Eavesdropping on Controller Acoustic Emanation for Keystroke Inference Attack in Virtual Reality

Shiqing Luo*, Anh Nguyen*, Hafsa Farooq†, Kun Sun*, Zhisheng Yan*
*George Mason University, Email: {sluo5, anguy59, ksun3, zyan4}@gmu.edu
†Georgia State University, Email: hfarooq5@student.gsu.edu

***Abstract*—Understanding the vulnerability of virtual reality (VR) is crucial for protecting sensitive data and building user trust in VR ecosystems. Previous attacks have demonstrated the feasibility of inferring VR keystrokes inside head-mounted displays (HMDs) by recording side-channel signals generated during user-HMD interactions. However, these attacks are heavily constrained by the physical layout or victim pose in the attack scenario since the recording device must be strictly positioned and oriented in a particular way with respect to the victim. In this paper, we unveil a placement-flexible keystroke inference attack in VR by eavesdropping the clicking sounds of the moving hand controller during keystrokes. The malicious recording smartphone can be placed anywhere surrounding the victim, making the attack more flexible and practical to deploy in VR environments. As the first acoustic attack in VR, our system, `Heimdall`, overcomes unique challenges unaddressed by previous acoustic attacks on physical keyboards and touchscreens. These challenges include differentiating sounds in a 3D space, adaptive mapping between keystroke sound and key in varying recording placement, and handling occasional hand rotations. Experiments with 30 participants show that `Heimdall` achieves key inference accuracy of 96.51% and top-5 accuracy of 85.14%–91.22% for inferring passwords with 4–8 characters. `Heimdall` is also robust under various practical impacts such as smartphone-user placement, attack environments, hardware models, and victim conditions.**

I. Introduction

Virtual reality (VR) is a highly promising computing platform with 171 million users [8]. It has enabled diverse applications, including gaming, healthcare, and training. However, due to its collection and storage of extensive personal data, VR poses unique privacy challenges. Leaks of biometrics, passwords, credit cards, or intimate behavioral data can result in identity theft and unauthorized access. Therefore, it is crucial to devise new measures to safeguard VR data.

Most of the existing approaches to protecting VR data focus on knowledge- or biometrics-based authentication schemes [17], [29], [47], [36]. These schemes aim to prevent unauthorized access to VR data when an attacker physically possesses the VR head-mounted display (HMD). The underlying assumption is that by securing the HMD, VR data will also be protected. However, recent studies have shown that the interaction between a user and an HMD creates various side channels that are completely exposed to the public. Attackers can record these side channel signals and exploit their subtle relationship with VR keystrokes to infer sensitive input, even without physical possession of the HMD. For example, by analyzing video recordings of a user [31], [40], [69] or variations in surrounding wireless signals [1] during a VR input session, attackers have successfully inferred keystrokes inside the HMD. Notably, the threat posed by these recording-based side channel attacks is significantly higher in VR environments than in traditional desktop or mobile environments due to the blocking of users' real-world visuals by HMDs and the decrease in situational awareness [20].

While these initial attacks demonstrate the feasibility of recording user-HMD interactions and inferring VR keystrokes, they are hindered by practical limitations. All these attacks are heavily constrained by the physical layout or victim pose in the attack scenario because the recording devices must be placed in a certain orientation and position with respect to the victim. For instance, in the video-based attack [31], [40], [69], a camera must be positioned directly in front of the victim to capture finger clicking and controller moving comprehensively, i.e., the camera must always maintain line-of-sight views of the controller. If the camera is shifted to a side position and the finger or controller becomes obscured, even partially, by body movement, the attack would fail. Similarly, the wireless signal based attack [1] requires the deliberate placement of a transmitter and a receiver such that the victim lies within a direct well-aligned link between the two malicious devices. Hence, users can effectively counter these *placement-sensitive* attacks by positioning themselves in a secure layout, such as sitting in a corner.

In this paper, we expose a *placement-flexible* VR keystroke inference attack named `Heimdall`, which leverages a novel side channel created by the acoustic emanation from the handheld VR controller during user-VR interactions. `Heimdall` targets the single hand controller that is universally supported across VR HMDs from Samsung, Google, HTC, and Oculus [57], [22], [24], [42]. When a user navigates through the virtual keyboard by moving the controller and commits a keystroke by clicking the controller, a distinct clicking sound is emitted and recorded by a nearby malicious smartphone. By analyzing the spatial relationship between the moving controller and the fixed smartphone, the sound of each keystroke can be uniquely defined and used to infer the corresponding key entered inside the HMD. Since the clicking sound can be recorded anywhere surrounding the victim, the malicious smartphone can be placed in any position and orientation without strict requirements. This includes non-line-of-sight scenarios where the smartphone has partial or no visibility of

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.24100
www.ndss-symposium.org

the controller, e.g., behind the victim. Consequently, Heimdall is more flexible with the layout or furniture near the victim and the victim's poses, offering attackers more options in choosing when, where, and how they record the keystroke sounds and posing greater threats in practice.

Challenges and Approaches. Although previous studies have analyzed acoustic emanations from physical keyboards and smartphone touchscreens to infer keystrokes [32], [72], [73], [34], [13], no research has exploited the acoustic side channels of VR devices that present a distinct interaction modality. Therefore, realizing Heimdall requires overcoming new challenges. First, when analyzing keystrokes on physical keyboards, the time difference of arrival of keystroke sounds can be recorded by omnidirectional microphones on the same 2D surface to differentiate them [73], [32]. However, keystroke sounds of the VR controller can come from anywhere in 3D space, so the omnidirectional signal and its phase are no longer sufficient to differentiate keystroke sounds in VR. To solve this problem, we propose a directional acoustic signal acquisition framework. We use porous plastic tubes to convert the omnidirectional microphones on a smartphone to directional microphones, allowing VR keystroke sounds from different 3D positions to create directional signals with uniquely-distorted phases and magnitudes for differentiation, similar to how 3D sounds are perceived by human ears via ear canals.

Second, unlike traditional acoustic keystroke attacks that place microphones on or next to the victim device and assume a minimal impact of placement position and orientation [32], [72], the relative position and orientation between the user and the microphones in VR can vary significantly in different attack scenarios. While we can record a directional keystroke sound and map its distinct direction-of-arrival (DOA) to the respective key in a given microphone-user placement, the same key's keystroke sound and DOA will change in a new placement. To overcome this challenge, we propose an adaptive DOA-Key mapping scheme that adjusts the DOA-Key mapping obtained in a baseline placement to the actual placement in the attack. The scheme iteratively applies translation and rotation to the DOAs in a baseline mapping until a subset of them matches the DOAs of victim keystroke sounds in the attack, thereby utilizing the updated DOA-Key mapping for keystroke inference.

Third, compared with traditional devices where a keystroke sound is emitted from the key that is pressed, VR systems generate monotonous keystroke sounds from a single handheld controller. As a VR user may navigate the virtual keyboard by rotating their wrist without significant hand movements, the acoustic signals with the same DOA can introduce different keystrokes. To remedy this issue, we devise an inter-key relation-based calibration model that corrects mapping errors caused by the mismatch between rotated hands and unchanged DOAs. The rationale is that while the sound source locations of two VR keystrokes may be close, the time interval and controller moving direction between the two keystrokes are distinct depending on the distance between the virtual keys. We accordingly generate a list of VR keystroke candidates to improve the attack performance.

Evaluations. We validate the inference accuracy and robustness of Heimdall through extensive evaluations of 30 participants using 45 passwords, three types of VR HMDs, and two types of smartphones. The results show that Heimdall reaches key inference accuracy of 96.51%. When hacking a password with 4–8 characters, Heimdall can derive the exact password in five attempts with accuracy from 85.14% to 91.22%. The robustness study demonstrates that Heimdall maintains a satisfactory performance under various practical impacts such as smartphone-user placement, attack environments, hardware models, and victim conditions.

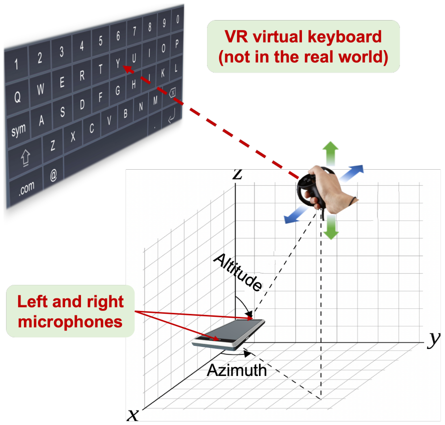


Fig. 1: The virtual keyboard, controller, and smartphone in a keystroke session.

Ethical Considerations. All experiments involving human subjects in this study have been approved by the IRB. We have only used Heimdall to perform attacks on the datasets described in this paper. Heimdall has never been used in other ways or released to other parties.

Contributions. The contributions of this paper include

- A directional acoustic signal acquisition framework that records differentiable VR keystroke sounds in 3D space via customized phone microphones (§V).
- An adaptive DOA-Key mapping scheme that adapts DOAs of keystroke sounds in the baseline placement to the attack case for keystroke inference (§VI).
- An inter-key relation-based calibration model to correct mapping errors caused by the mismatch between rotated hands and unchanged DOAs (§VII).
- An extensive evaluation of inference accuracy and its robustness under practical scenarios (§VIII and §IX).

II. Background and Challenges

A. User Behavior during VR Keystroke

As illustrated in Figure 1, VR devices employ a controller-based keystroke method. When keystrokes are needed, a virtual "QWERTY" keyboard is rendered in front of a user inside the HMD. During a sensitive keystroke session, the user continuously enters a sequence of keys using this keyboard, e.g., a password or an answer to a security question. The user stays at the same physical location and moves the controller around to complete the keystrokes. The VR operating system (OS) tracks the controller's position. When the controller is

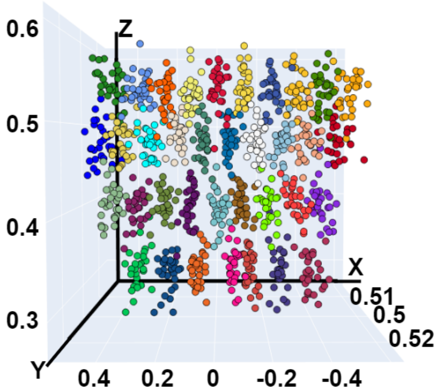


Fig. 2: The controller positions differ between keys, but cluster across users.

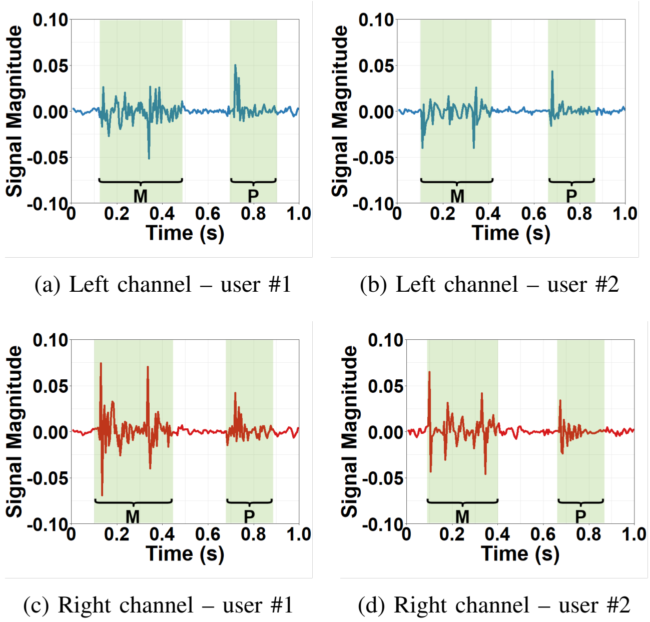
aimed at the target virtual key, the user clicks the confirm button on the controller and commits a keystroke.

Throughout this process, the user may slightly move her head and body for navigation, but it should be emphasized that the virtual keyboard remains fixed in front of the user, regardless of these movements. The VR OS only re-renders the virtual keyboard under large-scale movements when the original keyboard is completely out of the user's view, e.g., the user rotating her body 90° to the right or walking around the room. Although such movements are possible while exploring a VR App, they rarely occur in a sensitive keystroke session. In fact, a stationary position can reduce motion sickness while focusing on an object [59]. In this paper, we focus on a typical session without irregular behavior that commits a keystroke and then moves around to switch the view of the keyboard for subsequent keystrokes.

B. Correlation Between Key and Sound

The acoustic signal emitted from the controller upon a keystroke is correlated to the respective key entered in VR. As shown in Figure 1, when the user navigates the cursor through the virtual keyboard, the controller moves up/down and left/right in a plane approximately parallel to the keyboard. Given the malicious smartphone fixed in the 3D space and the origin point set at the center of the smartphone, keystroke sounds generated at different azimuths and altitudes can be recorded and associated with respective keys.

Moreover, the correlations between keystroke sounds and respective keys are similar across users [41], [47]. Under a recording placement with a given relative position and orientation between smartphone and user, different users generate similar keystroke sounds when committing the same key. This is because VR OS synchronizes the VR cursor movement with the controller movement. To move from one key to the other with a fixed distance on the virtual keyboard, different users need to move their hands with the same distance in the physical world. Thus keystroke sounds of the same key from different users are generated at similar 3D positions. Therefore, if the smartphone-user placement is consistent, attackers can pre-collect keystroke sounds of all target keys as a baseline and utilize this baseline mapping to infer the victim's keystrokes.



(a) Left channel – user #1
(b) Left channel – user #2
(c) Right channel – user #1
(d) Right channel – user #2

Fig. 3: VR keystroke sounds are distinct for keys "M" and "P", but remain consistent between two users.

Our study with 30 users validated the correlation between a keystroke sound and its respective key as well as the cross-user similarity of this correlation. Figure 2 depicts the controller 3D positions of 30 users (one circle per user) while typing 26 English letters and 10 digits (one color per key). We observe that the controller position is constrained to a small area for each key, incurring the unique keystroke sound. The positions of different users are closely clustered, as verified by a Silhouette Score of 0.64 via the K-means analysis (with a score of 1.0 indicating all samples are within their clusters) [52]. While nearby keys might share similar sound sources, the keystroke sound can be mapped to the closest cluster to identify the most probable key. Figure 3 further exemplifies two keystroke sounds "M" and "P" typed by two users, including the acoustic signals of both left and right channels recorded by the proposed directional microphones (§V). We observe that different keystroke sounds demonstrate distinguishable shapes, whereas the shapes of the acoustic signals from the same keystroke are consistent between different users.

C. Challenges

While the principle mentioned above is straightforward, there are practical challenges that hinder the realization of the attack. First, modern smartphones are only equipped with omnidirectional microphones and cannot record the differentiable directional signals illustrated in Figure 3. Second, the smartphone-user placement in the attack may be different from the placement when attackers pre-collect keystroke sounds, invalidating the mapping between a keystroke sound and a key. Third, sometimes different keystrokes may be generated at similar controller positions (overlapping clusters in Figure 2) because user hand rotation may lead to insignificant hand translation movement, causing mapping errors. In this paper, we strive to overcome these challenges.

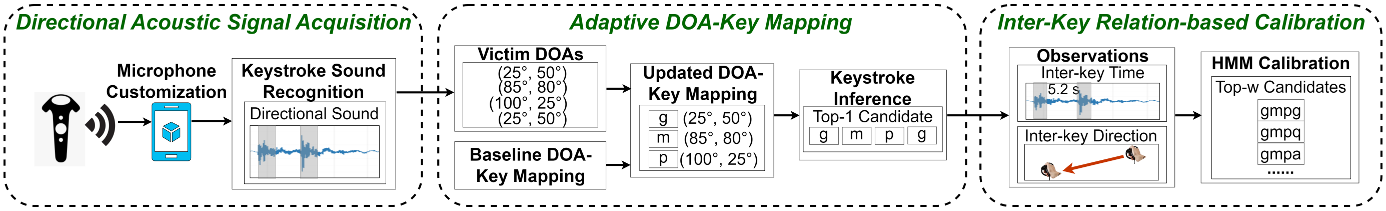


Fig. 4: The architecture of Heimdall.

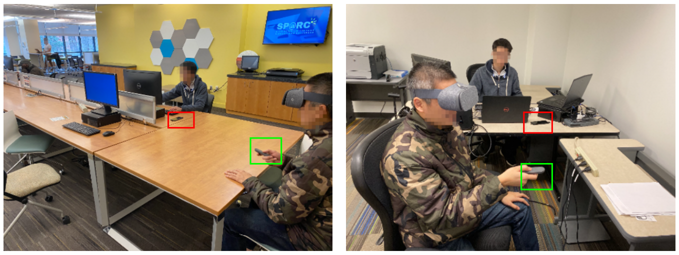

(a) Public attack – a library (b) Insider attack – a laboratory

Fig. 5: Sample attack scenarios.

III. Threat Model

We consider a scenario where a victim is typing sensitive input in VR using a controller. For example, the victim enters passwords or answers to security questions to log in to a medical therapy account. Meanwhile, a malicious smartphone is placed near the victim to record the keystroke sounds and infer the respective keys. This includes two representative scenarios, (a) public attack – the attacker places the smartphone in a public place, e.g., on a shared table in a library or airport (example shown in Figure 5a), and (b) insider attack – the victims know the attacker so that they (un)knowingly allow the attacker to put the smartphone near them, e.g., in an office or lab (example shown in Figure 5b). Similar attack scenarios were validated in acoustic keystroke attacks on physical keyboards and touchscreens [32], [6], [2], but our VR scenario is stealthier since the victim's eyes are covered by the HMD, making them less vigilant about the proximity, and thus the attacker has more flexibility in choosing how to place the smartphone.

Besides the access to the place where the victim enters VR inputs, we assume that the attacker knows the type of VR HMD the victim is using, which is straightforward to obtain since the attacker can visually recognize these commercial off-the-shelf devices. The HMD type information can determine the virtual keyboard layout and support the keystroke inference.

We do not assume any other capabilities for the attackers. Unlike existing VR keystroke attacks utilizing videos or wireless signals [31], [40], [69], [1], we do not require the attackers to have a line-of-sight view of the victim's HMD and handheld controller, i.e., hand and body movements may partially or completely obstruct the controller. We also do not require setting up a precisely aligned pair of wireless transmitter and receiver on both sides of the victim. Hence, countermeasures aimed at positioning users in a secure layout, e.g., in a corner, or steering users away from potential directions of the malicious camera capturing or wireless sensing, would not be effective against our attack. Our malicious smartphone can be positioned or oriented anywhere surrounding the victim, making the attack more flexible with the physical layout and victim pose. This flexible placement of the malicious recording device is also more practical than prior acoustic attacks that rely on the microphone being situated beneath a touchscreen [72], [34] or immediately adjacent to physical keyboards [6], [73], [32], [13]. Furthermore, the attackers do not have the capability of tricking the victim into installing malware on the HMD, making our attack easier to deploy than the other thread of VR keystroke attacks [35], [70], [62], [68] that depend on malware accessing raw HMD data.

IV. System Overview

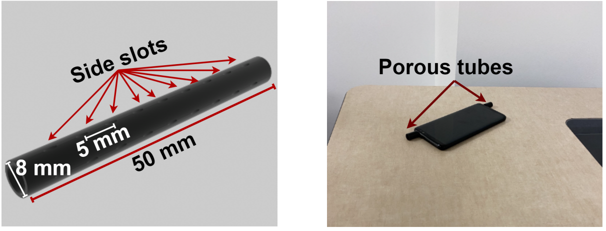
As shown in Figure 4, Heimdall employs three modules to address the three challenges discussed in §II-C. In §V, we propose a directional acoustic signal acquisition framework, where Heimdall records and segments keystroke sounds via easily-customized smartphone microphones. In §VI, we utilize the direction of arrival (DOA) to identify keystroke sounds and develop an adaptive DOA-Key mapping scheme, where Heimdall adapts the DOA-Key mapping pre-collected in the baseline placement to the actual attack and then maps victim keystrokes' DOAs to the respective keys. In §VII, we devise an inter-key relation-based calibration model, where Heimdall uses the time interval and controller moving direction between two keystrokes for a Hidden Markov Model to generate a list of input candidates and enhance the attack performance.

V. Directional Acoustic Signal Acquisition

A. Microphone Customization

Modern smartphones are equipped with two omnidirectional microphones, one at the bottom of the device for regular phone calls and the other at the top for the speakerphone mode. These microphones capture acoustic signals with equal magnitude regardless of their directions of arrival (DOAs). By comparing the signal phases captured by the two microphones and analyzing the time difference of arrival, acoustic attacks were performed on physical keyboards placed on the same 2D plane [73], [32]. However, signal phases alone cannot differentiate VR keystroke sounds in a 3D space because the virtual keyboard, sound source (controller), and microphones are not confined in a 2D plane [10]. To record VR keystroke sounds with distinguishable phases and magnitudes as shown in Figure 3, a new signal acquisition method is needed.

Inspired by how human ears localize sound sources and how sounds travel through ear canals, we posit that if the two



(a) The structure and porous appearance of the tube.

(b) The smartphone with customized microphones.

Fig. 6: The recording prototype of Heimdall.

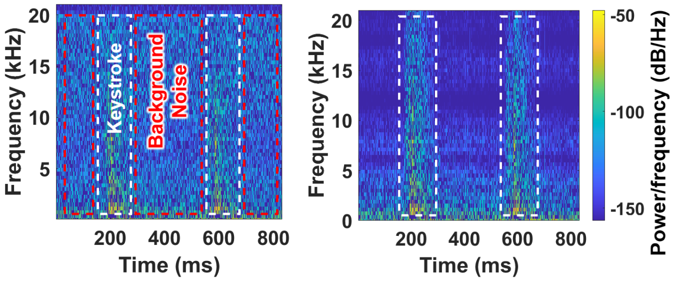


Fig. 7: Signal spectrograms – background noises of the original signal (left) are removed in the denoised signal (right), but the controller clicking sounds are preserved.

smartphone microphones are directional, differentiating the VR keystroke sounds is possible. Similar to human ear canals, directional microphones capture acoustic signals with maximum gain from the primary axis while depressing signals arriving from other directions. When a keystroke sound from a certain DOA arrives at the capsule of a directional microphone, it generates a group of sub-signals with different phases at the side slots of the capsule. These sub-signals travel along multiple paths within the capsule to reach the electronics at the end of the microphone. They interfere and sometimes cancel out each other, producing a uniquely-distorted recording with a distinct phase and magnitude [38]. Hence, we propose to convert off-the-shelf smartphone microphones into directional ones.

Heimdall requires a simple microphone conversion design that does not require advanced crafting techniques and can be launched by someone without a computing or engineering background. The directional microphones must also be implemented without drastically changing the smartphone. For example, separating the microphones with a sound-blocking barrier plate is not appropriate [16].

Directional microphones sense the directionality of sounds via an encapsulating tube with uniformly distributed side slots [38]. Based on this structure, we attach a 5-centimeter porous tube to both microphones that are embedded in the smartphone. As shown in Figure 6a and Figure 6b, the side slots are pierced over the entire tube with a 5-millimeter spacing, enabling the recording of uniquely-distorted signals from different DOAs. Such a centimeter-level porous tube has been shown to be sufficient to generate the unique responses of a directional microphone [4]. We choose plastic tubes for their easy accessibility and prepare them using a 3D printer.

To record the VR keystroke sounds using our customized hardware, we configure the smartphone microphones in stereo audio recording mode with a 48 kHz sampling rate so that the directionality of the sound can be perceived maximally [28]. As the left and right microphones record the keystroke sound independently, Heimdall can theoretically differentiate a minimum distance of 7 millimeters between two sound sources, given that the speed of sound is 340 m/s.

B. Keystroke Sound Recognition

Background Noise Removal. After recording raw acoustic signals, Heimdall removes background noise that can distribute over a wide frequency band, including people talking and walking by. As these noises may overlap with the frequency band of the keystroke sound, using a bandpass or bandstop filter for noise removal may also remove the desired acoustic signals. Therefore, wavelet denoising is employed to remove background noise, as it was proven effective at isolating transient, high-frequency sounds (like controller clicking) and minimizing persistent, low-frequency noise (like human speaking, computer humming, physical keyboard typing) [49]. Specifically, we use Maximal Overlap Discrete Wavelet Transform [46] with a Daubechies 3 wavelet to decompose the sound into 5 bands, i.e.,

$$\alpha_i{}^{(J)} = \sum_{n\in Z} X_n \bar{g}^{(J)}_{n-2^J i} \;\; \beta_i{}^{(j)} = \sum_{n\in Z} X_n \bar{h}^{(j)}_{n-2^j i'} \tag{1}$$

where α_i and β_i are the detail coefficients and the approximation coefficients, $J \in Z$, $j \in \{1, 2, ..J\}$ are the levels and $\bar{g}$, $\bar{h}$ are the discrete orthogonal wavelet functions. We choose this configuration since it effectively captures sharp, transient acoustic features of controller clicking sounds. After decomposing, high-frequency controller clicks primarily appear in the higher-level coefficients, whereas low-frequency background noise falls into the lower-level coefficients. We then apply thresholding to remove small-coefficient background noise using BayesShrink, an adaptive method that determines optimal thresholds for each level based on signal variance and estimated noise [49]. Finally, we reconstruct the acoustic signal based on the updated coefficients using the inverse transform,

$$X_n = \sum_{i\in Z} \alpha_i{}^{(J)} \bar{g}^{(J)}_{n-2^J i} + \sum_{j=1}^{(J)} \sum_{i\in Z} \beta_i{}^{(j)} \bar{h}^{(j)}_{n-2^j i'} \tag{2}$$

Figure 7 shows that Heimdall successfully eliminates a broad spectrum of noises (red dotted boxes) while retaining the controller clicking sounds (white dotted boxes).

Controller Click Segmentation. Controller clicking sounds only account for a small portion of the recorded acoustic signal. After the noise removal, there is a significant portion of irrelevant low-magnitude signal residuals that are non-clicking signal segments. To identify the controller clicking segments, we apply a magnitude threshold to the denoised signal because the controller clicks are salient in magnitude. For those signal periods exceeding the threshold, a peak within the period is identified. Then a 200 ms of signal centered at the peak is extracted as the controller clicking segment. To optimize the threshold for the controller click segmentation, we conducted a pilot study and utilized the data as ground truth. Five

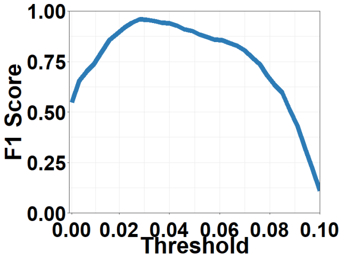


Fig. 8: Optimizing the threshold for click segmentation.

participants entered twice the sequence “GMPQZ” while we recorded the acoustic signals one meter away by a Samsung Galaxy S8 with the proposed directional microphones. We compare the recognized controller clicks with the ground truth and calculate the F1 score as the threshold varies from 0.01 to 0.1 in 50 steps. As shown in Figure 8, the highest F1 score of 0.967 is achieved when the threshold reaches 0.028, which is used for the click segmentation in Heimdall.

Keystroke Session Recognition. Apart from keystrokes, user interaction with other VR content also involves clicking the controller. In other words, not all controller clicks in VR are keystrokes. Similar to the findings of a recent study [68], we observe distinct patterns of inter-click interval in VR keystroke sessions compared to other VR interactions using controller clicks. Figure 9 depicts the controller clicks during a sample use case when a user views a webpage and logs into a private account. The keystroke session for entering the password shows temporally-stable clicks, which differ from other interactions with more sporadic clicks, such as launching the web browser App from the VR Home page, navigating the webpage via dragging, activating the virtual keyboard after clicking the login button, and confirming the password.

The reason behind this phenomenon is that users tend to move the controller around without a noticeable pattern when interacting with and clicking non-keyboard content. This is driven by the fact that non-keyboard VR content pops up or refreshes constantly and it may appear in varying locations in the virtual world. As a result, the sporadic hand movements and controller clicks introduce a higher variance in the inter-click interval between non-keyboard clicks. In contrast, the VR keyboard is fixed in the virtual world and users are familiar with its “QWERTY” layout. This constrains the movement scale of the controller and leads to more stable time intervals between keystroke clicks.

Given the similar observations, we apply the keystroke session recognition approach [68] used in a recent VR keystroke attack to differentiate the keystroke session from other VR clicks. Specifically, the click frequency based approach involves solving the optimization below,

$$\begin{aligned} &\arg\max_{t_s,t_e} \; t_e - t_s \\ \text{s.t.} \quad & f_{min} < Freq(t_s,t_e) < f_{max} \\ & t_{min} < t_e - t_s \end{aligned} \tag{3}$$

where t_s and t_e are the start and end time of the keystroke session, $Freq(t_s,t_e)$ is the controller clicking frequency in the keystroke session bounded by f_{min} and f_{max}, and t_{min} is the minimum duration of the keystroke session. According to the Word-Per-Minute (WPM) performance for VR input [7], we configure f_{min} and f_{max} as 0.2 and 0.4. respectively. Since we consider sensitive input of more than four characters in our experiments, we set t_{min} to 7.5 seconds given $f_{max} = 0.4$. Then t_s and t_e can be derived.

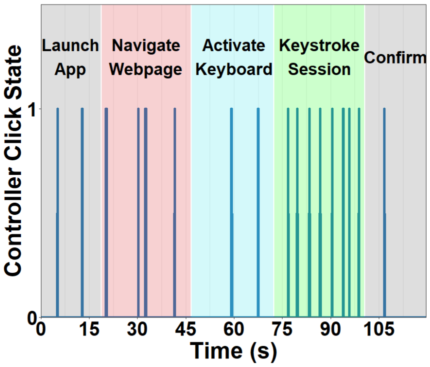


Fig. 9: The keystroke session presents distinct controller click patterns from other activities.

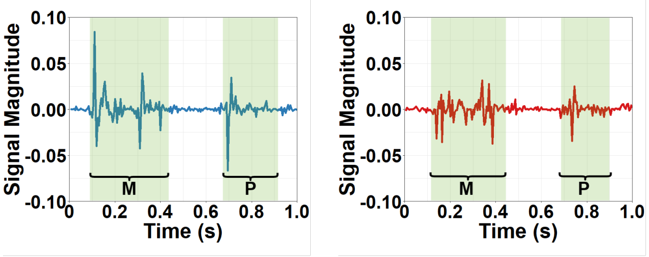
We validate this approach in our study. Five users were asked to enter three sets of keystrokes in three Apps – VR Browser, VR shopping, and VR sightseeing, where they can freely interact with the App and enter the keystrokes. The keystrokes for the three Apps are the password of an existing browser account, the password of an existing payment account, and a video title to search, respectively. The average duration of a study session is 3.5 minutes. We calculate True Positive Rate (TPR) as the number of correctly recognized keystrokes over the total number of keystrokes and False Positive Rate (FPR) as the number of incorrectly recognized keystrokes over the total number of non-keystroke clicks. The TPR of 0.98 and FPR of 0.04 are consistent with the previous results in [68], demonstrating the effectiveness of the approach.

VI. Adaptive DOA-Key Mapping

A. *Varying Sound-Key Mapping*

Once differentiable acoustic signals are recorded and keystrokes are recognized, attackers can pre-collect the keystroke sounds for all target keys as a baseline. If the microphone-user placement in the actual attack is similar to the baseline case, attackers can utilize the baseline sound-key mapping to map a victim’s keystroke sound to the respective key. This works well for traditional acoustic keystroke attacks that assumed consistent placements, e.g., exploiting compromised built-in smartphone microphones underneath a touchscreen [72], [34] or microphones placed immediately around physical keyboards [6], [73], [32], [13]. However, this strong assumption mitigates the threat in practice. In this paper, we consider the practical scenario where the microphone-user position and orientation in a VR keystroke attack can vary significantly from the baseline.

We validated the significant impacts of smartphone-user placements in our pilot study. After obtaining the results in Figure 3 by placing the smartphone one meter in front of the



(a) Left channel – user #1

(b) Right channel – user #1

Fig. 10: Keystroke sound signal collected from user #1 in a new smartphone placement is different from that in Figure 3.

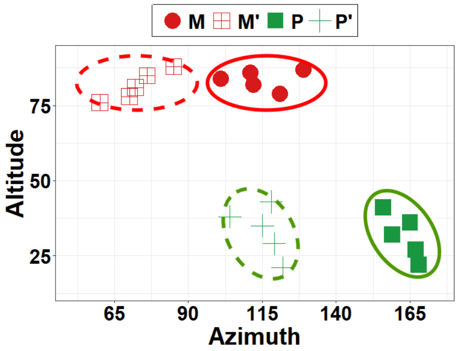


Fig. 11: DOAs collected from user #1 under two placements present traceable changes, e.g., about 45° change in azimuth.

user, we shifted the smartphone one meter to its right without changing its orientation. As exemplified in Figure 10, the phase and magnitude of the keystroke sound signals collected from even the same user #1 can drastically change from the Figure 3 results obtained in the original placement, e.g., earlier phase and higher magnitude for keystroke M in the left channel. This is because the left microphone is now closer to the sound source. The source of the keystroke sound is closer to the left microphone than the right microphone.

It is clear that the baseline mapping between keystroke sounds and keys pre-collected by attackers cannot be directly used in the new placement of an actual attack. Due to the altered spatial relationship between the victim and the smartphone, a keystroke sound may be mapped to a wrong key or may even have no match. Since both the signal phase and magnitude of the same key may change dramatically, no well-defined signal features can be extracted to represent a signal under different placements. Hence, we cannot use the keystroke sounds directly to infer the keys.

However, we discovered that the varying direction of arrival (DOA) of a keystroke sound presents a traceable pattern under different placements. The DOA is defined by azimuth and altitude [14] and is an indirect way to identify a keystroke sound. Figure 11 shows the DOAs of two keystroke sounds from user #1 in the original (M and P) and new placements (M' and P'). The new placement causes similar angular changes to the DOAs. For example, the azimuth changes of keystroke sounds M and P between two placements are both around 45° because the smartphone microphones were moved to the right by 1 meter. Given this noticeable pattern, we propose using DOAs to identify keystroke sounds and build the baseline mapping. We can then update the DOAs for target keys based on smartphone displacement in the attack to infer victim keystrokes.

B. Adaptive Mapping Procedure

Heimdall first pre-collects the DOAs for all target keys in a baseline smartphone-user placement, creating a DOA-Key mapping. It then derives the DOAs of the recorded victim keystroke sounds. Next, Heimdall updates the baseline DOA-Key mapping based on the new spatial relationship between the victim and smartphone in the attack. Finally, it maps the DOA of the victim keystrokes to the respective key via the updated DOA-Key mapping.

Step 1 – Collecting Baseline DOA-Key Mapping. In a baseline placement, attackers can enter a key k and directly access the coordinate of the controller in the VR OS. The VR OS coordinate system can be easily converted to the attack coordinate system with the origin point set at the center of the fixed smartphone since the layout between the smartphone and the attacker is known. Given the converted controller position (x_k, y_k, z_k), keystroke k's DOA $d_{k,base}$ can be derived, i.e.,

$$\begin{aligned} d_{k,base} &= [\varphi_k \quad \theta_k], \\ \varphi_k &= arctan(\frac{y_k}{x_k}), \\ \theta_k &= arccos(\frac{z_k}{r_k}) \end{aligned} \tag{4}$$

where φ_k is the azimuth of the controller position, θ_k is the altitude, and $r_k = \sqrt{{x_k}^2 + {y_k}^2 + {z_k}^2}$ is the distance between the origin point and the controller. By repeating this process for a set of keys $\mathcal{K}$ including 26 English letters and 10 digits, we have the baseline DOA-Key mapping $\{k \in \mathcal{K}, d_{k,base}\}$.

Step 2 – Deriving DOAs of Keystroke Sounds in the Attack. Since attackers cannot directly access the coordinates of the victim keystroke sounds in the attack, we propose a method to derive the DOAs. A keystroke sound interacts with the tube structure of the directional microphones before being recorded. The impulse responses of the left and right tubes T_l and T_r affect the phase and magnitude of the recorded signal. Besides, the signal propagation from the sound source to the left and right microphones in the recording environment, characterized by the environmental channel response E_l and E_r, also affects the recorded signal. Thus, the signal recorded by the left and right microphone in the frequency domain, Y_l and Y_r, are

$$\begin{aligned} Y_l &= X \cdot E_l \cdot T_l, \\ Y_r &= X \cdot E_r \cdot T_r \end{aligned} \tag{5}$$

where X is the source sound signal. Since left and right microphones are close to each other in the same environment, E_l and E_r are approximately equal [19]. Thus we have,

$$\frac{Y_l}{T_l} = X \cdot E_l \approx X \cdot E_r = \frac{Y_r}{T_r} \tag{6}$$

We then obtain the tube responses $T_l(d), T_r(d)$ under different DOAs d using the Maximum Length Sequence (MLS) method [51], a standard procedure for deriving impulse response. Specifically, our smartphone is placed at the center of an anechoic chamber to record acoustic signals, while a segment of MLS signal is generated from a given DOA d relative to the origin point. Without interference from the environmental channel response, the MLS signal recorded by the left microphone $Y_{0,l}(d)$ and right microphone $Y_{0,r}(d)$ can be expressed in the frequency domain as,

$$\begin{aligned} Y_{0,l}(d) &= X_0 \cdot T_l(d), \\ Y_{0,r}(d) &= X_0 \cdot T_r(d) \end{aligned} \tag{7}$$

where X_0 is the known MLS signal. For each DOA d, we use the $Y_{0,l}(d)$ and $Y_{0,r}(d)$ to obtain the tube responses, i.e.,

$$\begin{aligned} T_l(d) &= \frac{Y_{0,l}(d)}{X_0}, \quad \forall d \\ T_r(d) &= \frac{Y_{0,r}(d)}{X_0}, \quad \forall d \end{aligned} \tag{8}$$

We can then obtain the DOA of a keystroke sound d^* by finding the d that maximally satisfies Equation 6, i.e.,

$$d^* = \arg\min_{d \in \mathcal{D}} |\frac{Y_l}{T_l(d)} - \frac{Y_r}{T_r(d)}| \tag{9}$$

where $\mathcal{D}$ is the set of possible DOAs. Since VR keystroke sounds originate from a discrete number of DOAs, we consider a finite set $\mathcal{D}$ where the azimuth of a DOA increases from 0° to 359° and the altitude of a DOA increases from 0° to 180°, both at a step size of 1°.

By following the above methodology, attackers can obtain the DOA of victim keystroke sounds in the attack. Given the set of sensitive inputs $\mathcal{S}$, e.g., an 8-key password, the DOA $d_{s,atk}$ of a victim keystroke $s \in \mathcal{S}$ can be derived as,

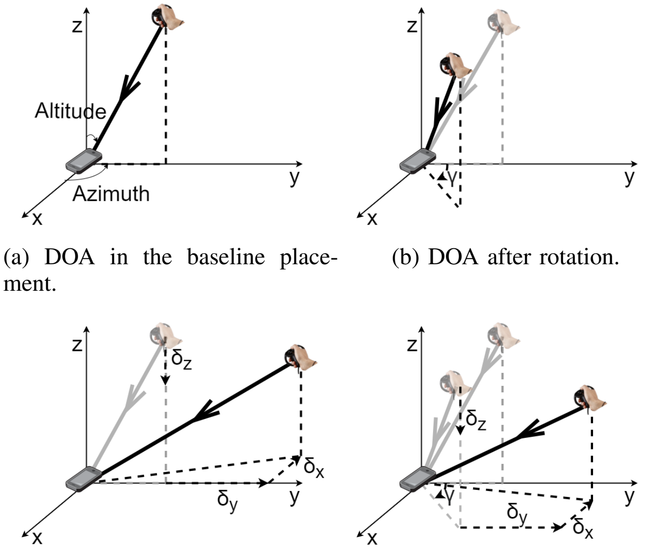
$$d_{s,atk} = \arg\min_{d \in \mathcal{D}} |\frac{Y_{s,l,atk}}{T_l(d)} - \frac{Y_{s,r,atk}}{T_r(d)}| \tag{10}$$

where $Y_{s,l,atk}$ and $Y_{s,r,atk}$ are the victim keystroke sounds of key s recorded by left and right microphones in the attack.

Step 3 – Updating DOA-Key Mapping. In the attack, the relative position and orientation between the smartphone and the user can differ from the baseline placement. This could change the "supposed" DOA of a keystroke and may be caused by the displacement of either the smartphone or the user. Note that our design explores the relative position and orientation between smartphone and user and thus is generic regardless of which is displaced. Without loss of generality, we consider a fixed smartphone to derive the displaced DOAs for all target keys and update the DOA-Key mapping. Given a keystroke sound originated at (x_k, y_k, z_k) with the DOA of $[\varphi_k, \theta_k]$ in the baseline placement, there are two basic types of displacement in the attack, namely rotation and translation [67].

Rotation is the case when the user circles around the smartphone while facing toward it at a fixed distance. Given the DOA of a sample keystroke in the baseline placement in Figure 12a, the DOA rotates around the z axis for a degree γ as shown in Figure 12b. The DOA after rotation, $d_k(\gamma)$, can be derived as,

$$\begin{aligned} d_k(\gamma) &= [\varphi_k(\gamma) \;\; \theta_k], \\ \varphi_k(\gamma) &= arctan(\frac{y_k}{x_k}) + \gamma \end{aligned} \tag{11}$$



(a) DOA in the baseline placement.

(b) DOA after rotation.

(c) DOA after translation.

(d) DOA after rotation and translation.

Fig. 12: An illustration of the DOA displacement.

Translation is the case when the user moves to a new position without changing her orientation. In Figure 12c, the DOA experiences a shift of δ_x, δ_y and δ_z along the x, y and z axis, respectively. The DOA after translation, $d_k(\delta_x, \delta_y, \delta_z)$, becomes

$$\begin{aligned} d_k(\delta_x, \delta_y, \delta_z) &= [\varphi_k(\delta_x, \delta_y) \;\; \theta_k(\delta_x, \delta_y, \delta_z)], \\ \varphi_k(\delta_x, \delta_y) &= arctan(\frac{y_k + \delta_y}{x_k + \delta_x}), \\ \theta_k(\delta_x, \delta_y, \delta_z) &= arccos(\frac{z_k + \delta_z}{r_k(\delta_x, \delta_y, \delta_z)}), \\ r_k(\delta_x, \delta_y, \delta_z) &= \sqrt{(x_k + \delta_x)^2 + (y_k + \delta_y)^2 + (z_k + \delta_z)^2} \end{aligned} \tag{12}$$

Overall, any displacement of the DOA in the attack can be represented by a combination of rotation and translation. The example in Figure 12 shows that after being rotated (Figure 12b) and translated (Figure 12c), the DOA is displaced as shown in Figure 12d. To generalize, the displaced DOA of a keystroke k in the attack placement $d_{k,atk}(\delta_x, \delta_y, \delta_z, \gamma)$ can be obtained by,

$$\begin{aligned} d_{k,atk}(\delta_x, \delta_y, \delta_z, \gamma) &= [\varphi_{k,atk}(\delta_x, \delta_y, \gamma) \;\; \theta_{k,atk}(\delta_x, \delta_y, \delta_z)], \\ \varphi_{k,atk}(\delta_x, \delta_y, \gamma) &= arctan(\frac{y_k + \delta_y}{x_k + \delta_x}) + \gamma, \\ \theta_{k,atk}(\delta_x, \delta_y, \delta_z) &= arccos(\frac{z_k + \delta_z}{r_k(\delta_x, \delta_y, \delta_z)}) \end{aligned} \tag{13}$$

Given the measured DOAs of the victim keystrokes $d_{s,atk}$, $s \in \mathcal{S}$, and the displaced DOA expressions for all keys in the attack $d_{k,atk}(\delta_x, \delta_y, \delta_z, \gamma), k \in \mathcal{K}$, we aim to find the displacement parameters $\delta_x, \delta_y, \delta_z, \gamma$ that allows a subset of the displaced DOAs to maximally match with the measured victim DOAs. This way, we obtain optimal displacement parameters π and use π to update the baseline DOA-Key mapping to the attack DOA-Key mapping. The optimization is expressed as,

$$\pi = \underset{\delta_x,\delta_y,\delta_z,\gamma}{\arg\min} \sum_{s\in S} \Delta(d_{k,atk}(\delta_x,\delta_y,\delta_z,\gamma), d_{s,atk}) \quad (14)$$

where the function $\Delta(d_k, d_s) = \min_{k\in\mathcal{K}} |\varphi_k - \varphi_s| + |\theta_k - \theta_s|$ identifies the displaced DOA that maximally matches with the given victim DOA (one out of 36 in $\mathcal{K}$) under a certain displacement parameter and then returns the absolute differences of azimuth and altitude between the two DOAs. By searching a finite range of displacement parameters, with $\delta_x, \delta_y, \delta_z$ from -1 meter to 1 meter at a step size of 0.05 meter, and γ from 0° to 359° at a step size of 1°, we can solve Equation 14. With the optimal displacement parameter π, we can obtain the updated DOA-Key mapping in the attack $\{k \in \mathcal{K}, d_{k,atk}(\pi)\}$.

Step 4 – Inferring Keystrokes in the Attack. To infer a victim keystroke s in the attack for a measured DOA $d_{s,atk}$, we simply retrieve the displaced DOA in the updated DOA-Key mapping that is most similar to $d_{s,atk}$ and then return the respective key, i.e.,

$$s = \underset{k\in\mathcal{K}}{\arg\min} |\varphi_{k,atk}(\pi) - \varphi_s| + |\theta_{k,atk}(\pi) - \theta_s| \quad (15)$$

By repeating this process for all $s \in \mathcal{S}$, Heimdall can infer a sequence of victim keystrokes.

VII. Inter-Key Relation-based Calibration

After using the adaptive DOA-Key mapping scheme, we found that keystroke mapping errors mostly occur when a user drastically rotates her wrist and the controller while navigating through an input sequence. This rotation leads to a minimal up/down and left/right hand translation movement. Thus the actual DOA of a keystroke sound may deviate from the "typical" DOA we derived in the DOA-Key mapping, failing keystroke mapping. It is even possible that the controller remains in the same DOA while aiming at different keys.

Figure 13 visualizes the DOAs of keystroke sounds in our pilot study. When users entered "GMPQZ", the DOAs of some "Z" samples were close to those of "Q" and would be incorrectly mapped to "Q". This is because, after hitting "Q", users sometimes rotated the controller down to aim at "Z" rather than translating it down. This error pattern aligns with our reasoning above. Such a one-to-multiple mapping between the DOA and the key makes VR keystroke inference more difficult than acoustic keystroke inference on physical keyboards and touchscreens with a one-to-one mapping.

A. Modeling Keystroke Transition

In §VI, each key is recognized individually based on the DOA of its keystroke sound. However, the transition between two keystrokes has not been explored and will be utilized here to correct the mapping errors. Specifically, even though hand rotation incurs minimal controller translation movement and DOA change, the time interval between two keystrokes still depends on their inter-key distance on the virtual keyboard. For example, after typing "P", rotating the controller to hit "O" takes less time than to hit "T". Furthermore, the direction of the controller's movement between keystrokes relies on where the key pair is located and does not deviate significantly from the mapping result. For example, when entering "G" and "M" successively, the DOA of "M" may be incorrectly mapped to "N" due to the small controller translation movement, but the inferred controller moving direction, i.e., bottom right, should be correct, and is unlikely to be upward, e.g., to "T".

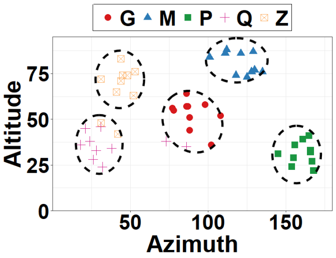


Fig. 13: Mapping errors caused by hand rotation, e.g., some DOAs of keystroke "Z" are mapped to "Q".

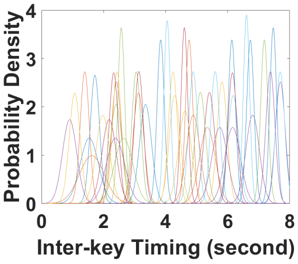


Fig. 14: Probability distributions of inter-key time for 46 keystroke transitions (one Gaussian model per each).

Motivated by these observations, we propose to model the keystroke transition as a Hidden Markov Model (HMM) [53] and then predict extra input candidates to correct the mapping errors. An HMM can be used to model a system where the system states are unobservable ("hidden") and there is an observable process whose outcomes are influenced by and used to infer the hidden state, HMMs have shown success in cases similar to ours, e.g., speech and text modeling [18], [15].

HMM Modeling Overview. The problem we want to solve is to predict the hidden state q that represents the transition between a keystroke pair, e.g., "$A \rightarrow B$", based on observed outcomes, the inter-key time interval o_1 and controller moving direction o_2 that are available in the mapping result of §VI. This meets the Markov properties because observations o_1 and o_2 are highly dependent on q, and the probability of transitioning to another state relies on the current state. Moreover, the probability distribution of o_1 and o_2 for each q only depends on the current state.

To build the HMM, we must derive the start probability of a hidden state $P_0(q)$, the transition probability among hidden states $\mathcal{P}$, and the probability distribution of the observed process given a hidden state $P(o_1|q)$ and $P(o_2|q)$. Then, we can easily calculate the probability of a keystroke transition given our observations $P(q|o_1, o_2)$ using Bayes' theorem [53].

As the input sequence can start from anything in the 26 English letters and 10 digits, $P_0(q) = \frac{1}{(26+10)\times(26+10)}$. The probability of moving from one keystroke transition to the other $\mathcal{P}$ is either 0 or $\frac{1}{36}$ depending on if they share the same "relaying" key. For example, given the current keystroke transition $A \rightarrow B$ (after entering "A" and "B" successively), the next keystroke transition has to originate from "B". Therefore, the probability from "$A \rightarrow B$" to "$C \rightarrow D$" is 0, while the probability from "$A \rightarrow B$" to "$B \rightarrow C$" is $\frac{1}{26+10}$.

Probability Distribution of Controller Moving Direction Given a Keystroke Transition. We consider nine possible directions: up, up-right, right, down-right, down, down-left, left, up-left, and no-move. Given the direction of a keystroke transition $\vec{t}$, four possible controller moving directions may be observed, including two that are 45° away from $\vec{t}$, one exactly matching $\vec{t}$, and one without movement. Each of the four has a probability of 0.25, which collectively represent $P(o_2|q)$. For example, given the inferred transition in §VI being "$G \rightarrow K$", the observed direction could be right, down-right, up-right, or no-move with an equal probability of 0.25. Other directions are considered impossible.

Probability Distribution of Time Interval Given a Keystroke Transition. We propose deriving $P(o_1|q)$ from a smaller set of representative keystroke transitions, rather than collecting time interval data from all possible keystroke transitions ($36 \times 36 = 1296$ cases). The rationale is that the time cost of a keystroke transition depends on the inter-key distance. For example, the time interval of "$Q \rightarrow R$" would be similar to "$W \rightarrow T$". We focus on 46 keystroke transitions with distinct inter-key distances based on a measurement study of keyboard inter-key distances [45]. The distance between keys in the same row is directly measured, e.g., 5 between "Q" and "Y". For keys in different rows, we measure their center-to-center distance and identify the most similar distance between two keys in the same row. For example, the distance between "Q" and "V" is the most similar to that between "Q" and "Y", i.e., 5 units. As the time interval of a keystroke transition forms a Gaussian-like unimodal distribution [63], we use univariate Gaussian distribution to model $P(o_1|q)$ for the 46 values of q [63], as shown in Figure 14. The coverage of the probability distribution functions indicates that the 46 keystroke transitions are representative.

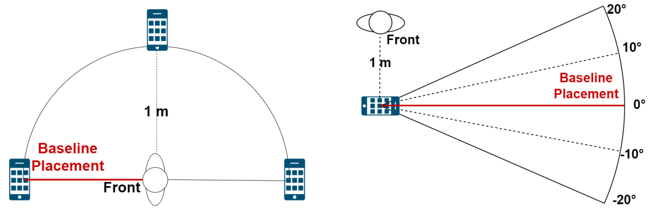
B. Keystroke Mapping Correction

Given $P(q|o_1, o_2)$ obtained in §VII-A, we can derive the probability of a keystroke sequence given our observations. The time interval of a keystroke transition o_1 can be observed from the recorded acoustic signals. The controller moving direction o_2 can be observed from the keystroke mapping result obtained in §VI. To calibrate the keystroke mapping result, our *calibration step 1* is to generate an exhaustive list of all possible keystroke sequences and rank them by their HMM-derived probabilities in decreasing order.

As mapping errors do not always happen, a sequence completely different from the mapping result is unlikely. Thus, we prefer sequences with a high probability but also a high similarity to the mapping result. Therefore, our *calibration step 2* is to remove from the list in step 1 those sequences that have a large Hamming distance from the mapping result. To configure the optimal Hamming distance for `Heimdall`, we measure the password inference accuracy under varying Hamming distances from 1 to 4 in our pilot study. We identify the optimal Hamming distance of 2 and observe that the accuracy variation among different Hamming distances is only around 2%. The small discrepancy is attributed to the fact that most of the high-probability calibrated sequences are similar to the initial mapping result. Finally, we are able to output the inference candidates, where the top-1 candidate is the mapping result in §VI and the remaining candidates can be obtained from the highest ranking sequences after step 2 above.

VIII. Experiment Setting

Apparatus. Our prototype consisted of a smartphone with the proposed directional microphones and a VR HMD under attack (Figure 5). As `Heimdall` does not have a real-time requirement, we separated the acoustic signal acquisition and the keystroke inference to minimize the workloads on the smartphone and thus avoid potential battery outages during the attack. We used a Samsung Galaxy S8 and a Google Daydream as the main smartphone and VR HMD, respectively. We also validated other devices in the robustness analysis.



(a) Various positions by translation (b) Various orientations by rotation.

Fig. 15: Top view of smartphone-user placements.

Participants. We recruited 30 university students (18 males and 12 females, aged from 19 to 30) for the study through social media and campus email lists. Seven participants had prior VR experience, while 23 had not used VR before. All participants were informed that their VR keystroke sounds would be recorded to infer their keystrokes. They signed a written consent form in accordance with an IRB approval which allows for recording human behavior in VR evaluations. In a practice session, they were instructed to enter at least 10 keys until they became familiar with VR keystrokes.

Data Collection. We first developed a Unity App featuring a standard QWERTY keyboard from Unity's Asset Store. In a single round of experiments, one participant acted as the attacker and the remaining 29 were the victims. To derive the baseline DOA-Key mapping, the attacker entered 26 English letters and 10 digits for 10 trials in the baseline placement in Figure 15 (marked by red). We collected the coordinates of the controller through VR OS and calculated the average of the 10 trials for the derivation. For the HMM modeling in §VII-A, the attacker entered 46 representative keystroke transitions for 10 trials.

To test the attack, each of the 29 victims entered a list of 45 passwords in 7 different smartphone-user placements. Since any relative position and orientation between the user and the smartphone can be achieved by fixing one of them and then translating and rotating the other, we considered placing the smartphone with various translations and rotations in Figure 15 in order to cover a diverse spatial relationship in the attack. Note that Figure 15 is a top view. For example, the user is facing toward the figure's left in Figure 15a. From the reader's perspective, the left, up, right smartphones correspond to smartphone placements in front of, on the right of, and behind the user in the physical world. The "behind" and "on-the-right-of" positions represent non-line-of-sight scenarios where body and arm movements completely or partially block the view between the smartphone and the VR controller. The passwords were selected from the most common passwords [60] consisting of 4 to 8 letters and digits. We repeated this data collection for 30 rounds so that each participant was

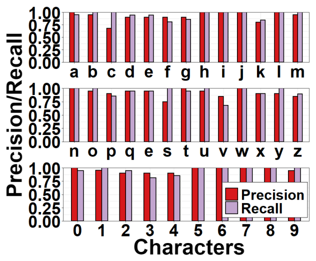


Fig. 16: Heimdall performs well across all 36 characters in the attacks.

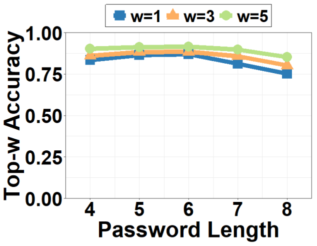


Fig. 17: Inference accuracy for passwords with different lengths.

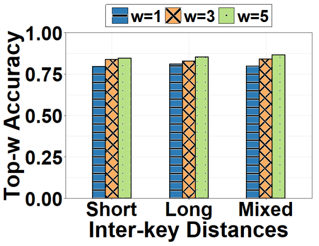


Fig. 18: Inference accuracy for passwords with different inter-key distances.

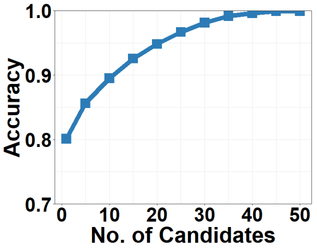


Fig. 19: Password inference accuracy versus the number of top candidates.

the attacker once. We conducted a total of 29 victims × 45 passwords × 7 placements × 30 rounds= $274,050$ attacks and herein report the average results.

Evaluation Metrics. We use top-w accuracy to evaluate password inference [31], [1]. It counts a successful inference only if one of the w candidates exactly matches the entered password (i.e., all inferred keystrokes of a password are correct). We also evaluate the key inference accuracy by retrieving the top-1 inference result and calculating the ratio between the number of correctly inferred characters and the total number of characters in all passwords.

IX. Evaluation Results

A. Performance of Keystroke Inference

Analysis of All Characters. As a benchmark study of the keystroke inference, we first dissect the results and evaluate the inference for each alphanumeric character on the virtual keyboard. We use precision and recall instead of accuracy because the dataset is unbalanced across these characters. Figure 16 shows that Heimdall has a promising performance for all characters. The average precision and recall across characters reach 95.14% and 96.29%. As we will discuss later, the more challenging task for Heimdall is to infer a random password that has a sequence of characters.

Analysis of Passwords Details. To study the impact of password lengths, we analyze top-w accuracy in Figure 17. We observe that top-1, 3, 5 accuracy initially increase with password length. The maximum top-1, 3, 5 accuracy of 86.97%, 88.43%, and 91.58% respectively is achieved with six-character passwords because more characters allow the DOAs of victim keystroke sounds to better match with the displaced DOAs in the updated DOA-Key mapping (Equation 14). The accuracy does not keep increasing for passwords longer than six characters since typing them introduces more hand rotation and mapping errors. However, it is important to note that even an 8-character password poses a legitimate threat, as top-5 accuracy of 85.14% can still be achieved.

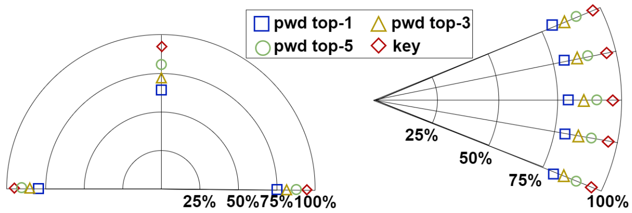


Fig. 20: Accuracy is stable under different smartphone-user placements tested in Figure 15.

Since the inter-key time is a major factor of the proposed calibration, we investigate the performance for passwords with different inter-key distances. The 45 passwords are categorized into three groups (15 passwords each). Passwords only containing inter-key distances of 5 characters or more are called "Long", while those only containing 5 or fewer are called "Short". "Mixed" passwords contain both long and short inter-key distances. Figure 18 shows that the password inference accuracy is stable across all three groups. This confirms that Heimdall can address a wide variety of different random inputs thanks to the proposed keystroke mapping and calibration schemes that explore the acoustic signal and inter-key relation of VR keystrokes.

Analysis of More Inference Candidates. As the proposed calibration model can generate more password candidates than five, we study to what extent Heimdall can be improved when more password candidates are included in the attack. Figure 19 demonstrates that the password inference accuracy quickly increases as the number of candidates increases. Specifically, a password of 4–8 characters can be hacked with the accuracy of 95% within 20 attempts. Hence, the time complexity for brute-force attacks towards a VR system does not appear to be a significant barrier for the attacker, making Heimdall a serious threat to VR systems.

We also observe that the accuracy converges at 98% with 50 candidates, i.e., when the HMM-generated candidates correct all inference errors caused by wrist rotation. Compared to 80% accuracy with one candidate when no wrist error is corrected, we can derive that wrist rotation would make 18% of inferences incorrect. The remaining 2% stems from inaccurate DOA-Key mapping.

Analysis of Smartphone-User Placement. We analyze the attack performance under various relative positions and orientations between the smartphone and the victim as illustrated in Figure 15. The results in Figure 20 show that Heimdall achieves consistent top-w accuracy (denoted by "pwd top-w") and key inference accuracy (denoted by "key") across different smartphone-user placements due to the proposed adaptive DOA-Key mapping scheme that can handle an arbitrary spatial layout. This confirms

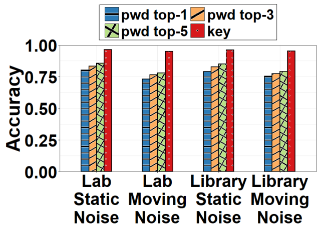


Fig. 21: Heimdall is resilient to different attack environments.

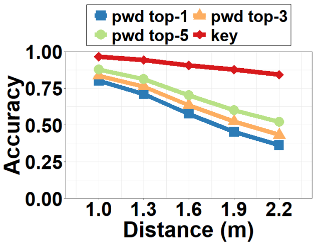


Fig. 22: Accuracy under different smartphone recording distances.

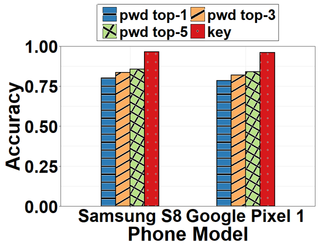


Fig. 23: Heimdall is applicable to different smartphones.

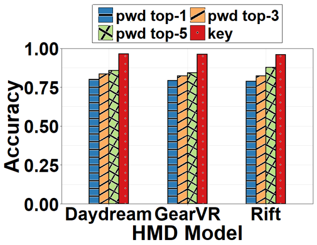


Fig. 24: Heimdall can be generalized to different VR HMD platforms.

that attackers can utilize the pre-collected baseline DOA-Key mapping to launch attacks in a placement-flexible manner, posing greater threats than previous placement-sensitive VR keystroke attacks.

We also observe that the non-line-of-sight scenarios where body movement completely or partially obscures the direct link between the victim and the controller, e.g., when the smartphone is behind or on the right of the victim, achieve acceptable accuracy. For example, the "on-the-right-of" position achieves `key inference accuracy` of 92.06%, a slight decrease from 96.51% in the baseline placement. The "on-the-right-of" position is more challenging than the "behind" position because the DOAs of clicking different keys become similar to, sometimes overlapping with, the orientation of the smartphone. Consequently, the key differentiation becomes more difficult.

B. *Robustness of Keystroke Inference*

We now perform a robustness analysis of Heimdall in various practical scenarios introduced by attacker capabilities and victim conditions. In each new scenario, five victims additionally entered the 45 passwords while the keystroke sounds were recorded in the baseline placement. For the attack, we re-used the attack system obtained in §VIII without collecting new system tuning data unless otherwise noted.

Impact of Attack Environments. We start with evaluating the performance under different attack environments. We conduct attacks in two rooms with different layouts and furniture (a laboratory office and an open space in a library), and tested two cases of background noise in each room – static noise (desktop fans and air conditioners) and moving noise (people talking and walking). Results are shown in Figure 21, where we see that Heimdall performs stably despite different attack environments. This is attributed to the wavelet denoising scheme in §V-B that removes noise in the background. Further, Heimdall can also address real-world environmental factors including multi-path distortion because our frequency-domain-division approach removes the environmental channel response and preserve the microphone-recorded signals (Equation 5-9).

Impact of Recording Distances. We then evaluate Heimdall under different distances between the victim and the smartphone. Figure 22 shows that, as the distance increases from 1 m to 2.2 m, `key inference accuracy` decreases from 96.51% to 84.22%. The decrease in accuracy stems from the small signal strength when the distance enlarges. This error accumulates for a password inference where multiple keystrokes must be all correctly identified. Despite the degradation, the `key inference accuracy` of about 85% at 2.2 meters can still expose significant information that might be exploited by other systems to launch secondary attacks. Attackers may balance attack performance and stealthiness by varying the recording distance. They may improve the inference accuracy at a given distance through signal enhancement algorithms [21].

Impact of Smartphone Models. Different smartphone models may have different microphones and distances between the dual microphones, but these factors do not necessarily affect the quality of the recorded signal. The main factor that affects Heimdall is the acoustic sampling rate. As some modern smartphones only support a sampling rate of 44.1 kHz (lower than the 48 kHz of Samsung Galaxy 8), we evaluate this case on Google Pixel 1. As shown in Figure 23, the accuracy decreases by about 1%. This indicates that a difference of ~ 4 kHz in sampling rate is not enough to cause drastic performance degradation, validating the applicability of Heimdall across smartphone models.

Impact of VR HMD Models. The main difference between attacking different VR HMD models is the virtual keyboard layout. We evaluate Heimdall on additional VR platforms including Samsung GearVR and Oculus Rift. In this evaluation, we re-collected the attacker data for tuning the baseline DOA-Key mapping and the calibration model before asking the victims to use these new HMDs. As shown in Figure 24, we observed a negligible difference in performance, confirming that Heimdall can be generalized to various HMDs.

Short-term Performance. In addition to the factors of the environment and the attacker, we are interested in the impact of victim conditions on Heimdall. We first conducted a short-term study to understand if the change of one's physical state during a day would affect the attack. We collected testing data four times in a day from 10 AM to 6 PM since growing fatigue may change user movement extent and speed. Figure 25 shows that the fluctuation of inference accuracy is within 2% across the day, suggesting that user fatigue does not significantly change keystroke patterns in VR.

Midterm Performance. We also conducted a midterm study to test Heimdall over a week, hypothesizing that user behavior would change over time and affect inference accuracy. We

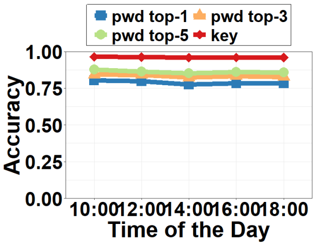


Fig. 25: Inference accuracy within a day.

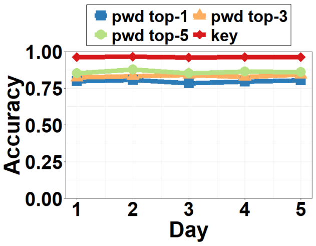


Fig. 26: Inference accuracy over five days.

evaluated the performance once a day for five days, but no significant variance was observed (Figure 26). The possible explanation is that the participants became more familiar with committing keystrokes in VR and thus their typing pattern remained unchanged during this intermediate term. Hence, we conclude that Heimdall can maintain its performance for a reasonable period.

X. Limitation and Discussion

Multiple Displacement Solutions. When updating the DOA-Key mapping for the attack placement, optimizing Equation 14 may lead to more than one displacement solutions. This is because various subsets of DOAs may match with the measured victim DOAs under a certain displacement. However, the occurrence of multiple solutions is less likely with longer passwords as the matching has to occur between more keys. Our experiments only had five such passwords, all with a length of four characters, resulting in an average of 5.6 solutions. These passwords typically have keys confined in a small region, e.g., "QWAS". For these passwords, we randomly selected a displacement solution and achieved a negligible effect on the top-5 and top-3 accuracy. To improve their top-1 accuracy and the overall performance of Heimdall, future work should focus on formulating and solving a constrained optimization problem to replace Equation 14.

Generalization. It is true that a user may occasionally have non-regular hand motion, which can potentially invalidate the attack. However, various VR keystroke attacks [35], [70], [62], [68] based on raw sensor data have demonstrated a consistent and traceable correlation between keystrokes and hand motion across different users. To address potential non-regular behavior in our scenario, Heimdall utilizes the HMM-based calibration scheme to accommodate hand rotation. Our evaluation involves 30 participants and 274,050 attack attempts, allowing us to capture diverse typing behaviors and further validate the overall threat to VR keystrokes.

Furthermore, Heimdall targets a single VR controller, but some commercial VR HMDs support two-controller interactions, which makes keystroke inference more challenging. However, the insight of this research, such as DOA-Key mapping and inter-key relation-based calibration, can still apply to a two-controller VR system. While the similar principle of Heimdall should work, we believe that a full-scale study is needed to explore this topic. Additionally, non-alphanumeric keys such as caps lock and special characters may be typed. To infer these keys, we can enlarge the DOA-Key mapping and use a more sophisticated decision algorithm.

Prior VR Experience. We observed no significant difference in results among participants with varying prior experience. This is attributed to the pre-study practice session familiarizing participants with VR navigation. Furthermore, the QWERTY keyboard layout is well-known, making prior experience less relevant.

Countermeasures. One method to prevent the keystroke inference is to intentionally disrupt the correlation between keystroke sounds and their respective keys. The victim may achieve this goal by utilizing a randomized keyboard layout for every single keystroke session [58]. However, this strategy can be cumbersome. This functionality is also not natively supported by VR OS, requiring users to possess advanced programming skills to implement it. In fact, due to these usability considerations, virtual keyboards used for entering sensitive data on most mobile platforms such as HoloLens, Android, and iOS still adhere to the "QWERTY" layout.

Another alternative countermeasure is to interrupt the continuity of the keystrokes and thus prevent the attack. The rationale is that existing keystroke attacks, including Heimdall, assume the continuous typing of characters on the virtual keyboard. The victim may conduct "fake" clicks outside the keyboard in between actual keystrokes, or move around on a large scale to switch the view of the virtual keyboard during the keystroke session. While these methods may be effective against keystroke attacks, they would disrupt the smoothness of VR interactions and affect user experience in VR [54], [39].

Users may also mitigate the attack by eliminating keystroke sounds through a mechanical keyboard. However, most VR headsets lack native mechanical keyboard support. While a few headsets support external keyboards, they require customization, often unfamiliar or non-trivial to users [50]. It is also important to note that using mechanical keyboards may expose users to common shoulder-sniffing attacks [56], [11], as the pressed keys are fully visible to the public.

XI. Related Work

Acoustic Keystroke Inference Attacks. Keystroke inference attacks have been conducted through side channels such as network traffic [63], videos [56], [11], electromagnetic signals [26], device motion [64], [37], [33], and ambient light variation [55]. Acoustic attacks are practical due to their stealthiness and easy access to sound signals. They collected acoustic signals by one or more smartphones and inferred keystrokes by analyzing language models [2], [6] or signal properties [73], [32], [13], [74], [3]. Similar efforts were made on touchscreens [72], [12], [61], [34], [44], e.g., using near-ultrasonic signals for finger tracking and keystroke inference. These attacks inspired the Heimdall design. However, prior systems cannot be applied to VR due to the unique VR keystroke modality relying on a moving controller in a 3D space. This paper presents the first acoustic keystroke attack on VR devices.

Attacks on Virtual Reality. Research has been conducted to attack VR systems. Previous attacks used video surveillance [31], [40], [69] and wireless signal sensing [1]. However, the video-based attack requires a line-of-sight view of the controller movement, and the wireless-based attack needs the user to be positioned between the transmitter and receiver to detect signal turbulence caused by hand movement. In contrast, Heimdall can place the malicious smartphone surrounding the victim in any orientation and position, even in non-line-of-sight scenarios. This flexibility makes the attack more practical to deploy as it is not heavily constrained by physical layout and victim poses. Attackers have also exploited raw motion data from HMDs for keystroke inference [35], [70], [62], [68], but they rely on the assumption of malware deployment on the target device for data collection. There is another line of work that disrupts user experience and causes motion sickness in VR by jamming or manipulating user tracking with infrared light [48] or malware [9], [43].

Sound Localization. Sound localization has been studied [5], [65]. Some systems used a microphone array pointing 360° around to sense the magnitude of incoming sounds and infer the azimuth of the sound source [27]. Others detected the sound source azimuth by a directional microphone [19]. However, these methods only localize the sound source in a 2D surface and are not suitable for VR. Some works achieved 3D sound source localization using multiple omnidirectional [25], [23], [66] or directional microphones [30], [71], but they require more than two microphones from multiple smartphones or a specialized microphone array. Instead, Heimdall strikes a balance between hardware setup and localization precision.

XII. Conclusion

We present Heimdall, a placement-flexible acoustic keystroke inference attack in VR. Heimdall can differentiate VR keystroke sounds introduced along with the victim's hand translation and rotation movement and map them to the respective keys in any smartphone position and orientation with respect to the victim. Compared with prior keystroke inference attacks, Heimdall is more practical to launch due to the blocked vision of VR users and the elimination of strict placement requirements for the malicious recording devices. Extensive evaluations validate that Heimdall achieves the key inference accuracy of 96.51% and the top-5 accuracy from 85.14% to 91.22% for inferring passwords with 4–8 characters. Moreover, Heimdall is robust under various impacts of attacker capabilities and victim conditions.

Acknowledgment

This work was in part supported by National Science Foundation grant NSF-IIS 2140620 and 2144764.

References

[1] A. Al Arafat, Z. Guo, and A. Awad, "Vr-spy: A side-channel attack on virtual key-logging in vr headsets," in *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2021, pp. 564–572.

[2] D. Asonov and R. Agrawal, "Keyboard acoustic emanations," in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*. IEEE, 2004, pp. 3–11.

[3] J.-X. Bai, B. Liu, and L. Song, "I know your keyboard input: a robust keystroke eavesdropper based-on acoustic signals," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1239–1247.

[4] M. R. Bai and Y.-Y. Lo, "Refined acoustic modeling and analysis of shotgun microphones," *The Journal of the Acoustical Society of America*, vol. 133, no. 4, pp. 2036–2045, 2013.

[5] S. Basu, B. Clarkson, and A. Pentland, "Smart headphones: enhancing auditory awareness through robust speech detection and source localization," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 5. IEEE, 2001, pp. 3361–3364.

[6] Y. Berger, A. Wool, and A. Yeredor, "Dictionary attacks using keyboard acoustic emanations," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 245–254.

[7] C. Boletsis and S. Kongsvik, "Controller-based text-input techniques for virtual reality: An empirical comparison," *International Journal of Virtual Reality (IJVR)*, vol. 19, no. 3, 2019.

[8] R. Carter, "Virtual reality statistics to know in 2021," https://www.xrtoday.com/virtual-reality/virtual-reality-statistics-to-know-in-2021/, 2021, accessed: 2021-09-15.

[9] P. Casey, I. Baggili, and A. Yarramreddy, "Immersive virtual reality attacks and the human joystick," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[10] Y. T. Chan and K. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on signal processing*, vol. 42, no. 8, pp. 1905–1915, 1994.

[11] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Eyetell: Video-assisted touchscreen keystroke inference from eye movements," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 144–160.

[12] P. Cheng, I. E. Bagci, U. Roedig, and J. Yan, "Sonarsnoop: Active acoustic side-channel attacks," *International Journal of Information Security*, pp. 1–16, 2019.

[13] A. Compagno, M. Conti, D. Lain, and G. Tsudik, "Don't skype & type! acoustic eavesdropping in voice-over-ip," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 703–715.

[14] J. Dmochowski, J. Benesty, and S. Affes, "Direction of arrival estimation using the parameterized spatial correlation matrix," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1327–1339, 2007.

[15] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 767–779, 2010.

[16] T. T. Forum, "How to make an omnidirectional microphone directional?" https://taperssection.com/index.php?topic=154955.0, 2012, accessed: 2021-09-15.

[17] M. Funk, K. Marky, I. Mizutani, M. Kritzler, S. Mayer, and F. Michahelles, "Lookunlock: Using spatial-targets for user-authentication on hmds," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.

[18] M. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[19] N. Garg, Y. Bai, and N. Roy, "Owlet: enabling spatial information in ubiquitous acoustic devices," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 255–268.

[20] C. George, M. Khamis, E. von Zezschwitz, M. Burger, H. Schmidt, F. Alt, and H. Hussmann, "Seamless and secure vr: Adapting and evaluating established authentication systems for virtual reality," in *Proceedings of Network and Distributed System Security Symposium.* NDSS, 2017.

[21] K. Ghribi, M. Djendi, and D. Berkani, "A wavelet-based forward bss algorithm for acoustic noise reduction and speech enhancement," *Applied Acoustics*, vol. 105, pp. 55–66, 2016.

[22] Google, "Introducing daydream standalone vr headsets," https://vr.google.com/daydream/standalonevr/, 2021, accessed: 2021-09-15.

[23] K. Guentchev and J. Weng, "Learning-based three dimensional sound localization using a compact non-coplanar array of microphones," in *Proc. AAAI Spring Symposium on Intelligent Environments*, 1998.

[24] HTC, "About the controller," https://www.vive.com/us/support/vive/category_howto/about-the-controllers.html, 2021, accessed: 2021-09-15.

[25] J. Huang, K. Kume, A. Saji, M. Nishihashi, T. Watanabe, and W. L. Martens, “Robotic spatial sound localization and its 3d sound human interface,” in *First International Symposium on Cyber Worlds, 2002. Proceedings.* IEEE, 2002, pp. 191–197.

[26] W. Jin, S. Murali, H. Zhu, and M. Li, “Periscope: A keystroke inference attack using human coupled electromagnetic emanations,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 700–714.

[27] S. Kagami, H. Mizoguchi, Y. Tamai, and T. Kanade, “Microphone array for 2d sound localization and capture,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, vol. 1. IEEE, 2004, pp. 703–708.

[28] H. Landau, “Sampling, data transmission, and the nyquist rate,” *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967.

[29] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser, “Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns,” in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[30] Y. Liang, Z. Cui, S. Zhao, K. Rupnow, Y. Zhang, D. L. Jones, and D. Chen, “Real-time implementation and performance optimization of 3d sound localization on gpus,” in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2012, pp. 832–835.

[31] Z. Ling, Z. Li, C. Chen, J. Luo, W. Yu, and X. Fu, “I know what you enter on gear vr,” in *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2019, pp. 241–249.

[32] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, “Snooping keystrokes with mm-level audio ranging on a single phone,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 142–154.

[33] Y. Liu and Z. Li, “aleak: Context-free side-channel from your smart watch leaks your typing privacy,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 8, pp. 1775–1788, 2019.

[34] L. Lu, J. Yu, Y. Chen, Y. Zhu, X. Xu, G. Xue, and M. Li, “Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 775–783.

[35] S. Luo, X. Hu, and Z. Yan, “Hologger: Keystroke inference on mixed reality head mounted displays,” in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022, pp. 445–454.

[36] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan, “Oculock: Exploring human visual system for authentication in virtual reality head-mounted display,” in *2020 Network and Distributed System Security Symposium (NDSS)*, 2020.

[37] A. Maiti, O. Armbruster, M. Jadliwala, and J. He, “Smartwatch-based keystroke inference attacks and context-aware protection mechanisms,” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 795–806.

[38] W. Mason and R. Marshall, “A tubular directional microphone,” *The Journal of the Acoustical Society of America*, vol. 10, no. 3, pp. 206–215, 1939.

[39] J. Mayor, L. Raya, and A. Sanchez, “A comparative study of virtual reality methods of interaction and locomotion based on presence, cybersickness, and usability,” *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1542–1553, 2019.

[40] Ü. Meteriz-Yıldıran, N. F. Yıldıran, A. Awad, and D. Mohaisen, “A keylogging inference attack on air-tapping keyboards in virtual environments,” in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022, pp. 765–774.

[41] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin, “Moving objects in space: exploiting proprioception in virtual-environment interaction,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 19–26.

[42] Oculus, “Oculus device specifications,” https://developer.oculus.com/resources/oculus-device-specs/, 2021, accessed: 2021-09-15.

[43] B. Odeleye, G. Loukas, R. Heartfield, and F. Spyridonis, “Detecting framerate-oriented cyber attacks on user experience in virtual reality,” 2021.

[44] S. Panda, Y. Liu, G. P. Hancke, and U. M. Qureshi, “Behavioral acoustic emanations: Attack and verification of pin entry using keypress sounds,” *Sensors*, vol. 20, no. 11, p. 3015, 2020.

[45] M. Parker, “Distances between all keys on a standard qwerty keyboard,” http://pi-ratebay.com/files/all_keyboard_distances.txt, 2021, accessed: 2021-09-15.

[46] D. B. Percival and A. T. Walden, *Wavelet methods for time series analysis*. Cambridge university press, 2000, vol. 4.

[47] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt, “Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[48] M. U. Rafique and S. C. Sen-ching, “Tracking attacks on virtual reality systems,” *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 41–46, 2020.

[49] A. S. Rathore, W. Zhu, A. Daiyan, C. Xu, K. Wang, F. Lin, K. Ren, and W. Xu, “Sonicprint: a generally adoptable and secure fingerprint biometrics in smart devices,” in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 121–134.

[50] Reddit, “Usb keyboards are supported with the quest.” https://www.reddit.com/r/OculusQuest/comments/dxmgfo/usb_keyboards_are_supported_with_the_quest_i_am/, 2019, accessed: 2021-09-15.

[51] D. D. Rife and J. Vanderkooy, “Transfer-function measurement with maximum-length sequences,” *Journal of the Audio Engineering Society*, vol. 37, no. 6, pp. 419–444, 1989.

[52] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[53] S. Russell and P. Norvig, “Artificial intelligence: a modern approach,” 2002.

[54] Y. S. Ryu, D. H. Koh, B. L. Aday, X. A. Gutierrez, and J. D. Platt, “Usability evaluation of randomized keypad,” *Journal of Usability Studies*, vol. 5, no. 2, pp. 65–75, 2010.

[55] M. Sabra, A. Maiti, and M. Jadliwala, “Keystroke inference using ambient light sensor on wrist-wearables: a feasibility study,” in *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*, 2018, pp. 21–26.

[56] ——, “Zoom on the keystrokes: Exploiting video calls for keystroke inference attacks,” in *Network and Distributed Systems Security (NDSS) Symposium 2021*, 2021.

[57] Samsung, “Samsung gear vr with controller,” https://www.samsung.com/global/galaxy/gear-vr/, 2021, accessed: 2021-09-15.

[58] D. Schneider, A. Otte, T. Gesslein, P. Gagel, B. Kuth, M. S. Damlakhi, O. Dietz, E. Ofek, M. Pahud, P. O. Kristensson *et al.*, “Reconviguration: Reconfiguring physical keyboards in virtual reality,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 11, pp. 3190–3201, 2019.

[59] L. Schutz, D. Zak, and J. F. Holmes, “Pattern of passenger injury and illness on expedition cruise ships to antarctica,” *Journal of Travel Medicine*, vol. 21, no. 4, pp. 228–234, 2014.

[60] SecLists, “1000 most common passwords,” https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-1000.txt, 2020, accessed: 2021-09-15.

[61] I. Shumailov, L. Simon, J. Yan, and R. Anderson, “Hearing your touch: A new acoustic side channel on smartphones,” *arXiv preprint arXiv:1903.11137*, 2019.

[62] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen, “Going through the motions: Ar/vr keylogging from user head motions,” in *USENIX Security*, 2023.

[63] D. X. Song, D. A. Wagner, and X. Tian, “Timing analysis of keystrokes and timing attacks on ssh.” in *USENIX Security Symposium*, vol. 2001, 2001.

[64] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, “Mole: Motion leaks through smartwatch sensors,” in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 155–166.

[65] J. Wang, K. Zhao, X. Zhang, and C. Peng, “Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained

keystroke localization," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 14–27.

[66] J. Weng and K. Y. Guentchev, "Three-dimensional sound localization from a compact non-coplanar array of microphones using tree-based learning," *The Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 310–323, 2001.

[67] L. Wilkinson, "The grammar of graphics," in *Handbook of computational statistics*. Springer, 2012, pp. 375–414.

[68] Y. Wu, C. Shi, T. Zhang, P. Walker, J. Liu, N. Saxena, and Y. Chen, "Privacy leakage via unrestricted motion-position sensors in the age of virtual reality: A study of snooping typed input on virtual keyboards," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2023, pp. 3382–3398.

[69] W. Yang, X. Dengxiong, X. Wang, Y. Hu, and Y. Zhang, ""i can see your password": A case study about cybersecurity risks in mid-air interactions of mixed reality-based smart manufacturing applications," *Journal of Computing and Information Science in Engineering*, pp. 1–12, 2023.

[70] 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*, 2023.

[71] S. Zhao, S. Ahmed, Y. Liang, K. Rupnow, D. Chen, and D. L. Jones, "A real-time 3d sound localization system with miniature microphone array for virtual reality," in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2012, pp. 1853–1857.

[72] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "Patternlistener: Cracking android pattern lock using acoustic signals," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1775–1787.

[73] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 453–464.

[74] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, pp. 1–26, 2009.