Speak Up, I'm Listening: Extracting Speech from Zero-Permission VR Sensors

Derin Cayir*, Reham Mohamed Aburas†, Riccardo Lazzeretti§, Marco Angelini¶
Abbas Acar*, Mauro Conti‡, Z. Berkay Celik◇, Selcuk Uluagac*
*Florida International University
†American University of Sharjah, §Sapienza University of Rome
¶Link Campus University of Rome, ‡University of Padua, ◇Purdue University
Emails: {dcayi001, aacar001, suluagac}@fiu.edu, raburas@aus.edu
lazzeretti@diag.uniroma1.it, m.angelini@unilink.it, mauro.conti@unipd.it, zcelik@purdue.edu

***Abstract*—As Virtual Reality (VR) technologies advance, their application in privacy-sensitive contexts, such as meetings, lectures, simulations, and training, expands. These environments often involve conversations that contain privacy-sensitive information about users and the individuals with whom they interact. The presence of advanced sensors in modern VR devices raises concerns about possible side-channel attacks that exploit these sensor capabilities. In this paper, we introduce IMMERSPY, a novel acoustic side-channel attack that exploits motion sensors in VR devices to extract sensitive speech content from on-device speakers. We analyze two powerful attacker scenarios: *informed* attacker, where the attacker possesses labeled data about the victim, and *uninformed* attacker, where no prior victim information is available. We design a Mel-spectrogram CNN-LSTM model to extract digit information (e.g., social security or credit card numbers) by learning the speech-induced vibrations captured by motion sensors. Our experiments show that IMMERSPY detects four consecutive digits with 74% accuracy and 16-digit sequences, such as credit card numbers, with 62% accuracy. Additionally, we leverage Generative AI text-to-speech models in our attack experiments to illustrate how the attackers can create training datasets even without the need to use the victim's labeled data. Our findings highlight the critical need for security measures in VR domains to mitigate evolving privacy risks. To address this, we introduce a defense technique that emits inaudible tones through the Head-Mounted Display (HMD) speakers, showing its effectiveness in mitigating acoustic side-channel attacks.**

I. INTRODUCTION

Virtual Reality (VR) devices have become increasingly popular due to ongoing advancements that have greatly enhanced their immersive capabilities. The global market for immersive technologies, valued at $27.41 billion in 2023, is projected to reach approximately $167.75 billion by 2032 [45]. The increasing digitalization and rapid technological advancements further add momentum to the integration of VR technologies across multiple domains, including education, healthcare, gaming, and retail.

VR devices integrate sophisticated sensors to construct immersive environments, which enhances the user experience by accurately tracking movements and interactions within the virtual space [27]. This creates a more realistic and engaging environment. However, attackers can exploit these sensors to obtain privacy-sensitive user information. For instance, an attacker can fool users into installing a malicious app, which can then transmit sensor data to infer the indoor location of a user and identify their identity, routines, and activities [28]. Such attacks have been thoroughly investigated on smartphones and, more recently, on VR devices. In response, Android and iOS implement permission-based access control mechanisms to mitigate these risks [9], [3]. Consequently, apps must request explicit user permission before accessing sensor data. However, certain sensors, such as accelerometers, gyroscopes and magnetometers, are classified as "zero-permission" sensors [50], granting access without requiring user permission due to their need for the functionality of the app.

Previous research has investigated the potential of exploiting sensors in smart devices to extract sensitive information. These include decoding smartphone touch events [38], [19], [29], detecting keystrokes [34], [52], and analyzing user activities [51] (e.g., driving and walking). Prior studies also investigated sensors in VR such as inferring user locations from spatial maps [28], estimating body fat ratios [55], and even re-identifying users across different sessions [39]. While some studies have explored Inertial Measurement Unit (IMU) side-channel vulnerabilities in VR [40], [34], [56], the specific exploitation of these sensors to recover speech content played on the device remains unexplored. In addition, there is a need for a defense mechanism that mitigates these attacks without compromising the functionality of VR devices [27].

In this paper, we investigate how an attacker can exploit the motion sensors on VR devices to extract sensitive user information, such as phone numbers, SSNs, and credit card numbers. We observe that audio signals produced by the VR device's speakers create fluctuations in the IMU sensors located on the Head Mounted Display (HMD). These fluctuations can reveal private information about the speech the VR user is listening to. Importantly, the closer the IMU sensors are to the speakers on the HMD, the more susceptible they are to audio

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.240164
www.ndss-symposium.org

signals, amplifying the impact of such side-channel attacks. As VR devices become more compact with smaller HMDs, this proximity increases, making such attacks increasingly feasible.

To demonstrate the feasibility of our observation in practice, we introduce IMMERSPY, which considers an attack model that does not require the victim's training data to leak user's sensitive data. Instead, we develop a model in which an attacker leverages Generative AI (GenAI) and text-to-speech models to collect their own training data. This approach is more practical than previous works in the extended reality literature, which typically rely on models trained with the target victim's data [39], [48], [52]. Specifically, our attacker model does not require the specific victim's pronunciation or head movement pattern data during speech [48]. Using GenAI, the attacker generates diverse audio content spanning various languages, age groups, genders, and contexts, facilitating the detection of specific words and user profiles. The freedom to generate audio content and collect corresponding sensor data allows the attacker to explore a multitude of scenarios and adapt their training set to real-world conditions.

To achieve this goal, we start by applying a series of preprocessing steps to the 3D accelerometer signals. These steps filter out residual noise and motion, focusing on extracting high frequencies that correspond to speech content. Subsequently, we design a Mel-spectrogram CNN-LSTM-based model that classifies the extracted speech features derived from the accelerometer data into their corresponding spoken digits. By analyzing spectrograms on the Mel scale, the human ear's perception of sound is mimicked, allowing our model to detect even subtle differences in frequencies and thereby achieving higher accuracy in understanding speech.

We demonstrate that an attacker with access to the victim's labeled data (*informed* attacker) can achieve over 72% accuracy in understanding spoken digits using off-the-shelf models. This accuracy is further improved to 85% with our proposed Mel-spectrogram CNN-LSTM model. In particular, even without prior knowledge about the victim (*uninformed* attacker), the attacker can still estimate spoken digits with an accuracy of 62% using off-the-shelf models, and 77% with the CNN-LSTM-based model. We also show that an attacker can retrieve consecutive digits, such as the last four digits of SSNs with an accuracy of up to 80%, date of birth with an accuracy of up to 78%, phone numbers with 77%, and credit card numbers with an alarmingly high success rate of up to 76%. In addition, we introduce a defense method against speech eavesdropping through benign sensors. By carefully playing additional pure frequencies through the HMD speakers, we reduce the accuracy of *informed* and *uninformed* attacks on average by 70%.

In summary, we make the following contributions:

- We introduce IMMERSPY, a novel sensitive data inference attack on VR devices that exploits motion sensors affected by speech vibrations from on-device speakers.
- We leverage GenAI and text-to-speech models to create diverse and realistic training data, improving the attack success rate without requiring victim data.
- We evaluate our system, IMMERSPY, using the Free Spoken Digit Dataset (FSDD) and TIDIGITS corpus datasets, which include 16 online English-speaking participants. The VR users are engaged in realistic movements to demonstrate the accuracy of IMMERSPY in predicting spoken digits under both *informed* and *uninformed* attack models.
- We introduce a novel defense method against speech eavesdropping through benign sensors and assess the attack performance when this defense is deployed.

II. BACKGROUND

A. VR Apps and App Development Platforms

VR technologies have found diverse applications across various domains from gaming, education, and healthcare. This is largely attributed to the availability of programmable VR devices. Many VR devices on the market support application development with different platforms such as Unreal Engine [18] and Unity [17]. These platforms provide comprehensive Application Programming Interfaces (APIs) [11], enabling developers to interact with the device's capabilities and sensors to create immersive experiences.

While the VR community benefits from the opportunity to customize applications to fit their needs, this flexibility and freedom also introduce significant security and privacy concerns. The customizability the APIs support, such as accessing raw sensor data from sensors that not only track the user's environment but also gather information about the user's behavioral characteristics, raises privacy concerns [44], [41], [43]. In addition to the app development environments, there are different app markets such as Meta Store [16] and Steam [12] for VR apps that support third-party apps. Independent app developers can publish on the marketplaces where any user can access that app and download it at their own cost.

B. Zero-permission Sensors

Sensors in devices can generally be categorized into two groups based on their access control requirements [50]: (i) sensors that require user permission to operate, such as GPS or camera; and (ii) zero-permission sensors, which can function without explicit user consent. IMU sensors such as accelerometers, gyroscopes, and magnetometers are zero-permission sensors. Typically, when an app attempts to access a sensor's readings, such as seeking access to the on-device camera, it must request the user's permission. Once granted, the app can then process the sensor data. In contrast, zero-permission sensors allow apps to access and process their data without any notification to the system or the user. This presents a profound privacy concern since malicious third-party app developers can access raw sensor data, potentially launching side-channel attacks without the user's knowledge. The distinction between these two categories is crucial, not only for app functionality, but also in terms of security and privacy implications. Zero-permission sensors, while less invasive, still collect potentially sensitive data.

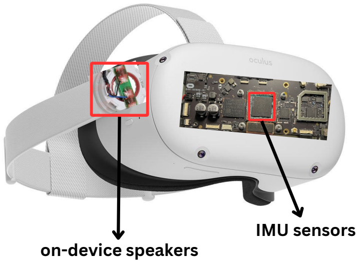


Fig. 1: Location of speaker and IMU on Meta Quest-2

C. Acoustic Signals' Effects on Sensors

The sound propagates through a medium through vibrations. As sound waves travel, they gradually lose energy and continue traveling until their energy completely dissipates. To hear a sound, an individual needs to be within the range of those vibrations. Speakers convert electrical energy to mechanical energy using a motor to produce sound. The motor drives the speaker back and forth, translating the movements into pressure waves traveling through the air. These pressure waves travel to the ear, which the brain interprets as sound.

Recognizing that the mediums through which sound travels can be the same environments in which sensors operate, researchers have considered the potential influence of sound waves on sensors [37]. Although IMU sensors are designed to remain unaffected by environmental factors such as humidity or temperature, they have been found to be susceptible to fluctuations induced by acoustic signals. The research of Michalevski and Boneh showed that the IMU sensors on smartphones, which share the same surface with the speakers, capture variations corresponding to the audio frequencies [37].

Other studies confirmed that when speakers and sensors share a medium, there are noticeable effects of speech on sensor readings [22], [23], [26]. In addition, researchers investigated whether machine-generated speech or live human speech has a more pronounced impact on sensor readings. They determined that machine-generated speech affects sensor readings when there are surface or conductive vibrations, whereas the effect of human speech on motion sensors is less noticeable [22]. Therefore, the attacker's success increases when extracting speech content from machine-rendered audio and when speakers and IMU sensors share the same medium, ideally within the same device, which is the case in VR devices.

D. VR Specific Challenges

Previous studies show that speakers located on the device create vibrations that travel to the IMU sensors located near the speakers [26], [23]. These studies focused on smartphones, where due to their compact nature, the motherboard containing the IMU is tightly packed to the speakers. However, in VR devices, e.g. Meta Quest-2, the two speakers are located on the sides of the HMD as shown in Figure 1, which emits more immersive sounds. However, the motherboard is located right behind the lenses on the front of the HMD. Consequently, the distance between the IMU sensors and speakers is significantly larger than in smarphone layouts.

Furthermore, the effect of head movement on accelerometer sensors when aiming to extract speech-related signals adds additional complexity, as VR devices are designed to be interactive, involving whole-body movements. Hence, it could be said that with a VR device, the effect of audio signals on IMU sensors would be less evident (compared to a more compact smart device) if the layout and use cases are considered. Additionally, side-channel attacks utilizing smartphone IMUs are currently limited to 500 Hz, with additional restrictions imposed by permissions starting from Android-12 and onward. In contrast, the high sampling rate capability of VR devices, which can reach up to 1000 Hz, continues to present an ongoing opportunity for acoustic side-channel attacks in the VR domain.

III. Threat Model

A. Attack Vector

VR devices provide app developers with access to a variety of sensor data through APIs. These devices are equipped with numerous sensors, ranging from LiDAR for environmental tracking to inward-facing cameras for monitoring users' eye movements [27]. When an app attempts to access the microphones or cameras, for purposes like hand tracking or environmental sensing, a permission request is triggered, as these sensors require explicit user consent before activation. Due to the sensitive nature of the data captured by these sensors, they are restricted to functioning only within the foreground app. Background apps are prohibited from actively using them [14]. In contrast, sensors such as accelerometers and gyroscopes operate without user permission. These sensors are generally regarded as benign, as they are designed to capture motion-related data, such as the acceleration and orientation of the user's head movements. However, in our attack, ImmerSpy, we exploit these zero-permission sensors to extract spoken content from an app running in the background, bypassing the typical permission requirements associated with more sensitive sensors. These zero-permission policies are consistent across several major VR operating systems, including Meta's MetaOS, Apple's VisionOS, and Microsoft's Windows Mixed Reality. An example of this is Meta Quest Move, which counts the calories burned through the background access to the accelerometer [46].

B. Attacker's Capabilities

In ImmerSpy attack a "malicious" app running in the background on a compromised host's (Alice) device uses motion sensors to retrieve the spoken digits from conversations with the victim (Bob) in the foreground app. The foreground app plays the audio of Bob while speaking sensitive credentials. Similarly, in our threat model, the malicious app may solely run on Bob's device, such as when he uses voice assistants like Meta AI to relay sensitive credentials which Meta AI repeats aloud for confirmation.

The audio produced from the foreground app travels from the onboard speakers to the IMU sensors. The attacker can

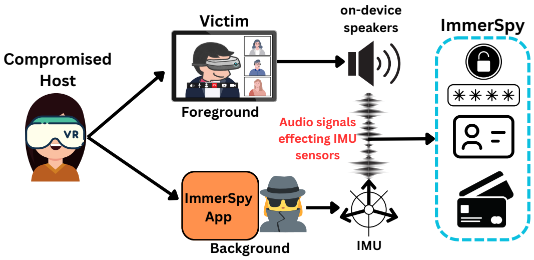
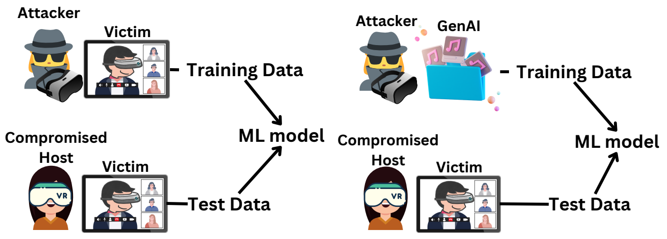


Fig. 2: Attack model overview.



(a) Informed attacker

(b) Uninformed attacker

Fig. 3: Comparison of *informed* and *uninformed* attackers.

analyze these fluctuations in the sensor readings to retrieve sensitive spoken digit content. Existing VR apps designed for private company and AI-agent meetings such as Horizon Workrooms [15] and Ovation [8] are susceptible to IMMERSPY attack by capturing sensitive information spoken during meetings (e.g. sales data and account numbers). These apps can operate in the foreground, while the IMMERSPY attack runs in a background application, relying solely on motion sensors to decipher the spoken digits in the foreground app.

To understand spoken digits of the victim through the accelerometer data of VR, the attacker conducts the attack in three stages: (1) malicious app creation, (2) data collection from the victim, (3) model training and testing.

1) Malicious App Creation: The attacker creates an app that has malicious intentions, named "IMMERSPY App", as shown in Figure 2. The app will be running on the background to extract the spoken digits through only the motion sensors. The app will exploit the perceived safety of permissionless apps, such as fitness and calorie trackers (e.g., Meta Move app [1]), which meet marketplace security standards [10] despite using IMU sensors for movement tracking. Due to zero-permission sensors, any app can access these sensors via their API through a background app [49], [50], [42] without requiring the user's explicit permission. This makes the app appear benign as it does not access other sensors that raise privacy concerns, such as external cameras used for hand tracking. Consequently, users who download the app should not hesitate, assuming that it poses no privacy risks. Additionally, the attacker does not need to include any malicious code in their app since the IMMERSPY attack relies solely on motion sensor data to extract users' private information [36].

2) Data Collection: Once the Alice downloads and operates the malicious app on her device, the app starts a background activity to collect data from the HMD's accelerometer sensor. Meanwhile, a benign app running in the foreground plays audio of Bob through the HMD's on-device speakers, affecting the motion sensors. Hence, the malicious app running in the background captures these audio-induced effects on the sensors. Once the sensor data are collected, processing is conducted entirely on the device.

3) Model Training and Testing: After successfully collecting motion sensor data from the HMD, the attacker launches the IMMERSPY attack to infer the audio signals emitted by the on-device speakers. IMMERSPY applies feature engineering to extract audio features from accelerometer data and leverages a Mel spectrogram CNN-LSTM model to understand private information about the victim, Bob, from the extracted audio.

Depending on the attacker's capabilities and access to the victim's data, model training can occur both offline and online. Initially, the attacker may train the model offline using a comprehensive dataset to develop a foundational understanding of the victim's typical audio patterns. Once the application is operational on the host's device, the attacker can further refine the model online by integrating new data continuously gathered from the victim. Conversely, in scenarios where the attacker lacks prior specific victim data, they may opt for a fully offline approach, employing a generic dataset enhanced through the use of GenAI techniques. The model is then evaluated on live data from the victim. The attacker tests the model's effectiveness by analyzing how well it can interpret new and live data without the benefit of pre-existing knowledge, detailed in Section III-C.

C. Attacker Scenarios

We examine two different attacker scenarios: i) *informed* attacker and ii) *uninformed* attacker.

1) Informed Attacker Scenario: This scenario assumes that the attacker has previously obtained labeled data about the victim. This could be achieved by analyzing recordings from previous meetings or obtaining the victim's video or audio from social media platforms, e.g. Instagram or YouTube.

Methodology: The attacker deploys a multifaceted approach to collect victim data. This involves using a malicious app (as explained in Section III-B) that continuously extracts motion sensor readings from the victim's HMD while playing the victim's audio through the speakers.

To obtain labels, the attacker requires prior knowledge of the victim, which is achievable in two ways. The first involves placing a microphone near the HMD correlating sensor data with real-time audio, such as spoken digits. Alternatively, the attacker could invite the victim to a virtual meeting on the VR platform or play the victim's videos (as shown in Figure 3a). The attacker's device would then play the victim's audio, impacting the motion sensors and allowing for data collection on the attacker's own device without the victim's use of a malicious app.

The data acquired from the victim are then fed into the ML model to understand the victim's spoken digits. We emphasize that this scenario is likely to perform better compared to the *uninformed* attacker scenario, since the *informed* attacker includes the victim data in the training set.

2) Uninformed Attacker Scenario: This attacker does not have access to the victim's labeled data. Thus, they create their model using audio data of other speakers.

Methodology: Since the attacker cannot use the victim's data in their training set, they must collect data from other speakers using their own device before launching the attack. The attacker has the flexibility to train their model with data gathered from various sources, including different GenAI models and publicly available online audio libraries. Once trained, the attacker extracts the victim's motion data through the "ImmerSpy App" running on the host's device.

Compared to the *informed* attacker scenario, the *uninformed* attacker scenario is more practical, as it allows the attacker to not have victim's labeled data. This method is also more scalable, allowing the attacker to target multiple victims without requiring specific video or audio files for each victim. However, due to the absence of the victim's labeled data, it is expected that the accuracy of this attack will be lower than that of the *informed* attacker scenario.

D. Examples on Attacker's Goals

The attacker's ultimate goal is to extract sensitive user information, primarily through the audio produced by speakers. For example, during a social gathering, the app could potentially analyze conversations around the user. Using hot-word classification, the attacker can extract details from personal or professional discussions. By training a model to classify spoken digits, the attacker could potentially deduce PINs, credit card numbers, or SSNs. For instance, Alice (a banker) and Bob (a customer) are in a virtual meeting to discuss opening a savings account where the attacker with a spy app running on Alice's device uses IMU sensor readings to eavesdrop on their conversation and infer sensitive information, such as Bob's credit card number, date of birth, or SSN.

Furthermore, an attacker might resort to using ImmerSpy if their primary surveillance methods are compromised, such as when a hidden microphone is discovered and removed. In this situation, ImmerSpy becomes an alternative approach to continue extracting private information from the victim.

While these methods are effective for limited dictionary attacks, the attacker could expand the scope of extracted sensitive conversations by enlarging the dictionary in the training set or recovering spoken vowels by analyzing word pronunciations. This would enable the attacker to retrieve dialogues from free speech, increasing their access to sensitive information.

E. Feasability Study

We evaluate the impact of different audio volumes on the VR device's IMU sensor readings to understand the attack feasibility. To achieve this, we play audio on a Meta Quest-2 device at varying volumes; no-volume (0%), low-volume (25%), mid-volume (50%), high-volume (75%), and full-volume (100%). For each volume level, the audio is played for three seconds, while the sensor readings are collected. Furthermore, in each scenario, the VR device is held stationary on a desk with no external movement affecting the sensor.

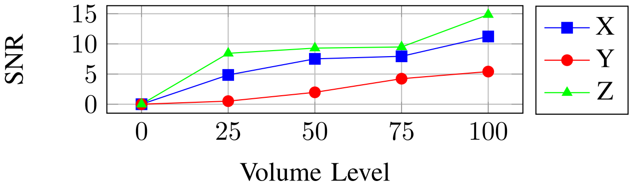


Fig. 4: Signal-to-Noise Ratio (SNR) across different volume levels for each axis (x, y, z) of the VR HMD's accelerometer.

To quantify the influence of audio volumes on the sensor readings, we calculate the Signal to Noise Ratio (SNR) for each session. In this context, the "signal" refers to sensor readings with speech-induced vibrations, and the "noise" corresponds to the readings taken during silence. The SNR for each axis is calculated using the following formula, where P_S and P_N represent the power of sensor readings during speech playback and silence, respectively, where $\text{SNR}_{\text{axis}} = 10 \cdot \log_{10}\left(\frac{P_S}{P_N}\right)$.

Our results for SNR values are presented in Figure 4. The results show that, as the volume in the speakers increases, the SNR values across all positional values increase. This trend is particularly marked on the x and z axes, where the SNR values show substantial increases at higher volumes, confirming a strong dependency of sensor signal clarity on audio volume. This analysis shows the consistent and increasing impact of audio volume on motion sensors, particularly at higher volumes.

F. Challenges

(C1) Removing Head-associated Movements' Effect: IMU sensors are specifically designed to capture the user movements within a VR device. Consequently, the effects of head movements on the sensor readings are typically more pronounced than those caused by audio signals from the speakers. Therefore, to accurately isolate and analyze the impact of audio signals on sensor readings, it is essential to remove the interference caused by head movements. We achieve this by applying a preprocessing pipeline to filter out the signals associated with head movements that reside in lower frequencies.

(C2) Low Sampling Rates: Human speech resides in the frequency range of 100 Hz to 5 kHz. With the Nyquist theorem, a sampling frequency of at least 10kHz is needed to capture the entire range of human speech. However, modern smart device vendors limit the sampling rates of the IMU sensors, and game development engines limit the VR HMD sensor sampling rate. Hence, capturing the complete content of the speech from motion sensors in HMDs presents significant challenges. Therefore, we choose to represent the accelerometer data as spectrograms. Although sensors cannot capture the full frequency spectrum of human speech due to limited sampling rates, spectrograms transform time-series data into a frequency-time representation and capture aliased signals, avoiding loss of valuable information.

(C3) Deriving Audio-Signal Characteristics from Sensor Readings: Understanding audio signals from motion sensor recordings is an unconventional approach that demands a deep understanding of the characteristics that represent specific aspects of speech. It remains unclear which types of features can effectively reveal speech content or the distinctive characteristics of the speaker. For this, we create a Mel spectrogram-based CNN-LSTM model with time-distributed layers to learn short and long-term features in the audio signals.

IV. ImmerSpy Attack Overview

We present ImmerSpy, speech inference attack that leverages zero-permission sensors on VR devices to infer private speech information. Figure 5 illustrates the overview of ImmerSpy attack. The input of the attack is the accelerometer data from VR devices, which continuously capture head-associated movements. The attacker's goal is to extract spoken digits from this data, despite the predominance of motion-related noise.

First, we segment the data into window samples. As VR devices are worn on the head and capture continuous head-associated movements, it is crucial to mitigate the influence of these movements on sensor readings. To effectively remove the impact of these movements, which occupy lower frequencies, we leverage high-pass filters (addressing **C1**). Once the data is filtered, our next step involves detecting segments containing speech. To achieve this, we compute the standard deviation (STD) of sensor readings within each window across the x, y, and z axes. Subsequently, we apply a predefined threshold to these STD values to determine the presence of speech, since speech patterns tend to exhibit higher variability in the signal after filtering the head movement effect.

Following the detection of speech areas, we generate the Mel spectrograms from the extracted sensor data, capturing a wide range of frequencies (addressing **C2**). Spectrograms transform the time-series sensor data into a frequency-time representation, making it suitable for image-based classification models. The Mel scale is particularly advantageous as it mimics the human ear's perception of sound, enabling effective detection of speech information. Hence, we build our model based on Mel spectrograms to recover spoken digits (addressing **C3**). Our model comprises multiple convolutional blocks connected in sequence, each designed to extract and learn different features of the input spectrograms. After passing through these layers, the data is flattened and fed into LSTM layers. These LSTM layers play a crucial role in capturing temporal dependencies and dynamics within speech signals, essential to accurately reconstruct the sequence of spoken digits.

Comparison with Prior Work: Prior research on speech analysis using IMU sensor data [30], [37], [23], [26] has focused on smartphones which by nature the sensors gets effected by motion less compared to the whole body motions in VR devices. Hence ImmerSpy starts off the analysis by removing the head-associated movements. Previous work focused on dominant-axis values to interpret speech, but ImmerSpy integrates multimodal data from all axes, extracting more comprehensive information from accelerometer readings. Unlike previous work on smartphones [26], [37], [23], ImmerSpy attack utilizes GenAI to not require victim data during training, making it more practical. Furthermore, it employs a Mel-spectrogram-based CNN-LSTM model that captures the entire 1000 Hz sampling spectrum, mimicking how the human ear interprets speech. By analyzing the accuracy in predicting up to sixteen consecutive digits, ImmerSpy evaluates realistic scenarios. More details of the comparison results of ImmerSpy with previous work are given in Section VI-E.

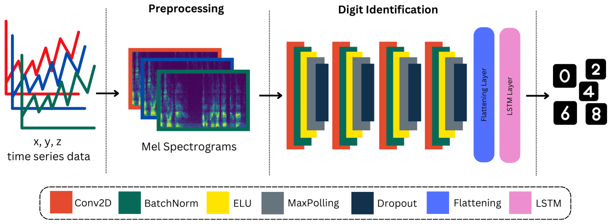


Fig. 5: Overview of ImmerSpy attack.

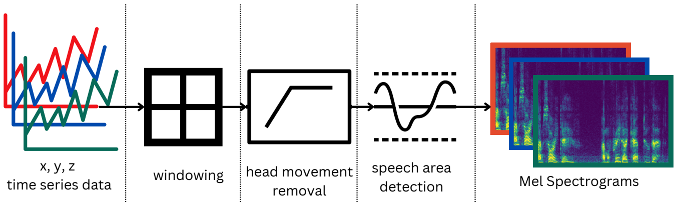


Fig. 6: ImmerSpy's preprocessing steps.

V. ImmerSpy Design

A. Data Collection

The attacker can extract the digits spoken through the accelerometer data only. For this, they run their app that continuously captures the IMU data from the HMD in the background, while a foreground app plays audio of the victim. This setup enables the attacker to access accelerometer data at 1000 Hz without requiring user permission. More details on the app designed for data collection are provided in Section VI.

B. Data Preprocessing

Figure 6 illustrates our data preprocessing module. Our objective is to separate speech-induced noise from accelerometer sensor data. This step is essential for filtering the sensor data to enable the extraction of spoken digits from continuous x, y, and z data streams. The preprocessing pipeline involves several key steps: segmenting the data into manageable parts, filtering out noise associated with head movements, detecting relevant data segments, and normalizing the data to standardize its range. Below, we detail each of these steps.

1) Head Motion Removal: Accelerometer sensors are designed to capture the device's movements. Therefore, to extract speech-related signals from accelerometer data, the signals coming from head motion should be removed. Figure 7 shows the

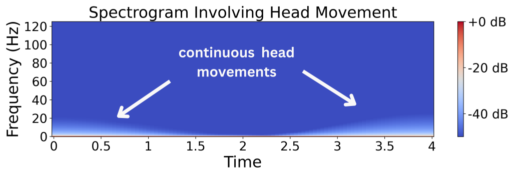


Fig. 7: The (zoomed in) spectogram results of head movement.

continuous head movement spectrogram. The horizontal axis displays the time and the vertical axis shows the frequencies. As can be seen, head movements created a periodic energy in low frequencies. Hence, using a high-pass filter to filter out head-associated movements will be ideal.

For effective removal of body movement in VR environments, where sensor capabilities surpass those typically found in smartphones, we utilize a Butterworth high-pass filter with a cutoff frequency of 20 Hz. Unlike smartphones, which often have a sampling rate around 250 Hz and cannot afford significant data loss, VR devices have much higher sampling rates. This allows for the application of such filters without risking the loss of a substantial amount of useful data. To reduce the transient response, the signal is pre-padded. Although some speech-related information at 20 Hz may be subject to aliasing, its impact is minimal within the broader 1000 Hz frequency spectrum. Therefore, the cost-benefit tradeoff is favorable.

2) Windowing: Windowing methods play a crucial role in speech analysis, allowing temporal segmentation of audio signals. This segmentation is essential for accurately capturing the dynamic nature of speech, including variations in pronunciations and vowels. For the window size, we have chosen the duration as 0.15 seconds, the duration of the shortest pronounced word. Previous work has used different values for window size parameters, such as also choosing it as the word duration that is the shortest [37], or selecting it as a random parameter [23], [26].

3) Speech Area Detection: The segmented data is passed through the speech area detection to detect the presence of speech in each window. For the digit classification task, a line of work has used isolated words during data collection [22], [37]. However, this method assumes the attacker is already aware of any speech occurrences. In practical scenarios, accelerometer data might lack any speech components during intervals where speakers do not emit any sounds. Hence, it is crucial to develop techniques to detect specific speech zones.

To detect speech areas, we conducted experiments in which a subject wears the Meta Quest 2 device and the audio of the speakers is played at varying volumes from 0% (indicating no speech impact on the accelerometer data) to 100%. With this, we find that the fluctuations in the STD across the accelerometer data were more evident compared to the change in energy from the summed spectrograms (a method proposed in previous works [48]). In particular, while STD increased noticeably in all parameters. Therefore, we propose a threshold that integrates the STD values of each of the x, y, and z parameters.

4) Zero-Mean Normalization: After the speech area is detected, the last step in data processing is zero-mean normalization. By ensuring a zero-mean data, we remove the bias introduced by the device's internal parameters like sensor offset.

C. Feature Extraction

In this section, we propose a set of features in time and frequency domains used for the digit classification through the accelerometer signal. The features used focus on capturing unique patterns associated with the pronunciation of each digit.

Time-Domain Features: The HMD's motion sensors record the position values as x, y and z. For time-domain features, we use six statistics: mean, STD, minimum, maximum, median, and variance, calculated from each of the values. This results in a total of $3 \times 6 = 18$ time-domain features.

Frequency-Domain Features: We leverage Mel-Frequency Cepstral Coefficients (MFCC) and was found effective for capturing the unique characteristics of spoken phonemes [37]. We choose the top 13 MFCC coefficients for each of the three recorded values and take their mean and STD as the features. This leads to $13 \times 2 \times 3 = 78$ frequency-domain features. Additionally, we use energy, entropy, and the dominant frequency ratio as the energy level and entropy can vary significantly among individuals, while the dominant frequency ratio helps capture variations in pitch and tone.

Spectrograms: The influence of audio on IMU sensors is observable across all x, y, and z axes. To fully capture the speech-related signal nuances, it is crucial to utilize spectrograms from all axes, a method not commonly employed in previous works. Previous studies in smartphone applications focused mainly on the analysis of dominant axes (specifically the z axis), where the influence of speech was more evident [26], [35]. In contrast, VR devices do not exhibit a single dominant axis, making the selection of one axis over others impractical. This necessitates a comprehensive multiaxis approach to accurately model the audio-IMU interaction within VR environments.

Spectrograms visualize and analyze the frequency spectrum over time. The Short-Time Fourier Transform (STFT) is applied with Hamming windows to convert the time-domain signal into the frequency domain. The squared magnitude of the STFT coefficients is mapped onto the Mel scale using 128 filters, capturing essential frequency components up to 500 Hz. Finally, the Mel spectrogram is converted to decibels, enhancing the perceptual relevance for further analysis. This approach allows for a better understanding of the intricate audio interactions in VR devices, focusing even on the subtle changes between frequencies with a Mel scale that resembles the human ear's perception of sound [24], [47].

D. Digit Identification

To recover spoken digits, we tested both conventional machine learning algorithms and advanced neural network architectures. We develop a neural network model designed to process time-distributed data, specifically tailored for Mel spectrogram inputs. The model architecture is composed of

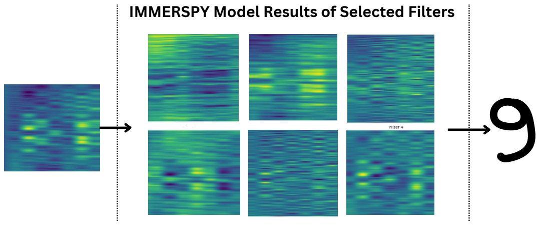


Fig. 8: Activation maps of the second layer in the neural network model for six selected filters. Each subplot represents the averaged activation of a specific filter across all time steps.

sequential time distributed convolutional layers to capture spatial features, followed by a bidirectional LSTM to analyze temporal dependencies, and dense layers for classification. Our model takes multiple channels (x, y, and z) of spectrograms as input. By analyzing these spectrograms from different coordinates altogether, the model captures a more holistic understanding of the spatial aspects of the data over time.

The spectrograms are first subjected to a series of 2D convolution (CNN) layers followed by batch normalization, ELU activation, and max pooling with time-distributed layers. Figure 8 shows the results of the activation layers of the second CNN layer. Each filter identifies different patterns from the input data. Some filters focus on specific frequency bands, while others capture broader spectral patterns. This diversity in filter responses allows the model to extract a wide range of features from the input spectrograms. Then, with LSTM, the model learns from the long-term temporal dependencies within the data. Finally, dense layers perform classification using the learned features of the preceding layers, with the final Softmax activation layer that outputs class probabilities.

To test the effectiveness of both *informed* and *uninformed* attackers, we compare the results of ImmerSpy with the baseline models [37]. We use random forest, decision tree, logistic regression, five closest neighbors, and support vector machine models with 10-fold cross-validation. These models require less computational resources and provide a strong baseline for performance comparison.

VI. Evaluation

We report the performance of ImmerSpy using FSDD [31] and TIDIGITS [25] datasets, while considering the effect of head movement on the sensors. We evaluated ImmerSpy for both *informed* and *uninformed* attacker scenarios. We analyze how the attacker's performance changes when their model operates as a complete black box and whether the size of the training set affects their accuracy. We examine ImmerSpy under different parameters for volume, audio format and headsets and test the effectiveness of our head-movement removal method by performing the attack with and without this filter. Finally, we conduct experiments to assess the performance of ImmerSpy in predicting consecutive digits, the attackers' abilities in using GenAI, and compare ImmerSpy's result with prior approaches.

Our study shows that ImmerSpy achieves an average accuracy of 85.6% in recognizing spoken digits when the attacker has access to the victim's labeled data. In a completely black-box scenario, where the attacker has no access to such data, ImmerSpy still performs robustly with an accuracy of 71.5%. These results significantly outperform the baseline models, demonstrating the effectiveness of ImmerSpy.

To evaluate our attack, we answer the following research questions:

RQ1: How effective is ImmerSpy for *informed* and *uninformed* Attackers in understanding the spoken digits?

RQ2: Does the size of the window affect the accuracy of the attack?

RQ3: How does the size of the training influence the accuracy in both attacker scenarios?

RQ4: How does ImmerSpy perform under different parameters (volume, audio format and headsets)?

RQ5: How effective is the attack at detecting the presence of speech and removing the effects of head-associated movements?

RQ6: How does ImmerSpy perform in predicting consecutive digits?

RQ7: Could the attacker use GenAI to enhance their dataset?

Evaluation Methodology: Our evaluation metrics include top-3 accuracies, precision, and recall. We focus on analyzing the model's first three predictions to determine how well it can potentially identify private information, such as the digits spoken by the user. We compute the accuracy as the number of correctly predicted digits in the test set. We calculate the precision for each digit class as the average ratio of correctly predicting the digit to the total number of windows classified to that type. We report recall as the total number of correctly predicted digits in that class to the ratio of the number of estimates (either true or false) of that digit.

Data Collection: We utilized the FSDD [31] and TIDIGITS [25] datasets. FSDD consists of recordings from six English-speaking male speakers where each speaker speaks digits 0 through 9, for 50 distinct times. These repetitions are not identical, and each utterance of a digit by a speaker is characterized by subtle variations in pronunciation. Each victim contributed 500 unique audio files with a total of 3,000 files across all victims. We evaluated ImmerSpy's ability to recover their spoken digits using accelerometer data while their audio files are played via an app. To examine how our attack performed on more speakers, we tested our model through TIDIGITS [25] dataset that contains five female and five male speakers, with 220 audio files. This data is collected in the same way as the FSDD data. Each audio file is played for 5 seconds, followed by a 1-second silence interval. This results in a total of 19,320 seconds of data collection with a roughly 1000 Hz sampling rate. During our data collection, we did not restrict the movements of our volunteers. They were free to sit, move around and even act like taking notes and attending a lecture/meeting. We collected data at 20-30 minute intervals across several days.

The pseudocode for the data collection and labeling process

Algorithm 1 *Data Collection:* This algorithm records the HMD's positional data synchronized with audio playback where the data is automatically labeled through the file's name.

```
Initialize Oculus SDK and create session
Set ‘audioFolderPath‘ as the directory of audio files
Set ‘dataFilePath‘ as the path for saving data
for each file in audioFolderPath do
    CollectData(audioFile, dataFilePath, session)
end for
Cleanup: Destroy Oculus session & Shutdown Oculus SDK
procedure CollectData(audioFile, filePath, session)
    Initialize data array
    Play audio file and record HMD data into data
    Format: ⟨ time, x, y, z, audioFile.wav ⟩
    Stop audio and record silence for a few seconds
    SaveDataToFile(filePath, data)
end procedure
procedure SaveDataToFile(filePath, dataLines)
    Open, write each line from data, and close file
end procedure
```

is presented in Algorithm 1. Each audio file is played sequentially. As each file is played, the app automatically labels the collected data with the audio file's name, which follows the format speakername_digit_trial.wav for automated processing.

To investigate the use of GenAI to enhance the dataset, we used BARK [20], TTS [5], Amazon Polly AI [13] and Meta Audiobox [32], open-source GenAI text-to-speech conversion tools. We chose 80 English-speaking AI-generated voices that speak the digits 0-9 to add to our training set. We collected the accelerometer data of these voices similarly.

Implementation Details: For data collection, we run a custom app developed in C++ in Visual Studio 2022 on Meta Quest 2. For the reproducibility of the work, we published our app code. The app uses OpenXR to continuously extract the motion sensor recordings from the HMD while it also plays audio files in the WAV format through the HMD's speakers. We achieved a sampling rate of around 1000 Hz during our data collection and stored these data locally for further extraction.

For building and refining the models, we used Python 3.11 with TensorFlow and Keras. The Mel spectrograms were created using Librosa, and high-pass filtering was performed with SciPy. Additional libraries such as Pandas were used for data manipulation, and Matplotlib was employed for visualizations.

A. Effectiveness of ImmerSpy *(RQ1)*

1) Informed Attacker Scenario: In this scenario, the attacker has access to the victim's labeled data. To enhance their dataset, the attacker may also include motion sensor data captured by playing online audio files through the HMD's speakers. To mimic this scenario, we assess the performance of our attack with a dataset that includes speech from both the victim and the other speakers.

ImmerSpy achieves an overall accuracy of 86% for detecting spoken digits when the attacker has labeled data of the victim in their training set. The average precision and recall are found as 86%, as shown in Table I.

TABLE I: ImmerSpy model results for both attackers.

Model	Accuracy	Avg. Precision	Avg. Recall
Informed	0.856	0.86	0.86
Uninformed	0.715	0.72	0.72

TABLE II: Informed attacker top-3 accuracy results with 10-fold cross-validation. The model that yields the best accuracy is ImmerSpy, highlighted in the table.

Model	Top-1	Top-2	Top-3
Random Forest	0.657 ± 0.022	0.806 ± 0.024	0.902 ± 0.025
Decision Tree	0.507 ± 0.023	0.601 ± 0.023	0.672 ± 0.021
Logistic Regression	0.504 ± 0.021	0.673 ± 0.02	0.792 ± 0.015
KNN	0.491 ± 0.019	0.647 ± 0.021	0.768 ± 0.023
SVM	0.723 ± 0.02	0.861 ± 0.02	0.937 ± 0.016
ImmerSpy	0.856 ± 0.02	0.962 ± 0.03	0.99 ± 0.01

We also compare the performance of ImmerSpy with baseline models. The performance metric results with the baseline models are given in Table II. The SVM model, the best-performing baseline model yields an accuracy of 72% in correctly estimating the spoken digit. With a second and third guess, the accuracy increases to 94%. With the ImmerSpy model, the first guess accuracy increases by 14% and the top-2 and top-3 accuracies increase by 10% and 6% respectively, reaching up to 99%. The superior performance of ImmerSpy is attributed to its ability to leverage higher sampling rates and effectively utilize spectrogram-based image classification.

2) Uninformed Attacker Scenario: We evaluate the performance of ImmerSpy in the *uninformed* attacker scenario by cross-validating through selecting the test set from each six victim's data and averaging the outcomes. The results for the *uninformed* attacker scenario for ImmerSpy model are given in Table I. When the attacker lacks access to the victim's labeled data (*uninformed*), their accuracy drops by 14%. This is expected due to the model being entirely black-box.

We compare the performance of ImmerSpy in the *uninformed* attacker scenario using the five baseline models in Table III. ImmerSpy outperforms the SVM model by almost 10% with top-1 accuracy. With three guesses, ImmerSpy achieves 90% accuracy despite the model being entirely black box. Compared to the *informed* attacker, the STD of the accuracy increases in the *uninformed* attacker. This indicates that the model's performance in the *uninformed* scenario is highly dependent on the attacker's ability to generalize from other audio samples to accurately capture the victim's voice and pronunciation characteristics.

B. Robustness of ImmerSpy

1) Effect of the Window Size (RQ2): As discussed in Section V-B, selecting an appropriate window size is an aspect of data preprocessing that is essential to segment the large volume of collected data. Window sizes can vary widely: they may be arbitrary, aligned with the duration of each pronounced digit, or set to a random duration. For robustness analysis, we

TABLE III: Uninformed attacker top-3 accuracy results with per victim cross-validation. The model that yields the best accuracy is IMMERSPY, highlighted in the table.

Model	Top-1	Top-2	Top-3
Random Forest	0.542 ± 0.045	0.717 ± 0.044	0.833 ± 0.043
Decision Tree	0.405 ± 0.043	0.513 ± 0.044	0.605 ± 0.043
Logistic Regression	0.416 ± 0.078	0.578 ± 0.081	0.702 ± 0.075
KNN	0.347 ± 0.058	0.494 ± 0.069	0.632 ± 0.060
SVM	0.619 ± 0.036	0.754 ± 0.046	0.869 ± 0.04
IMMERSPY	0.715 ± 0.032	0.821 ± 0.031	0.902 ± 0.028

TABLE IV: Model accuracy for different window sizes.

Window Duration	Informed A.	Uninformed A.
Shortest word	0.723	0.619
Each word	0.734	0.623

TABLE V: Model accuracy for different number of pronunciations and speakers.

Attacker	No. of Pronounciations			No. of Speakers			
	10	25	50	16	8	4	1
Informed	0.842	0.844	0.848	0.848	0.854	0.866	0.879
Uninformed	0.642	0.660	0.762	0.792	0.715	0.443	-

experimented with window sizes tailored to the length of each digit to ensure accurate data labeling and to capture the entire duration of each spoken digit. However, this method proved impractical for real-world applications, as an attacker would unlikely know the exact duration of the digits spoken by a victim. For the evaluation, we employed the SVM model. This choice was made because using variable-length windows for spectrogram-based image classification in CNNs could cause the model to prioritize learning from the temporal dimension of the audio (i.e., Mel time bins) over the critical frequency characteristics of the signal.

We show our results in Table IV. Although using window sizes tailored to each digit slightly improves accuracy, the gains are minimal. Hence, we adopt a fixed 0.15 second window, approximately the duration of the shortest pronounced digit. This proved to be effective in accurately labeling the dataset.

2) Effect of the Training Set (RQ3): Training set size plays a crucial role in the performance of ML models. Because our dataset consists of motion sensor recordings while different audios are played, it is especially important to understand the effect of training set size on the performance of the model. Therefore we examined different scenarios: i) the effect of the number of pronunciation of each digit made by the speakers in the training set and ii) the effect of the number of users in the training set. By evaluating the model in each of these cases, we examine how robust IMMERSPY is.

Effect of Different Pronunciations: The performance of the model, assessed across varying numbers of pronunciations, is detailed in Table V for both the *informed* and *uninformed* attacker scenarios. We employed the CNN model due to its superior overall performance. To evaluate the impact of pronunciation variability, we randomly selected subsets of 10 and 25 pronunciations per digit from each user, in contrast to the complete set of 50 available in the dataset. The evaluation metrics show minimal variation across these subsets in the *informed* attacker scenario, suggesting that the model's predictive capability remains robust despite a reduction in pronunciation data. This indicates a resilience to variations in pronunciation from a single speaker. Conversely, in the *uninformed* attacker scenario, the variability in pronunciations has a more pronounced effect, demonstrating that access to a broader range of variations enhances the success of the attack when the attacker operates in a completely black-box manner.

Effect of the Number of Speakers: To evaluate how the diversity of speakers and the uniqueness of their tones impact the performance of IMMERSPY, we utilized a combined dataset consisting of six speakers from the FSDD dataset [31] and 10 speakers from the TIDIGITS dataset [25], resulting in a total of 16 speakers. For the 16-speaker case, all 16 speakers were included in the training set, except the victim in the *uninformed* attacker scenario. Additionally, we assessed the performance of IMMERSPY using a subset of eight speakers (the victim and seven randomly selected non-victim speakers), applying leave-one-out cross-validation across the victims. For the *informed* attacker scenario, we also conducted experiments using data exclusively from each victim (one-speaker case) for training and testing. However, the one-speaker case does not apply to the *uninformed* attacker scenario, as it assumes that the attacker lacks labeled data for training.

The results for both *informed* and *uninformed* attacker scenarios, with varying numbers of speakers, are detailed in Table V. In the *informed* attacker scenario, reducing the number of speakers in the dataset to focus more on the victim's data enhances the model's accuracy. However, if the attacker lacks sufficient labeled data from the victim, they may opt to include data from other speakers, as this does not significantly affect accuracies. This strategy allows attackers to supplement their dataset, potentially using GenAI or online audio data.

However, in the *uninformed* attacker scenario, the performance of the attack declines sharply as the number of speakers in the training set decreases. This shows that in situations where the attacker does not have prior knowledge of the victim, increasing the diversity of the data by including various speakers is advantageous. The specific number of speakers to include depends on what the attacker can infer about the victim; for instance, if the attacker can determine that the victim speaks English based on their location, they may focus on collecting English-speaking audio data. However, in a completely black-box scenario, it is ideal for the attacker to incorporate a wide range of accents, genders, and languages to maximize accuracy.

3) Effect of Different Parameters (RQ4): We conduct multiple experiments to understand how different parameters including volume, audio file format, and different headsets affect the performance of IMMERSPY.

Volume Level: We evaluate the performance of the attack when the device's audio volume is adjusted from maximum

TABLE VI: Effect of head movement removal with ImmerSpy.

Movements	Attacker	Without HMR	With HMR
Minor head	Informed	0.591	0.873
	Uninformed	0.434	0.728
Full-body	Informed	0.401	0.843
	Uninformed	0.360	0.702

to lower levels. Specifically, we configure the device volume level to 100%, 75%, 50%, and 25% and collect our data at these levels. The accuracy results for the *informed* attacker yield 86%, 78%, 63%, and 21%; and 71%, 62%, 52%, and 20% for the *uninformed* attacker, respectively. This gradual drop in accuracy is expected as lower audio volumes reduce the influence of speech signals on sensor readings.

Audio File Format: To understand how audio file formats impacts the performance of ImmerSpy, we tested our model against WAV and MP3 formats which are commonly used for uncompressed and compressed files. We noted a slight decrease in accuracy to 83% with MP3 formats compared to 86% with the WAV format. This is due to the lossy compression applied in MP3, which affects the volume dynamics.

Different Headset: To assess how ImmerSpy's performs across different devices, we conducted experiments on Meta Quest 3. Although it shares the same IMU chip as Quest 2, Quest 3 offers upgraded speakers and a more compact design, being about 40% smaller than Quest 2 [7]. This has led to an improvement in accuracy to 89%, due to louder sound output and reduced distance between speakers and sensors, which allow for more speech-related information to be captured. This indicates that advances in VR technology could expand the attack surface and expose devices to greater security risks.

4) Speech Detection and Head-associated Movements Removal (RQ5): We tested our proposed head-associated movement removal (HMR) across different scenarios. These scenarios included minor head movements, such as resting the head on a hand while watching a video, and full-body movements, such as shooting objects and walking. We compared the classification accuracies using both the original and HMR-filtered signals in both *informed* and *uninformed* attacker scenarios. The accuracy results, presented in Table VI, demonstrate that HMR significantly improves performance. Specifically, HMR improves the accuracy of digit classification by 29% for scenarios involving minor head movements. In scenarios requiring full-body movements, the improvement is even more pronounced, with an overall increase in accuracy of 39%.

Additionally, we analyzed scenarios where the application screen was positioned in front of or on the side of the user. When the screen was directly in front, head movements generally remained below 8 Hz. However, when the screen was positioned to the side, head movements reached up to 15Hz. These frequencies fall within the range of our proposed HMR threshold. Therefore, the effects of head movements due to screen position are effectively filtered out of the data used in the machine learning pipeline.

Furthermore, to evaluate the robustness of ImmerSpy in accurately detecting speech, we conducted a case study using Meta Quest 2 sensor data. These data were collected during periods of audio activity (where the user pronounced each digit, played through the HMD's speakers) and silence. Our model successfully identified 97% of the windows containing speech. Furthermore, speech detection achieved a precision, recall and F measure of 0.98, illustrating high accuracy and reliability in speech identification.

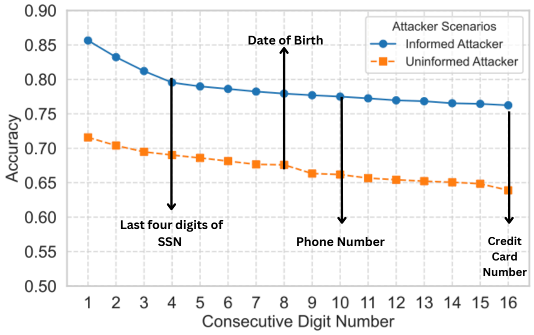


Fig. 9: Consecutive digit recognition accuracies.

C. Accuracy for Consecutive Digits Prediction (RQ6)

The ImmerSpy attack demonstrates robust performance in accurately identifying spoken digits across various scenarios, including changes in the number of speakers and pronunciation variations. To further assess the effectiveness of our proposed model and the features it extracts, we specifically examine the attack's ability to correctly identify consecutive digits. In realistic settings, an attacker can aim to accurately identify sequences of digits, such as SSNs, telephone numbers, or dates of birth. For example, if a victim articulates digits sequentially, such as when disclosing their phone number or the last four digits of their SSN, it becomes crucial to correctly predict every consecutive digit. A single error in these sequences can significantly alter the interpreted number, making the accuracy of successive digit identification critical.

In this study, we explore the capability of our attack model to predict consecutive numbers, analyzing sequences ranging from 2 to 16 digits. Four consecutive digits could be the last four digits of an SSN, eight could be the victim's DOB, ten digits could be their phone number, and sixteen digits could be their credit card. The accuracy for both attacker scenarios is given in Figure 9. Both models achieve more than 70% accuracy in predicting four consecutive digits and more than 65% for ten consecutive digits. These findings suggest that an attacker can successfully acquire sensitive information such as the victim's SSN, phone number, credit card details, and date of birth, even without prior knowledge of the victim.

D. Effect of Using GenAI

Table V shows that the number of users significantly affected the attack performance in the *uninformed* attacker scenario. This is because the victim's labeled data are not in the training set, whereas the victim's pronounciation of each digit might be

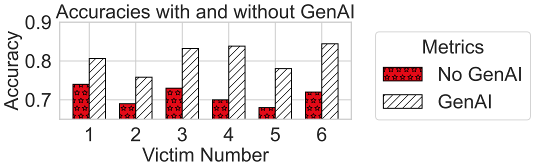


Fig. 10: The impact of GenAI on *uninformed* attacker.

different from the pronunciations made by those who constitute the training set. To maximize the effectiveness of the attack, the attacker can leverage GenAI to enrich the training set by incorporating AI-generated audio from synthetic speakers.

To achieve this, we leverage BARK [20], TTS [5], Amazon Polly AI [13] and Meta Audiobox [32], which are open-source GenAI text-to-speech conversion tools. We expand our training set by adding AI-generated voices that pronounce the digits from 0-9. To detail, we added 80 AI-generated voices (39 female, and 41 male) who are all English-speaking with different age groups and accents (i.e., British, Irish, Australian, South African). We intentionally do not consider AI-generated speakers as "victims" and only use their data to increase diversity in the training set.

Figure 10 shows the results of using GenAI for *uninformed* attackers. By enhancing the training set through GenAI, the *uninformed* attacker accuracy increases to 82% from 71%. Using GenAI, attackers can capture a wide range of vocal traits, such as accents, gender, and language, to build a comprehensive dataset, especially in cases where they have no prior knowledge of the victim, enabling a complete black-box attack. Incorporating audio from synthetic voices further enhances the accuracy of predictive models by allowing attackers to refine their datasets. The real strength of GenAI lies in its ability to generate these customized datasets, specifically tailored to improve attack outcomes. Our experiments confirm its effectiveness, showing that text-to-speech tools can significantly boost model accuracy.

Generally, GenAI can be further integrated to amplify an attacker's impact by enabling them to create or enhance datasets they would not otherwise have access to. Using text-to-speech models, attackers can generate data even for highly specific scenarios, filling gaps like missing audio in the victim's language. This flexibility allows attackers to build highly customized datasets, increasing the accuracy of identifying sensitive information about their targets.

E. Comparison with SOTA

To evaluate IMMERSPY against state-of-the-art (SOTA) methods, we selected recent works on extracting spoken digits from smartphone sensors, specifically Spearphone [23], AccelEve [26], and EarSpy [35]. We replicated and compared the performance of our model with these approaches.

The Spearphone model detects speech based solely on the STD changes along the z-axis. MFCCs and statistical features in the time domain are used with a random forest classifier. Initially, we evaluated Spearphone under its original conditions, where the VR device remained stationary, achieving 69% accuracy. However, when we introduced head movements into the set-up, performance significantly decreased to 27%, likely due to its inability to compensate for these movements. In contrast, IMMERSPY is designed to mitigate head movements by analyzing outliers on all axes, leading to superior performance. Under the same conditions, IMMERSPY achieved 89% accuracy utilizing Mel spectrograms for speech detection, further highlighting its robustness in dynamic environments.

Using AccelEve's pipeline, we achieved 72% accuracy, compared to IMMERSPY's 86%. The lower performance of AccelEve's model is likely due to its reliance on STFT spectrograms, which may not capture the full 1000 Hz spectrum as effectively as Mel-spectrograms do. By converting data into decibels through logarithmic operations, Mel-spectrograms are better suited for capturing subtle frequency variations and extracting more meaningful features for speech analysis. Lastly, the EarSpy pipeline yielded an accuracy of 29%, as it does not remove silent segments and head movements from the data. These comparisons demonstrate that the pipeline we propose for VR devices in IMMERSPY effectively leverages the 1000 Hz sampling rate to capture spoken content, addressing the challenges associated with speech detection in VR.

VII. COUNTERMEASURE

A. Motivation

The increasing sensitivity and precision of motion sensors in high-end smart devices, particularly VR headsets, make acoustic side-channel attacks very successful. In attacks like IMMERSPY, attackers exploit how these precise motion sensors can misinterpret acoustic vibrations from speakers as motion signals, causing significant privacy risks.

In benign scenarios, high-frequency data collection from these motion sensors, operating at rates such as 1000 Hz, is critical. For example, the Meta Move app tracks user movements and estimates calorie expenditure by continuously monitoring sensor data at high sampling rates, ensuring the accuracy needed for real-time fitness tracking. Previous research has proposed reducing sensor sampling rates to mitigate the risk of speech leakage, but this solution has proven only partially effective. Even with reduced rates, such attacks remain feasible [21]. Additionally, VR systems, which require 6DoF to capture precise rotational and positional movements, cannot afford to compromise sensor precision, unlike smartphones that use simpler 3DoF tracking. Consequently, alternative defensive strategies are needed.

As a novel countermeasure, we propose to take advantage of the inherent vulnerabilities of IMMERSPY by introducing noise into sensor data using inaudible frequencies emitted through the speakers of the HMD. These inaudible sounds induce minor, yet strategic, fluctuations in accelerometer readings. We recommend continuously varying the frequency of these sounds to prevent attackers from adapting and filtering out this noise. By sweeping through a spectrum of frequencies, we create a dynamic defense that resembles the tactics used in advanced electronic warfare technologies designed to obfuscate and protect sensitive information from adversaries [6].

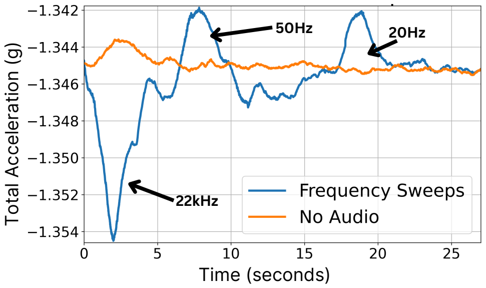


Fig. 11: Total sum of x, y, and z axes of the accelerometer during frequency sweeping.

TABLE VII: Performance of the proposed defense solution.

Attacker	Without Defense	With Defense
Informed	0.856	0.093
Uninformed	0.715	0.096

B. Feasibility and Evaluation

We investigate the impact of emitting pure frequencies through the HMD's speakers on accelerometer data. For this, we played varying frequencies ranging from 10 Hz to 50 Hz and from 20 kHz to 24 kHz. Because the speakers of Meta Quest 2 have a sampling rate of 48 kHz, the frequencies above 24 kHz cannot be accurately reproduced by the on-device speakers. Therefore, in our experiments, we limited our pure-tone frequency generation to between 10-50 Hz and 20-24 kHz. Our application randomly picks a value between this interval and plays that pure frequency as audio.

To assess the impact on the accelerometer, we plot the summed positional acceleration measurements across the x, y, and z axes of a stationary device. This analysis helps us understand how the device responds to pure inaudible frequency emissions compared to periods of complete silence, where only sensor noise is present. The results of this case study, illustrated in Figure 11, demonstrate that the inaudible frequencies emitted significantly affect the device's sensor readings, inducing random fluctuations throughout the experiment.

We then evaluated the performance of our defense with an experiment where audio of users pronouncing digits was played through speakers, superimposed with pure frequency tones ranging from 10 Hz to 50 Hz and 20 kHz to 24 kHz. Table VII summarizes the results of this defense solution. The proposed defense significantly reduces the accuracy of estimating the pronounced digits, even with access to prior data about the victim. This impact is particularly increased in the *informed* attacker scenario, where the defense effectively neutralizes the attacker's advantage of having prior data.

C. Usability of the Defense Solution

We evaluated the usability, safety, and functionality of the proposed defense method. First, we examined the effect of emitting inaudible pure tone frequencies on the HMD's gaze tracking while stationary. This experiment assessed whether these frequencies could cause user discomfort by altering gaze movements, as detected by the HMD's accelerometer. We measured the STD of sensor data across different frequency ranges, as shown in Table VIII. The range of 0-50 Hz represents low inaudible frequencies, 20-24 kHz covers high inaudible frequencies, and 100-5000 Hz captures the human speech spectrum. These show that employing inaudible sounds did not introduce significant noise to the sensors beyond what is typically observed when playing speech audio. Hence, this approach will not cause discomfort for the user.

TABLE VIII: Average std. dev. across frequency ranges.

Frequency (Hz)	x	y	z
0-50	2.5e-4	1.3e-4	8.0e-5
100-5000	3.1e-4	5.7e-4	6.6e-4
20k-24k	8.0e-5	4.0e-5	4.8e-4

The decibel levels of the emitted tones remained between 30-40 dB. The OSHA and NIOSH set the permissible exposure to noise at 90 dB for up to 8 Hrs/day to prevent hearing damage [33]. Studies also confirm that exposure to low or non-audible frequencies does not cause hearing loss when kept below 120 dB [53]. Thus, our defense method is unlikely to cause hearing damage.

Additionally, we assessed the impact on application performance by evaluating key metrics such as CPU, and memory using Oculus Developer Hub. Performance metrics were recorded in the foreground application without background audio, and in the foreground application with background audio playing. When high-pitched audio was played in the background, CPU utilization increased by 1%, and total audio memory usage was found to be around 1.3MB. This minimal effect is expected, as the user is already interacting with the audio content in the foreground application. To enhance the practicality of our defense solution, the foreground app designed to mask sensitive content leakage from sensors could selectively insert inaudible sounds during sensitive speech by monitoring the speech through the microphone.

D. Potential Counter Defenses

The proposed defense injects low and ultrasonic frequency noise into the accelerometer to mask sensor signals that could be exploited for extracting sensitive data. Attackers may attempt to counter this with techniques like principal component analysis (PCA), filters, autoencoders, or digital signal processing (DSP) techniques to remove the noise. While PCA can eliminate low-variance noise, its effectiveness is limited due to varying frequency ranges and the complexity of multi-dimensional signals. Furthermore, filtering techniques will also be ineffective due to the overlap of the speech signal and aliased sensor readings with the inaudible frequency range. Although autoencoders and DSP techniques like adaptive filtering could reduce noise, they require extensive training or precise knowledge of jamming signals and are computationally intensive. Thus, each counter-defense method faces inherent challenges.

E. Other Potential Countermeasures

Isolating Speakers. With the VR HMDs getting increasingly compact with each version manufactured by the companies, the impact of acoustic side-channel attacks will increase even more. The challenge is preventing these privacy issues without limiting the sensors' capabilities. One method to prevent acoustic side-channel attacks is to isolate the speakers from the motion sensors. As shown in Figure 1, the motherboard that contains the motion sensors is closely packed with the on-device speakers. To reduce the transmission of vibrations from the speakers, materials for anti-vibration can be padded to act as a physical barrier between the loudspeaker and the motion sensors. Similarly, the motion sensors could be covered with materials that would absorb sound waves to prevent the acoustic signals from reaching the sensors.

Enforcing via the Privacy Policy. Currently, Meta enforces a set of privacy requirements that app developers must comply with regarding sensor data collection. In these policies, they state that for the purpose of maintaining and improving the content, app developers can collect and process sensor data if and only if the user cannot be identified or deanonymized through the collected data [4]. Meta also has Privacy Tabs embedded in each app where the user can see which sensors are enabled and which sensor data are being processed by these apps [2]. However, due to motion sensors being "zero-permission" sensors, these devices are not displayed on the Privacy Tab of the application. Therefore, it becomes challenging to identify a widely used sensor across possibly all applications as acting maliciously and enforce policies about the sensor usage.

VIII. Discussion

A. Limitations and Future Research Directions

Our work achieves high accuracy even with off-the-shelf ML models for both attacker scenarios. Even in the *uninformed* attacker scenario, the accuracy remains comparable to that achieved with the training set from the victim. However, the current ImmerSpy attack operates as a restricted library attack, requiring the attacker to train their model on specific words to decode speech. This limitation suggests that in free speech scenarios, ImmerSpy's performance may decrease, as its training set only includes digit pronunciations. Recent studies suggest that training models on vowels could expand this to unconstrained speech reconstruction [30]. Therefore, future work should enhance ImmerSpy to facilitate free speech extraction, leveraging GenAI to develop robust models capable of operating in fully black-box scenarios, even when the attacker's primary language is unknown. Testing across different languages will further evaluate its efficacy in diverse eavesdropping contexts.

Another future research direction involves testing the proposed defense solution on other IoT devices susceptible to acoustic eavesdropping attacks. One potential approach is to utilize frequency levels that closely match the audio output from the device's speakers, thereby making it difficult for users to distinguish between the defense solution's operation and the normal audio played.

B. Future Implications of ImmerSpy

As with any advancing technology, increasing its integration into daily life requires making it lighter and user-friendly. Just as telephones evolved from bulky models to compact, pocket-sized devices, similar trends are expected in the rapidly developing field of XR. As these devices become smaller, the attack surface will become more critical. The proximity of on-device speakers to IMU sensors will likely reduce, increasing the sensors' sensitivity to audio signals. Additionally, as devices' immersiveness enhances, the speakers will get even more powerful to deliver 3D spatial audio which will further intensify the effect of sound waves on sensors.

Furthermore, advances in AI have introduced new threat vectors, including the use of text-to-speech tools to generate custom audio samples. Attackers can tailor these models to match specific profiles, such as languages, accents, or emotional tones, making it easier to detect these attributes in speech. AI can also create deepfake audio of public figures, like fabricating a president's voice, to simulate an *informed* attacker scenario, even without access to real labeled data.

IX. Related Work

Zero-permission Sensor Attacks in VR Devices: Zero-permission sensors on VR devices have started to be tackled recently in the academic literature, from keylogging attacks to individual de-anonymization attacks. In one study, the orientation angles of the controllers were collected through a malicious app to understand where a key click occurs and retrieve the user password [34]. Similarly, Slocum et al. [52] use HMD motion sensors to estimate the direction of user gazes and extract their passwords. Nair et al. [39] analyzed user hand and body movements during a VR game to deanonymize more than 50,000 users, and Tricomi et al. [54] showed that even the gait pattern of users can tell them apart. Furthermore, Face-Mic attack [48] captured speech-associated facial movements of the users through HMD's motion sensors and identified speech content and gender of users, and FaceReader attack extracted viral signs to infer gender and body fat information of the users. In contrast to these studies, our method ImmerSpy explores the effect of machine-rendered speech on IMU sensors, leveraging audio signals from on-device speakers rather than speech-induced facial movements or gait patterns, thus expanding the scope of VR sensor exploitation.

Speech Extraction from Smartphone Zero-permission Sensors: Michalevsky and Boneh [37] first analyzed audio signals' effect on motion sensor recordings, using external loudspeakers and smartphones placed on the same surface. Anand et al. [22] expanded this by investigating machine-generated speech through surface and conductive vibrations. Spearphone [23] later demonstrated that on-device speakers can create acoustic side-channel attacks, with speech vibrations that significantly affect smartphone sensors. AccelEve [26] showed that adult speech could be fully captured with sampling frequencies up to 500 Hz, while AccEar [30] used deep learning to reconstruct audio signals from motion sensors. EarSpy [35] further proved that eavesdropping through ear speakers was feasible. In this paper, we exploit motion sensors to eavesdrop on user conversations on VR devices and examine two attacker models, one with access to the victim's labeled data and another operating in a black-box scenario. To our knowledge, this is the first study to employ GenAI to generate extensive datasets tailored to attackers' objectives. We also explore consecutive-digit prediction, an overlooked scenario in prior research, and propose a novel defense mechanism that significantly reduces attacker performance while maintaining the VR system's usability and practicality.

X. Conclusion

We present ImmerSpy, a novel side channel leakage through motion sensors in VR devices. ImmerSpy successfully captures spoken digits with high precision even with entirely black-box scenarios. Using GenAI and text-to-speech models, ImmerSpy creates diverse training data, enhancing its success rate without the need for victim-specific data. Our evaluations demonstrate ImmerSpy's effectiveness in both *informed* and *uninformed* attack models, even during realistic VR user movements. In addition, we propose a defense mechanism that emits pure frequencies from speakers to obscure sensor data, which proves effective in mitigating this attack.

Acknowledgment

We thank the anonymous reviewers for their helpful feedback and time. This work has been partially supported by the US National Science Foundation's (NSF) Intergovernmental Personnel Act Independent Research & Development Program, NSF grant CNS-2144645, the US National Security Agency (Award: H982302110324), Florida International University Graduate School, Cyber Florida, Google ASPIRE Program, Microsoft, La Sapienza University of Rome within the 2023 Bando Ricerca project "METAVERSEc: Securing Metaverse and eXtended Reality", the European Commission under the Horizon Europe Programme project LAZARUS, project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU, and the Italian Electrical System under the Contract Agreement between ENEA, the Ministry of the Environment and Energetic Safety – Project 2.1 "Accordo di Programma 2022–2024". The views expressed are those of the authors only, not of the funding agencies.

References

[1] "Track your vr fitness stats with the oculus mobile app or apple health," https://about.fb.com/news/2022/06/track-your-vr-fitness-stats-with-the-oculus-mobile-app-or-apple-health/, 2022, [Accessed: 2023-09-10].

[2] "App privacy tab on meta quest store for increased transparency," https://www.meta.com/blog/quest/app-privacy-tab-meta-quest-store-increased-transparency/, 2023, [Accessed: 2023-12-02].

[3] "Core motion," https://developer.apple.com/documentation/coremotion, 2023, [Accessed: 2024-05-01].

[4] "Data use policy," https://developer.oculus.com/policy/data-use/, 2023, [Accessed: 2023-12-02].

[5] "Deep learning for text to speech," https://github.com/mozilla/TTS, 2023, [Accessed: 2023-12-05].

[6] "An introduction to jammers," https://jemengineering.com/blog-an-introduction-to-jammers/, 2023, [Accessed: 2023-04-29].

[7] "Meta quest 3 review: Impressive hardware searching for a killer app," https://www.tomshardware.com/reviews/meta-quest-3, 2023, [Accessed: 2023-09-30].

[8] "Ovation vr: Public speaking training in virtual reality," https://www.ovationvr.com/, 2023, [Accessed: 2024-09-28].

[9] "Permissions on android," https://developer.android.com/guide/topics/permissions/overview, 2023, [Accessed: 2024-05-01].

[10] "Security 2: Best practices for oculus quest," https://developer.oculus.com/resources/vrc-quest-security-2/, 2023, [Accessed: 2024-07-07].

[11] "Vr development in unity," https://docs.unity3d.com/Manual/VROverview.html, 2023, [Accessed: 2024-04-24].

[12] "Welcome to steam," https://store.steampowered.com/, 2023, [Accessed: 2024-01-20].

[13] "Amazon polly," https://aws.amazon.com/tr/polly/, 2024, [Accessed: 2024-05-01].

[14] "Foreground services in android 11," https://developer.android.com/about/versions/11/privacy/foreground-services?hl=tr, 2024, [Accessed: 2024-10-02].

[15] "Meta horizon workrooms," https://forwork.meta.com/horizon-workrooms/, 2024, [Accessed: 2024-09-28].

[16] "Meta quest vr games, apps, deals and more," https://www.meta.com/experiences/view/777072216186618/, 2024, [Accessed: 2024-04-24].

[17] "Unity learn vr development pathway," https://learn.unity.com/learn/pathway/vr-development, 2024, [Accessed: 2024-04-24].

[18] "Unreal engine vr documentation," https://docs.unrealengine.com/4.26/en-US/SharingAndReleasing/XRDevelopment/VR/, 2024, [Accessed: 2024-04-24].

[19] R. M. Aburas, "User-centered data access control techniques for secure and privacy-aware mobile systems," Ph.D. dissertation, Purdue University, 2024.

[20] S. AI, "Text-prompted generative audio model," https://github.com/suno-ai/bark, 2023, accessed: 2023-12-05.

[21] A. Al-Haiqi, M. Ismail, and R. Nordin, "On the best sensor for keystrokes inference attack on android," *Procedia Technology*, 2013.

[22] S. A. Anand and N. Saxena, "Speechless: Analyzing the threat to speech privacy from smartphone motion sensors," in *IEEE Symposium on Security and Privacy (SP)*, 2018.

[23] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "Spearphone: a lightweight speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021.

[24] J. Ancilin and A. Milton, "Improved speech emotion recognition with mel frequency magnitude coefficient," *Applied Acoustics*, 2021.

[25] "Clean digits," http://www.ee.columbia.edu/~dpwe/sounds/tidigits/, 2016, [Accessed: 2024-09-26].

[26] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-based practical smartphone eavesdropping with built-in accelerometer." in *Network and Distributed System Security Symposium (NDSS)*, 2020.

[27] D. Cayir, A. Acar, R. Lazzeretti, M. Angelini, M. Conti, and S. Uluagac, "Augmenting security and privacy in the virtual realm: An analysis of extended reality devices," *IEEE Security & Privacy*, 2023.

[28] H. Farrukh, R. Mohamed, A. Nare, A. Bianchi, and Z. B. Celik, "{LocIn}: Inferring semantic location from spatial maps in mixed reality," in *USENIX Security Symposium*, 2023.

[29] H. Farrukh, T. Yang, H. Xu, Y. Yin, H. Wang, and Z. B. Celik, "S3: Side-channel attack on stylus pencil through sensors," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2021.

[30] P. Hu, H. Zhuang, P. S. Santhalingam, R. Spolaor, P. Pathak, G. Zhang, and X. Cheng, "AccEar: Accelerometer Acoustic Eavesdropping with Unconstrained Vocabulary," in *IEEE Symposium on Security and Privacy (SP)*, 2022.

[31] Z. Jackson, "Free spoken digit dataset (fsdd)," *Technical report*, 2016.

[32] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, and W.-N. Hsu, "Voicebox: Text-guided multilingual universal speech generation at scale," in *Advances in Neural Information Processing Systems*, 2023.

[33] A. Lie, M. Skogstad, H. A. Johannessen, T. Tynes, I. S. Mehlum, K.-C. Nordby, B. Engdahl, and K. Tambs, "Occupational noise exposure and

hearing: a systematic review," *International archives of occupational and environmental health*, vol. 89, pp. 351–372, 2016.

[34] Z. Ling, Z. Li, C. Chen, J. Luo, W. Yu, and X. Fu, "I know what you enter on gear vr," in *IEEE Conference on Communications and Network Security (CNS)*, 2019.

[35] A. T. Mahdad, C. Shi, Z. Ye, T. Zhao, Y. Wang, Y. Chen, and N. Saxena, "Earspy: Spying caller speech and identity through tiny vibrations of smartphone ear speakers," *arXiv preprint arXiv:2212.12151*, 2022.

[36] Y. Mekdad, F. Naseem, A. Aris, H. Oz, A. Acar, L. Babun, S. Uluagac, G. S. Tuncay, and N. Ghani, "On the robustness of image-based malware detection against adversarial attacks," in *Network Security Empowered by Artificial Intelligence*. Springer, 2024, pp. 355–375.

[37] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *USENIX Security Symposium*, 2014.

[38] R. Mohamed, H. Farrukh, Y. Lu, H. Wang, and Z. B. Celik, "iStelan: Disclosing Sensitive User Information by Mobile Magnetometer from Finger Touches," in *Privacy Enhancing Technologies (PoPETs)*, 2023.

[39] V. Nair, W. Guo, J. Mattern, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song, "Unique identification of 50,000+ virtual reality users from head & hand motion data," *arXiv preprint arXiv:2302.08927*, 2023.

[40] V. Nair, C. Rack, W. Guo, R. Wang, S. Li, B. Huang, A. Cull, J. F. O'Brien, L. Rosenberg, and D. Song, "Inferring private personal attributes of virtual reality users from head and hand motion data," *arXiv preprint arXiv:2305.19198*, 2023.

[41] H. Oz, A. Acar, A. Aris, G. S. Tuncay, A. Kharraz, and S. Uluagac, "(in)security of file uploads in node.js," in *ACM Web Conference 2024*, 2024.

[42] H. Oz, A. Aris, A. Acar, G. S. Tuncay, L. Babun, and S. Uluagac, "{RøB}: Ransomware over Modern Web Browsers," in *USENIX Security Symposium*, 2023.

[43] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A survey on ransomware: Evolution, taxonomy, and defense solutions," *ACM Computing Surveys (CSUR)*, 2022.

[44] H. Oz, D. C. D'Elia, G. S. Tuncay, A. Acar, R. Lazzeretti, and S. Uluagac, "With great power comes great responsibility: Security and privacy issues of modern browser application programming interfaces," in *IEEE Security & Privacy*, 2024.

[45] Precedence Research, "Immersive technology market size, share, and trends," https://www.precedenceresearch.com/immersive-technology-market, 2023, accessed: 2023-11-17.

[46] M. Quest, "Learn about meta quest move," https://www.meta.com/help/quest/articles/in-vr-experiences/oculus-apps/learn-about-move/, 2024, accessed: 2024-10-02.

[47] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018.

[48] C. Shi, X. Xu, T. Zhang, P. Walker, Y. Wu, J. Liu, N. Saxena, Y. Chen, and J. Yu, "Face-mic: inferring live speech and speaker identity via subtle facial dynamics captured by ar/vr motion sensors," in *ACM Conference on Mobile Computing and Networking*, 2021.

[49] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thsense: A context-aware sensor-based attack detector for smart devices," in *USENIX Security Symposium*, 2017.

[50] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, "A Survey on Sensor-based Threats and Attacks to Smart Devices and Applications," *IEEE Communications Surveys and Tutorials*, 2021.

[51] P. Singh, N. Juneja, and S. Kapoor, "Using mobile phone sensors to detect driving behavior," in *ACM Symposium on Computing for Development*, 2013.

[52] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen, "Going through the motions:{AR/VR} keylogging from user head motions," in *USENIX Security Symposium*, 2023.

[53] B. Smagowska, "Effects of ultrasonic noise on the human body—a bibliographic review," *International Journal of Occupational Safety and Ergonomics (JOSE)*, 2013.

[54] P. P. Tricomi, F. Nenna, L. Pajola, M. Conti, and L. Gamberi, "You can't hide behind your headset: User profiling in augmented and virtual reality," *IEEE Access*, 2023.

[55] T. Zhang, Z. Ye, A. T. Mahdad, M. M. R. R. Akanda, C. Shi, Y. Wang, N. Saxena, and Y. Chen, "Facereader: Unobtrusively mining vital signs and vital sign embedded sensitive info via ar/vr motion sensors," in *ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[56] Y. Zhang, C. Slocum, J. Chen, and N. Abu-Ghazaleh, "It's all in your head(set): Side-channel attacks on AR/VR systems," in *USENIX Security Symposium*, 2023.