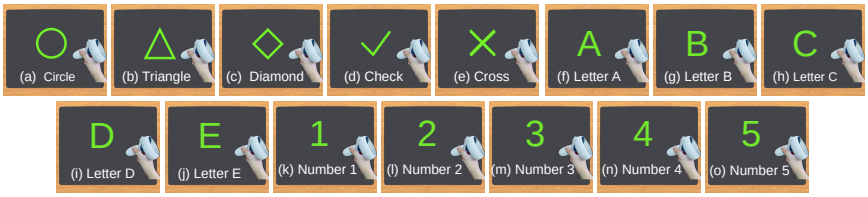
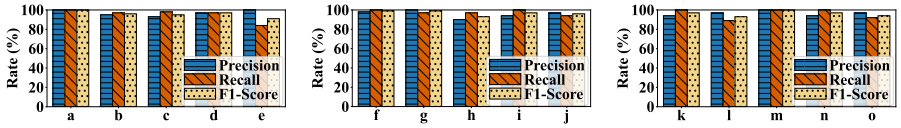


Fig. 8: Handwritten patterns.



(a) Handwritten shapes (b) Handwritten letters (c) Handwritten numbers

Fig. 9: Recognition of 15 handwritten drawings.

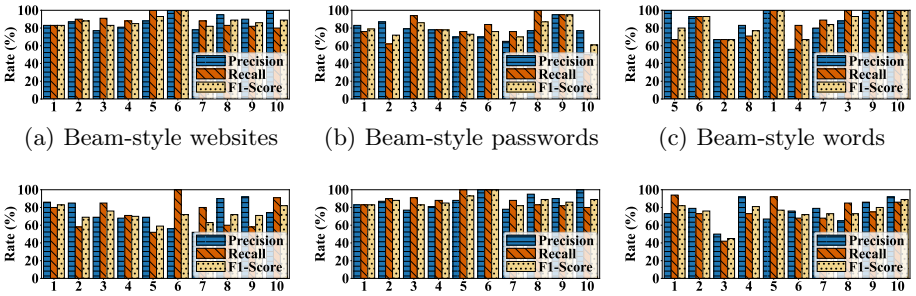
5.3 Attack Scenario 2: Hand-Written Content Inference

Handwriting in Virtual Meetings. In VR video conferencing applications, users engage in immersive communication and collaboration. For example, in Horizon Workrooms, a user can write or draw on a virtual whiteboard. In this scenario, the user first designates an area on the physical desk as the virtual whiteboard, then holds and moves the controller like a pen to write on it.

Attack Scenario. We consider a situation where a victim sits at a desk and participates in video conferences on the Horizon Workrooms platform using the MQ-2 device. During the meeting, the victim writes and draws on the virtual whiteboard. A mobile phone running a camouflaged application secretly analyzes the victim's hand movements by emitting and collecting inaudible sounds. By analyzing the acoustic signals, attackers can track the victim's hand movements and recognize the content being written on the virtual whiteboard.

Experimental Description. Each participant sits at their desk, entering the Horizon Workroom virtual conference room through the MQ-2 HMD. Using the hand controllers as pens, they write a series of shapes, letters, and numbers on the table. Fig. 8 illustrates the main shapes, letters, and numbers evaluated in our experiments. These drawing actions are rendered in the virtual space as inscriptions on the virtual whiteboard. Meanwhile, the mobile device placed next to the table sends and records inaudible sounds to capture variations in the acoustic signals. In total, 2,550 data samples were collected.

Attack Results. As shown in Fig. 9, the experiment yields an average precision, recall, and F1-score of 95%. Handwritten numbers exhibit the highest score (over 96%) due to their distinct shapes, which distinguish them from each other. Furthermore, each type of handwritten pattern achieved an F1-score greater than 90%, demonstrating better recognition performance compared to the recognition of unlock patterns. This may be attributed to the relatively fixed trajectory of handwriting actions, making the patterns easier to capture than unlock patterns.



(a) Beam-style websites (b) Beam-style passwords (c) Beam-style words

(d) Drum-style websites (e) Drum-style passwords (f) Drum-style words

Fig. 10: Recognition of beam-style and drum-style typing.

5.4 Attack Scenario 3: Hand Typing Inference

Besides unlocking patterns and handwriting, typing on virtual keyboards is another common interaction method in VR. We focus on two representative VR typing techniques. *Beam-style Keyboard.* In this method, the user holds a controller that emits a visible virtual beam (e.g., a laser pointer) to select keys on the virtual keyboard by pointing and clicking. *Drum-style Keyboard.* In this method, the controller becomes a drumstick in the virtual environment, used to strike virtual keys in mid-air, resembling the action of drumming but without any physical contact. We consider three types of hand-typing attacks:

(1) Monitoring Web-Surfing Habits. Peeping into MQ-2's keyboard input can reveal the website URLs entered by the user in the browser, thereby allowing for wiretapping their web-surfing habits. When users utilize MQ-2's browser, they must use their hand controller to input various web addresses and search terms on the virtual keyboard. During this process, an adversary can employ inaudible sound to capture the user's hand movements and infer the URLs being entered. Once the adversary successfully obtains this website information, they can deduce the user's browsing preferences, interests, and potentially sensitive data. The misuse of this information could lead to targeted advertising, personal data analysis, or other malicious purposes.

(2) Inferring Passwords. When using the MQ-2, users often need to enter various passwords via the virtual keyboard, such as login credentials or unlocking codes in virtual meetings or games. This input process can present security risks. Attackers can use inaudible sound to capture the user's hand movements and infer the entered password through analysis of these movements. By emitting inaudible sound around the user, attackers can leverage the device's microphone or other sensing techniques to obtain continuous input recognition (CIR) features during the user's hand movements. By analyzing these features, attackers can deduce which keys the user has pressed, thereby guessing the composition and order of the password. This method poses a significant threat to the confidentiality of the user's sensitive information, such as passwords, personal identification numbers (PINs), and other confidential data. If an attacker successfully infers

the user's password, they may exploit this information for unauthorized access, theft of sensitive personal information, or identity fraud.

(3) Inferring Words. Similar to typing URLs and passwords, users may also type on a virtual keyboard using beam-style or drum-style inputs. During this process, confidential information may be exposed to attackers who can use inaudible sound to capture changes in dCIR during the user's hand movements and analyze these changes to infer the content being entered. This attack poses a potential threat to the user's input and privacy. Once attackers infer the user's input, they can monitor the user's activities, obtain sensitive information, or engage in other forms of abuse. For instance, if a user is participating in a video conference using the VR device and types important information such as meeting notes on a virtual keyboard, the attacker could infer the user's input and monitor their behavior during the meeting.

Experimental Description. In this experiment, volunteers sit at their desks and type content using the MQ-2 HMD. Table 2 in Appendix B lists the typing contents, which are divided into three scenarios as mentioned above. (1) For the password inference scenario, we selected the top 10 most commonly used passwords[6]. (2) For the keyword inference scenario, we selected the first word from the names of the 10 most popular VR apps[7]. Users typically only need to enter the first word when downloading or searching for an app, making this word a strong indicator of the app they intend to run. (3) For the web-surfing habits inference scenario, we selected the top 10 most popular websites[8] as test subjects. The layout of the virtual keyboard is set to the default QWERTY configuration. A total of 5,450 data samples were collected.

Attack Results. The results are shown in Fig. 10. Across the three types of attacks, the average F1-score for beam-style typing is 84%, while that for drum-style typing is 79%, resulting in an overall average F1-score of 80%.

(1) In website URL inference, beam-style typing (average F1 88%) demonstrates superior performance compared to drum-style typing (average F1 80%). This is primarily due to the longer lengths of website URLs. When using single-hand beam typing, the method better captures continuous trajectories, allowing for more significant differentiation. Further, the keyboard layout plays a crucial role. For instance, when typing `facebook.com` (average F1 93%) and `amazon.com` (average F1 89%), a volunteer must move back and forth across the keyboard, resulting in salient hand movements and distinct trajectory differences.

(2) In password inference, drum-style typing with an average F1-score of 83%, outperforms beam-style typing with an average F1-score of 78%. This difference may be attributed to the short length and proximity of these simple passwords on the keyboard. Shorter input lengths lead to more similar trajectories in single-handed beam-style typing, while the use of both hands in drum-style typing

[6] `https://techcult.com/most-common-passwords/`
[7] `https://shardeum.org/blog/best-metaverse-platform/`
[8] `https://www.expireddomains.net/alexa-top-websites/`

separates the input into two parts, thereby amplifying the differences between them. This input method enhances the ability to distinguish between different typing contents more effectively.

(3) In word inference, beam-style typing (average F1-score: 87%) demonstrates a higher degree of distinguishability compared to drum-style typing (average F1-score: 81%). One possible reason is that the letters forming these words (such as VR app names) are often distributed across different positions on the keyboard, allowing one-handed beam-style typing to produce significantly more distinguishable trajectories. In summary, the overall accuracy of content recognition is influenced not only by the length of the content but also by its specific distribution on the keyboard.

6 Real-World Impact Factors

We summarize the following factors in real-world deployments based on extensive experimental observations and results (detailed in § 5.1).

Environmental Noise. Environmental noises, such as voices, music, footsteps, and air-conditioning sounds, typically fall within the low-frequency band [35]. `AcouListener` is robust against low-frequency noise because it utilizes only the high-frequency band during transmission and filters out signals below 18 kHz during demodulation (detailed in § 4.2).

Attack Range. The feasibility is experimentally validated within a recognition distance of approximately 1.2 m. Beyond this range, the strength of acoustic signals diminishes rapidly, making them unrecognizable. This limitation is primarily due to the power constraints of portable mobile and AR/VR devices. We propose a future solution for this problem in § 7.2.

Bystanders. We have observed that the movement of bystanders can affect the received acoustic signals, as recognition depends on variations in the acoustic channel caused by moving objects. However, this effect is confined to a limited range (approximately 1.2 m) due to the devices' power limitations. Therefore, we primarily focus on scenarios where their impact is negligible (beyond the attack range), which is typical in home or private settings when using AR/VR.

Inconsistent Position and Posture. `AcouListener` imposes no strict constraints on the relative position or posture of the victim. While distance affects CIR strength, this effect can be mitigated through normalization (§ 4.2). Variations in angle, posture, hand movement speed, and motion range may deform CIR patterns though data augmentation (§ 4.3) enables the model to handle such distortions.

7 Discussion

7.1 Potential Countermeasures

To defend against various side-channel attacks, we recommend implementing the following countermeasures from both hardware and software perspectives.

Automatic Elimination of Inaudible Sound at 18 kHz. Most AR/VR devices possess advanced audio processing capabilities that operate effectively within the range of human hearing. AR/VR devices need to filter out inaudible sounds with frequencies exceeding 18 kHz.

Secure Your Applications. AR/VR users should ensure that their devices have the latest security updates installed and only download new applications from official app stores (e.g., Meta Store) or other trusted sources.

Restrict Application Permissions. AR/VR users should carefully evaluate permission requests from each application, especially for microphones and sensors, authorizing them only when required [15].

7.2 Limitations

This study still has room for improvement, which we will enhance in future work.

Limited Attack Range. Constrained by AR/VR hardware and the restricted sound propagation, `AcouListener` has a limited attack range. Future work may use high-power emitters, sensitive microphone arrays and collaborative devices to form distributed sensing systems that expand spatial coverage.

Limited Inference Content. The current framework can only classify straightforward gestures (e.g., unlocking patterns, handwriting letters, and typing words). Future work includes developing more accurate tracking algorithms, incorporating microphone arrays, and exploring multimodal sensor fusion and deep learning techniques to support more diverse hand gestures.

8 Conclusion

In this paper, we investigate a novel inaudible acoustic side-channel attack within emerging AR/VR scenarios. We primarily leverage the unique characteristics of received acoustic signals generated by a victim's hand movements, which reflect inaudible signals emitted by a camouflaged VR device or mobile phone. After processing the received signals into dCIR sequences, we employ a portable deep CNN to effectively extract key features from the collected dCIR images. Moreover, we have developed a user-friendly mobile application that seamlessly integrates the entire system for practical use. Extensive experiments conducted with ten volunteers in three different scenarios have demonstrated that our attack system can accurately infer victims' unlocking patterns, handwriting styles, and typed words across two types of virtual keyboards. Furthermore, we also propose several effective countermeasures to mitigate this new and potentially dangerous acoustic side-channel attack, thereby improving user security in AR/VR.

Acknowledgements. This work is partially supported by HKBU (with no. RC-SFCRG/23-24/R2/SCI/06) and NSFC (with no. 62172277).

References

1. Ali, K., Liu, A.X., Wang, W., Shahzad, M.: Keystroke Recognition Using WiFi Signals. In: Proceedings of the 21st annual international conference on mobile computing and networking. pp. 90–102 (2015)
2. Alla, I., Olou, H.B., Loscrì, V., Levorato, M.: From sound to sight: Audio-visual fusion and deep learning for drone detection. In: Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec). pp. 123–133. ACM (2024)
3. Andriotis, P., Oikonomou, G.C., Mylonas, A., Tryfonas, T.: A study on usability and security features of the android pattern lock screen. Inf. Comput. Secur. **24**(1), 53–72 (2016)
4. Arafat, A.A., Guo, Z., Awad, A.: VR-Spy: A side-channel attack on virtual keylogging in VR headsets. In: IEEE Virtual Reality and 3D User Interfaces, VR 2021, Lisbon, Portugal, March 27 - April 1, 2021. pp. 564–572. IEEE (2021)
5. Bae, S.H., Choi, I.K., Kim, N.S.: Acoustic scene classification using parallel combination of LSTM and CNN. In: Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE). pp. 11–15 (2016)
6. Bai, Y., Lu, L., Cheng, J., Liu, J., Chen, Y., Yu, J.: Acoustic-based sensing and applications: A survey. Comput. Networks **181**, 107447 (2020)
7. Deshotels, L.: Inaudible sound as a covert channel in mobile devices. In: 8th USENIX Workshop on Offensive Technologies (WOOT 14) (2014)
8. Du, H., Li, P., Zhou, H., Gong, W., Luo, G., Yang, P.: Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning. In: IEEE Conference on Computer Communications (INFOCOM). pp. 1448–1456 (2018)
9. Espi, M., Fujimoto, M., Kinoshita, K., Nakatani, T.: Exploiting spectro-temporal locality in deep learning based acoustic event detection. EURASIP J. Audio Speech Music. Process. **2015**, 26 (2015)
10. Fu, C., Du, X., Zeng, Q., Zhao, Z., Zuo, F., Di, J.: Seeing Is Believing: Extracting Semantic Information from Video for Verifying IoT Events. In: Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec). pp. 101–112. ACM (2024)
11. Genkin, D., Nissan, N., Schuster, R., Tromer, E.: Lend me your ear: Passive remote physical side channels on pcs. In: 31st USENIX Security Symposium, USENIX. pp. 4437–4454. USENIX Association (2022)
12. Gopal, S.R.K., Shukla, D., Wheelock, J.D., Saxena, N.: Hidden reality: Caution, your hand gesture inputs in the immersive virtual world are visible to all! In: 32nd USENIX Security Symposium. pp. 859–876. USENIX Association (2023)
13. Huang, W., Tang, W., Chen, H., Jiang, H., Zhang, Y.: Unauthorized microphone access restraint based on user behavior perception in mobile devices. IEEE Transactions on Mobile Computing **23**(1), 955–970 (2024)
14. Kaminska, D., Sapinski, T., Wiak, S., Tikk, T., Haamer, R.E., Avots, E., Helmi, A., Ozcinar, C., Anbarjafari, G.: Virtual reality and its applications in education: Survey. Inf. **10**(10), 318 (2019)
15. Kim, Y., Goutam, S., Rahmati, A., Kaufman, A.E.: Erebus: Access control for augmented reality systems. In: 32nd USENIX Security Symposium, USENIX Security 2023. pp. 929–946. USENIX Association (2023)
16. Li, D., Liu, J., Lee, S.I., Xiong, J.: Fm-track: pushing the limits of contactless multi-target tracking using acoustic signals. In: Nakazawa, J., Huang, P. (eds.) SenSys '20: The 18th ACM Conference on Embedded Networked Sensor Systems, Virtual Event, Japan, November 16-19, 2020. pp. 150–163. ACM (2020)

17. Li, D., Liu, J., Lee, S.I., Xiong, J.: Room-scale hand gesture recognition using smart speakers. In: Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys). pp. 462–475. ACM (2022)
18. Li, J., Meng, Y., Zhan, Y., Zhang, L., Zhu, H.: Dangers behind charging VR devices: Hidden side channel attacks via charging cables. IEEE Trans. Inf. Forensics Secur. **19**, 8892–8907 (2024)
19. Ling, K., Dai, H., Liu, Y., Liu, A.X., Wang, W., Gu, Q.: Ultragesture: Fine-grained gesture sensing and recognition. IEEE Trans. Mob. Comput. **21**(7), 2620–2636 (2022)
20. Lu, L., Yu, J., Chen, Y., Zhu, Y., Xu, X., Xue, G., Li, M.: Keylistener: Inferring keystrokes on QWERTY keyboard of touch screen through acoustic signals. In: 2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019. pp. 775–783. IEEE (2019)
21. Luo, S., Hu, X., Yan, Z.: Hologlogger: Keystroke inference on mixed reality head mounted displays. In: IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2022, Christchurch, New Zealand, March 12-16, 2022. pp. 445–454. IEEE (2022)
22. Luo, S., Nguyen, A., Farooq, H., Sun, K., Yan, Z.: Eavesdropping on controller acoustic emanation for keystroke inference attack in virtual reality. In: The Network and Distributed System Security Symposium (NDSS). vol. 2 (2024)
23. Mei, L., Liu, R., Yin, Z., Zhao, Q., Jiang, W., Wang, S., Lu, K., He, T.: mmspyvr: Exploiting mmwave radar for penetrating obstacles to uncover privacy vulnerability of virtual reality. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **8**(4), 172:1–172:29 (2024)
24. Meteriz-Yidiran, Ü., Yýldýran, N.F., Mohaisen, D.: Acoustictype: Smartwatch-enabled cross-device text entry method using keyboard acoustics. In: CHI Conference on Human Factors in Computing Systems Extended Abstracts. pp. 352:1–352:7. ACM (2022)
25. Meteriz-Yildiran, Ü., Yildiran, N.F., Awad, A., Mohaisen, D.: A keylogging inference attack on air-tapping keyboards in virtual environments. In: IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2022, Christchurch, New Zealand, March 12-16, 2022. pp. 765–774. IEEE (2022)
26. Nguyen, A., Zhang, X., Yan, Z.: Penetration vision through virtual reality headsets: Identifying 360-degree videos from head movements. In: 33rd USENIX Security Symposium. USENIX Association (2024)
27. Pensieri, C., Pennacchini, M.: Overview: Virtual Reality in Medicine. Journal For Virtual Worlds Research **7**(1) (2014)
28. Qamar, A., Karim, A., Chang, V.: Mobile malware attacks: Review, taxonomy & future directions. Future Gener. Comput. Syst. **97**, 887–909 (2019)
29. Slocum, C., Zhang, Y., Abu-Ghazaleh, N.B., Chen, J.: Going through the motions: AR/VR keylogging from user head motions. In: 32nd USENIX Security Symposium. pp. 159–174. USENIX Association (2023)
30. Stephenson, S., Pal, B., Fan, S., Fernandes, E., Zhao, Y., Chatterjee, R.: Sok: Authentication in augmented and virtual reality. In: 43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022. pp. 267–284. IEEE (2022)
31. Stephenson, S., Pal, B., Fan, S., Fernandes, E., Zhao, Y., Chatterjee, R.: Sok: Authentication in augmented and virtual reality. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 267–284 (2022)

32. Wang, H., Zhan, Z., Shan, H., Dai, S., Panoff, M., Wang, S.: Gazeploit: Remote keystroke inference attack by gaze estimation from avatar views in VR/MR devices. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024. pp. 1731–1745. ACM (2024)
33. Wang, R., Huang, L., Madden, K., Wang, C.: Enhancing QR code system security by verifying the scanner's gripping hand biometric. In: Kim, Y., Kim, J., Koushanfar, F., Rasmussen, K. (eds.) Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2024, Seoul, Republic of Korea, May 27-29, 2024. pp. 42–53. ACM (2024)
34. Wang, W., Liu, A.X., Sun, K.: Device-free gesture tracking using acoustic signals. In: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom). pp. 82–94. ACM (2016)
35. Wang, Y., Shen, J., Zheng, Y.: Push the limit of acoustic gesture recognition pp. 566–575 (2020)
36. Wu, Y., Shi, C., Zhang, T., Walker, P., Liu, J., Saxena, N., Chen, Y.: Privacy leakage via unrestricted motion-position sensors in the age of virtual reality: A study of snooping typed input on virtual keyboards. In: 44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023. pp. 3382–3398. IEEE (2023)
37. Yang, Z., Sarwar, Z., Hwang, I., Bhaskar, R., Zhao, B.Y., Zheng, H.: Can virtual reality protect users from keystroke inference attacks? In: 33rd USENIX Security Symposium. USENIX Association (2024)
38. Yun, S., Chen, Y., Zheng, H., Qiu, L., Mao, W.: Strata: Fine-grained acoustic-based device-free tracking. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys). pp. 15–28. ACM (2017)
39. Zhang, Y., Slocum, C., Chen, J., Abu-Ghazaleh, N.B.: It's all in your head(set): Side-channel attacks on AR/VR systems. In: Calandrino, J.A., Troncoso, C. (eds.) 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023. pp. 3979–3996. USENIX Association (2023)
40. Zhou, W., Zhou, Y., Jiang, X., Ning, P.: Detecting repackaged smartphone applications in third-party android marketplaces. In: Second ACM Conference on Data and Application Security and Privacy, CODASPY 2012, San Antonio, TX, USA, February 7-9, 2012. pp. 317–326. ACM (2012)
41. Zhou, Y., Wang, Z., Zhou, W., Jiang, X.: Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. In: 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012. The Internet Society (2012)

Appendix A CNN model

We have developed an Android-based application (app) to infer hand gestures[9]. Table 1 lists the detailed settings of our adopted MobileNet V2 model.

- t denotes the expansion factor, which is the factor by which the number of channels is expanded in the bottleneck layer.
- c denotes the number of output channels from each layer.

[9] `https://github.com/adhakdh/AcouListener`

Table 1: The adopted CNN model structure.

Input	**Operator**	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgPool 7×7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	-

– n denotes the number of repetitions, meaning how many times the operation is repeated.
– s denotes the stride of the convolution or pooling operation.
– k denotes the number of classes ($k = 10$ in Attack Scenario 1 (§ 5.2) and Attack Scenario 3 (§ 5.4); $k = 15$ in Attack Scenario 2 (§ 5.3)).

Appendix B Typing Content

Table 2 lists major typing contents inferred by adversaries in § 5.4.

Table 2: Typing contents.

ID	**(1) Websites**	**(2) Passwords**	**(3) Words**
1	google.com	`123456`	gym
2	youtube.com	`123456789`	first
3	baidu.com	`qwerty`	virtual
4	bilibili.com	`password`	netflix
5	facebook.com	`1234567`	bigscreen
6	qq.com	`12345678`	deovr
7	twitter.com	`12345`	skybox
8	zhihu.com	`iloveyou`	fitxr
9	wikipedia.org	`111111`	vr
10	amazon.com	`123123`	win