

Peeking the Invisible: Eavesdropping VR Keystrokes Exploiting Headset-embedded Wi-Fi

Jiachun Li, *Member, IEEE*, Yan Meng, *Member, IEEE*, Yuling Chen, Jun Luo, *Fellow, IEEE*, and Haojin Zhu, *Fellow, IEEE*

Abstract—Though leveraging side-channels to eavesdrop keystrokes has been intensively studied, majority of the efforts have targeted “physical and visible” keyboards/pads, even though certain attacking schemes may not require the existence of *line of sight* path towards the targets. Nonetheless, not all keystrokes require a visible keyboard; this is especially so under *virtual reality* (VR) situations, where most of the existing attacks would fail. To this end, we specifically propose VK-Eve for VR keystroke eavesdropping. VK-Eve exploits the Wi-Fi embedded in VR headset to capture the keystroke behavior on invisible keyboards: it leverages the *beamforming feedback information* to avoid the need for hacking Wi-Fi hardware, and it innovates in the cross-domain invariant features called *user-motion-profile* features for sensing the typing behaviors on invisible keyboards. Finally, we implement a VK-Eve prototype and conduct comprehensive evaluations on it. The experimental results show that VK-Eve can achieve an attack accuracy of 87.1% when inferring single keystroke, while it can reach top-100 accuracy of 86.1% when eavesdropping the whole password with 6 digits in real world environments, like logging in VR applications.

Index Terms—Keystroke inference, virtual reality, beamforming feedback information.

I. INTRODUCTION

With the capability of creating a 3D virtual world, virtual reality (VR) offers an immersive experience for users, thus attracting 171 millions of users globally by February 2024 [1]; its market size is estimated to reach \$435.36 billion in 2030 with a compound annual growth rate of 27.5% [2]. When joining a virtual world (via VR headset), users could experience scenarios requiring password based authentications for accomplishing sensitive actions such as account accessing or in-app billing [3]. In contrast to traditional password input

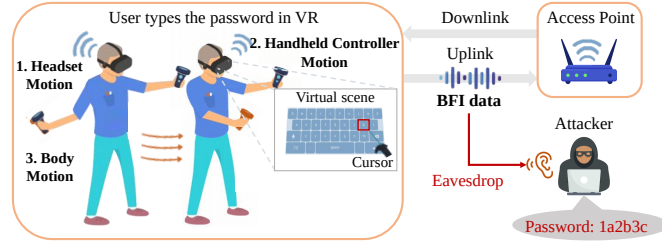


Fig. 1: Wi-Fi based attack (CSI, BFI) when user types password in VR devices.

scenarios featuring a *physical and visible* keyboard, a VR headset offers its user a *virtual* keyboard visible only through the headset’s screen, which has created an illusion to VR users of being largely immune to keystroke inference attacks.

In particular, different from physical keyboards, the virtual keyboard in VR has the following unique properties: i) floating in the air, ii) wide coverage in view angle, and iii) placement varying with viewing angle; all these remain invisible to outsiders including potential attackers. As shown in Fig. 1, typing on a virtual keyboard is also different from conventional typing: i) a user should first move his/her VR headset to look for the hitting points on the keyboard; ii) he/she then moves the virtual cursor via a handheld controller to type a specific key. The entire process involves not only the motion of headset and handheld controller but also the user’s posture changes, which may lead to a series of intertwined motions to be observed by an outsider. All in all, keystrokes in VR scenarios involve fundamentally different *motion characteristics* from those on physical keyboards/pads.

During the past two decades, various side-channel attacks have been proposed to steal the keystroke input via the physical keyboard; they typically fall into three categories: *vision based* [4], [5], [6], [7], [8], *sensor based* [9], [10], [11], [12], [13], [14], [15], [16], [17], and *Wi-Fi based* [18], [19], [20], [21]. Thanks to the aforementioned fundamental differences, directly applying vision and sensor based would surely be ineffective on VR keystrokes, as such behaviors incur distinct vision cues [7], [6] and offer no place to install necessary sensors [9], [10], [11], [12], [13], [14], [15].¹ Consequently, attacks based on Wi-Fi sensing [19], [18] appear to be the only choice that may still be effective on inferring VR keystrokes. As the necessary communication protocol to provide VR users

Jiachun Li is with Guizhou Provincial Laboratory of Big Data, State Key Laboratory of Public Big Data, Guizhou University, Guiyang, Guizhou 550025, China (e-mail: lijie@gzu.edu.cn).

Yan Meng and Haojin Zhu are with the School of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China. (Email: yan_meng@sjtu.edu.cn; zhu-hj@cs.sjtu.edu.cn)

Yuling Chen is with Guizhou Provincial Laboratory of Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou 550025 (e-mail: ylchen3@gzu.edu.cn).

Jun Luo is with the College of Computing and Data Science, Nanyang Technological University (NTU), Singapore 639798 (e-mail: junluo@ntu.edu.sg).

This work was supported by the National Natural Science Foundation of China (U24A20241, 62132013, 62325207, 62302298), Major Scientific and Technological Special Project of Guizhou Province ([2024]014), Scientific and Technological Innovation Platform Research Project of Guizhou Province (CXPTXM[2025]024), Major Research Project of Guizhou Provincial Laboratory (QKHPT-SSYS [2026]ZD015), Guizhou Provincial Basic Research Program (Natural Science) (Qiankehe Foundation MS[2026]207), the Foundation of State Key Laboratory of Public Big Data, and Natural Science Special Project (Special Post) Research Foundation of Guizhou University ([2025]23).

Yan Meng is the corresponding author.

¹Acoustic sensors often need to be very close to the target keyboard [9], [10], while other sensors may even require direct attachment to the keyboard [12], [15], e.g., sharing the same table surface.

with untethered (thus flexible) interfaces, Wi-Fi is ubiquitously equipped with and embedded in a VR headset [22]. Although the confidentiality of wireless communication is safeguarded by encryption schemes (e.g., TLS/SSL, WPA2, WPA3), the physical layer information of Wi-Fi signals, i.e., *channel state information* (CSI) [23] and *beamforming feedback information* (BFI) in its various forms, remains highly correlated with user motions; this has been heavily exploited for side-channel attacks [19], [20], [18].

Within Wi-Fi based side-channel attacks, two sub-categories exist depending on whether the Wi-Fi channel used for inferring keystrokes is the one carrying the keyboard-input data, leading to the differentiation between *in-band* and *out-of-band keystroke inference* (i.e., IKI vs. OKI) [20]. Recently, a few proposals have validated the inefficacy of OKI, pertaining to its inability to sustain interference from the environment [18]. Therefore, dwelling on OKI-related attacks becomes meaningless [19], [24], [25], leaving IKI approaches [20], [18] as the only viable choice. Nonetheless, directly applying IKI-related attacks [20], [18] on virtual keyboard cannot be effective, as such attacks by far target only keypads (e.g., on smartphones); it is highly questionable if they can be adapted to target virtual keyboards with wide coverage in view angle. In particular, since these attacks unanimously assume a one-to-one mapping between finger motions and typed (physical) keys, they would highly likely to fail when adopted to handle the complicated motion characteristics involved in VR keystrokes.

As illustrated in Fig. 1, we embark on a novel attack, coined as VK-Eve, leveraging the VR-headset-embedded Wi-Fi and aiming to effectively infer keystrokes on (invisible) VR keyboards. The rationale behind VK-Eve is twofold: on one hand, exploiting the headset-embedded Wi-Fi enables an attack to switch the perspective from a “bystander” to a “participant” (similar to near-field effects demonstrated in [26]), hence largely getting rid of the interfering motions (e.g., from bodies) while focusing on those from head, arm, and hand; on the other hand, VK-Eve can leverage the wide coverage of VR keyboard (a seemingly obstacle) to gather more CSI or BFI samples incurred by a relatively long sequence of motions required to type a key, thus increasing the dimension of the features that potentially characterize a keystroke. However, translating this rationale into a real world attack still faces three technical challenges: i) How to depict the impact of the headset and the handheld controller, and separate it from the mixed wireless signal? ii) When facing the challenges of new domain information introduced by the motion characteristics in VR, how to mitigate the effects and extract effective cross-domain invariant features to handle the problem? iii) As VK-Eve is the first work in cross-domain attacks in VR, how to verify its effectiveness in various real world VR settings (e.g., inferring keystroke inputs in virtual meeting, gaming)?

VK-Eve tackles the above-mentioned technical challenges as follows. Firstly, we study the *user-keystroke-habit* (UKH) during keystroke in VR. This allows VK-Eve to leverage the motion speed differences between headset and handheld controller motion during typing to separate the mixed wireless signal from the headset and handheld controllers. Then, given the distortions caused by keystroke-unrelated domains, we

construct the *user-motion-profile* (UMP) to characterize the impact of users’ body motion on the wireless signals, while preserving only the keystroke-related cross-domain invariant UMP features. Finally, we implement VK-Eve on our constructed large dataset in real world scenarios, considering and evaluating various factors (e.g., attacking distance, typing speed, user posture, head orientation). Experimental results demonstrate that VK-Eve can achieve an accuracy of 87.1% when inferring single keystroke, and reach a top-100 accuracy of 86.1% when inferring the password in 6 digits with a combination of characters (a-z) and numbers (0-9) in VR. The contributions of this work are summarized as follows:

- *Novel attack.* We introduce VK-Eve, the first cross-domain and hacking-free keystroke eavesdropping attack in VR. Our theoretical analysis models keystroke typing in VR by mapping wireless signals to user motion, demonstrating VK-Eve’s feasibility for keystroke eavesdropping. Case studies further confirm VK-Eve’s superiority over existing methods and its ease of deployment in real-world scenarios.
- *Effective features.* In designation, we collect the broadcasting BFI in Wi-Fi, utilize the user’s keystroke habit for signal separation, and derive a cross-domain invariant feature named UMP features, along with other supplementary features for training and classification.
- *Robust performance.* Experimental results demonstrate that VK-Eve can achieve an effective performance on keystroke inference and password stealing under various impact factors. As for countermeasures, we propose a randomization based scheme, and verify it with comprehensive experiments.

II. BACKGROUND AND MOTIVATION

In this section, we first introduce the unique characteristics of typing keystroke in VR, followed by our attack scenario. Then, we elaborate on the attacking media, including the typical CSI, and emerging BFI with hacking-free advantage. Finally, we show the motivation of VK-Eve, and validate it by conducting case studies.

A. Threat Model of Keystroke Inference in VR

1) *Keystroke Inputting in Invisible Keyboard:* Recently, VR devices have become prevalent in various ranges, including the entertainment, healthcare, and education. The mainstream VR devices can be categorized into two types: screenless viewer devices driven by smartphones and standalone devices, which contain various components, like a micro-controller, LCD display, microphone, loudspeaker, and a pair of handheld controllers. Essentially, when the user types on the screenless viewer devices (e.g., enhanced iPhone 13, enhanced Xiaomi 13 Pro) and the standalone devices (e.g., Meta Quest 2 [27], Pico 4 [28]), it all requires the cooperation of the headset and the handheld controllers.

Transitioning from typing on the physical keyboard to the invisible keyboard in VR causes significant behavior changes. In VR scenarios, virtual keyboard has the following properties: floating in the air, wide angle covering, and placement varying

with viewing angle. Besides, all these remaining are invisible to outsiders including potential attackers. When the user tries to type keystroke on virtual keyboard, it contains the following steps. Due to the wide coverage in view angle, the user should first look for the hitting points, and then move the cursor to strike, which is controlled by the handheld controller. Besides, it is more complex in the whole process for keystroke typing in VR, which not only involves the motion of headset and handheld controller, but also has the body motion influence. In summary, the new characteristics also lead to new challenges for the keystroke inference attack in VR. As mentioned in Section I, the new challenges lead to the failure of existing works (e.g., the vision based, sensor based, Wi-Fi based attack) when directly applying on the keystroke input in VR, which motivates our work.²

2) *Attack Scenario*: As shown in Fig. 2, we optimize the IKI scenario as the overhearing in-band keystroke inference (o-IKI) scenario [18], which means the attacker neither hacks the victim's VR devices nor intrudes the network channel between victim's VR device and Wi-Fi access point (AP). Specifically, the victim enters the virtual world with the VR headset equipped, and he/she connects to a Wi-Fi access point with a public password or directly without Wi-Fi password. It is common in the public places, like the virtual reality experience room, shopping malls, airports and cafes. After that, the victim conducts some activities (e.g., logging accounts, making payment) on VR devices, which is to use handheld controllers to input the password on the virtual keyboard of VR devices. As a result, we can leverage the physical layer information of Wi-Fi signals (e.g., CSI, BFI) emitted by VR devices to perform the side-channel attack for password stealing as follows.

It is worth noting that, when using BFI as the side channel information, the threat of VK-Eve can be further magnified due to the plain-text and broadcasting characteristics in BFI. When deploying VK-Eve, the attacker can directly perform the hacking-free attack only with a monitoring laptop to capture the BFI signal in the air, which does not require the victim to connect to the specific access point. Compared to the IKI scenario [20], since less than 6% of the hardware chips can be hacked for CSI extraction [29], VK-Eve optimizes the threat model without hacking hardware thanks to the BFI data characteristics, which is much more practical and convenient for the real world deployment. Note that, the attacker has no pre-knowledge of the victims VR device type and typing positions in the o-IKI scenario. Besides, no additional malware is installed on the VR devices, which is a weaker assumption for the attacker compared to intruding the VR system.³

B. Wireless Signal Based Attacking Media

1) *Channel State Information*: It is well known that CSI is the most classic media for wireless sensing. It can reveal the propagation information in the channel from the transmitter

²The detailed experimental comparison of existing works and our work is shown in Section II-C4.

³Although some standalone VR devices do not support broadcasting BFI, it is believed to be implemented in the future, and we evaluate the performance of VK-Eve on these VR headsets in Section IV.

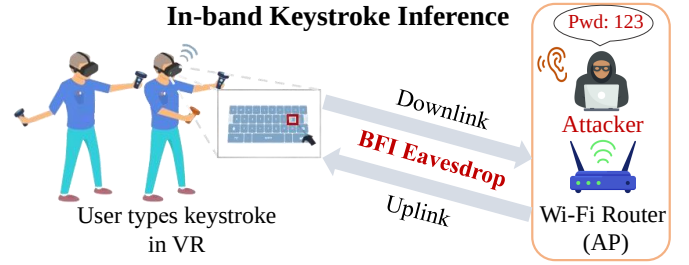


Fig. 2: Threat model of the o-IKI scenario on VR devices.

to the receiver, which involves the environmental effect (e.g., multi-path fading). For a multiple-input and multiple-output (MIMO) Wi-Fi system along with the popularity of IEEE 802.11n standard, it is possible to collect and analyze the CSI signal in this condition. Specifically, as the orthogonal frequency division multiplexing (OFDM) techniques are widely used in MIMO system [25], the resulting subcarrier can be calculated as: $y_{csi} = h_c \times x_{csi} + n$, where x_{csi} , y_{csi} , n represent the transmitted signal, received signal, and the communication noise respectively. c , h denote the c -th subcarrier and the frequency response of the communication channel. Then, the overall CSI samples with N_c subcarriers are denoted as $[h_1, h_2, \dots, h_{N_c-1}, h_{N_c}]$, which is a combination of channel response h_k in a complex number form in k -th subcarrier. Note that, h_k can be represented by $h_k = |h_k| \exp(j \sin(\theta))$, where $|h_k|$ is the channel gain, and θ is for the phase value of the k -th subcarrier.

Since the overall CSI stream h_k is the wireless transmitting gain over the air, the user's motion can result in a unique pattern in CSI, which can be captured by the attacker with the hacked devices and analyzed for keystroke inference (e.g., WindTalker [20]).

2) *Beamforming Feedback Information*: BFI is a compressed version of the CSI samples, which is more robust when facing the changes in communication channel [30], [18]. Different from CSI, BFI is broadcasting over the air when communicating with the Wi-Fi access point, which is more practical for a hacking-free collection. Based on the analysis in CSI, for an $N_t \times N_r$ MIMO system with r subcarriers, a combination of complex matrices can be represented as H in a size of $R \times N_t \times N_r$. Firstly, the beamformer will pre-code the spatial signals for alignment, and the response H is compressed and fed back to the beamformer. We leverage H_k to represent the submatrix of H in the r -th channel, which can be derived by the singular value decomposition as: $H_j^T = Q_j S_j G_j^\dagger$, where Q_j , S_j are the $N_t \times N_t$ unitary matrices, $N_r \times N_r$ diagonal matrix respectively. By collecting the first columns of unitary matrices G_j , we can obtain the complex valued beamforming matrix V_j .

While BFI follows the same principle in wireless sensing as CSI, the following advantages of choosing BFI as the attack media are listed as: plain-text, broadcasting over the air, free of hacking hardware (e.g., WiKi-Eve [18]). As a result, the attacker can leverage such advantages to achieve more covertly keystroke eavesdropping in VR scenarios. Note that, it is compatible to implement our proposed attack VK-Eve

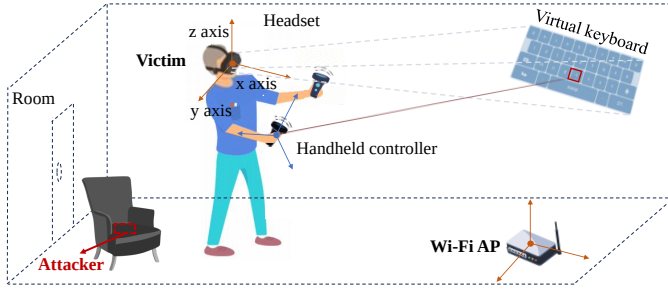


Fig. 3: Modeling keystroke typing in VR theoretically.

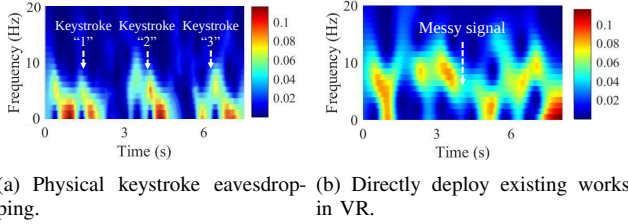


Fig. 4: Deployment of existing works on physical keyboard and virtual keyboard.

based on both CSI and BFI.

C. Motivations of VK-Eve

1) *Theoretical Analysis of VK-Eve's Feasibility:* As illustrated in Fig. 3, to show the feasibility of eavesdropping the keystroke from the wireless signal, we conduct a theoretical analysis. As described in Section II-B, as the wireless signal can be affected by the user's typing motion $m_{x,y,z}$, we can construct a mapping rule $m_{x,y,z} = f_1(h_1, \dots, h_{N_c})$. Similarly, the user's motion will cause the variation of the pitch and roll in handheld controller, which is used to depict the route on the virtual keyboard denoted as $vm_{x,y,z}$. As a result, we can also generate a mapping relationship as $vm_{x,y,z} = f_2(m_{x,y,z})$. Finally, we can find that the information from the wireless signal in the physical world (e.g., CSI, BFI) can be translated into the keystroke typing motion in the virtual world.

2) *Why Existing Works Fail to Function in VR:* By leveraging CSI (e.g., WindTalker [20]), or BFI (e.g., WiKI-Eve [18]) side channel information, existing attacks can eavesdrop the keystroke on physical keyboard. However, as mentioned in Section II-A1, the keystroke eavesdropping on virtual keyboard in VR is significantly different, which will cause the failure of existing works.

To prove the findings, we conduct an experiment for comparison, which includes deploying the existing works (WindTalker [20], WiKI-Eve [18]) to collect the wireless signal (e.g., CSI, BFI) in two conditions: keystroke eavesdropping on physical keyboard, on virtual keyboard. As illustrated in Fig. 4(a), when testing existing works in physical keyboard like smartphone keypad, the spectrogram of the collected signal can clearly show the typing of different keys. However, when deploying in VR in Fig. 4(b), the signal is mixed and messy. It is because the keystroke typing on virtual keyboard is more complex with the motion characteristics. To be specific,

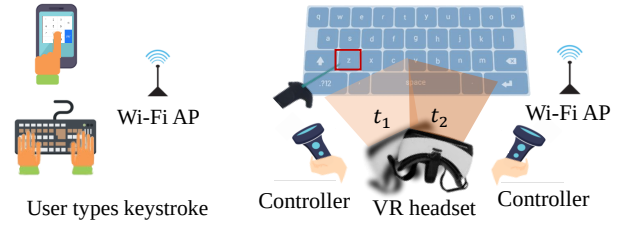


Fig. 5: View changes: "bystander" view in physical keyboard, and "participant" view in VR.

typing in VR will involve more factors: headset motion, handheld controllers motion, positions and orientations.

3) *Workable Rationale behind VK-Eve:* To fix the aforementioned concerns, we find the following rationales: i) we exploit the headset-embedded Wi-Fi to switch the user's view from the "bystander" to a "participant", ii) a longer sequence of motions when typing a key in VR will increase the dimension of features.

As illustrated in Fig. 5(a) and Fig. 5(b), when changing the view from the "bystander" to a "participant", it is much easier for the inference compared to the physical keystroke from the third person view. It can get rid of the interfering motions (e.g., from bodies), and pay more attention to the "participant"-centered motions from head, arm, and hand. Furthermore, as mentioned in Section II-A1, the keystroke typing in VR is different and complex with the collaboration of headset and handheld controllers. The wide range of virtual keyboard and the novel typing style involves a longer sequence of motion when typing in VR, while the time sequence is usually more than 2 seconds compared to physical keystroke (less than 1 second). Thus, it also means more fine-grained data will be generated in both temporal and spatial dimensions.

Based on the rationales, we show our motivation as follows: *As existing works fail when facing the motion challenges in virtual keyboard, our work considers to mitigate the motion impact and exploit the headset-embedded BFI to uncover the corresponding keystroke.* Specifically, we design the user-keystroke-habit (UKH) based algorithm to separate mixed signal into hand-related and head-related components, and then construct a cross-domain invariant feature named user-motion-profile (UMP) feature to uncover the keystroke characteristics in VR, whose details are shown in Section III.

4) *Comparison with Existing Works:* To show the superiority of our work, we compare our work VK-Eve with classic works (i.e., WindTalker [20], WiKI-Eve [18]), and consider the static and dynamical scenarios in VR for evaluation.

Case study on the static scenario. We start the case study from the static scenario, which means the recruiter is instructed to type keystroke by handheld controllers and keep the other part (e.g., headset, body) in a fixed posture. The extracted features by VK-Eve, WindTalker, and WiKI-Eve are shown in Fig. 6. It is observed that the features extracted by WindTalker based on CSI introduce more noise, and also rely on a strong

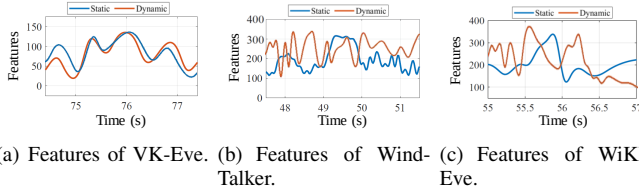


Fig. 6: Comparison of VK-Eve with existing works in static/dynamical scenario.

hacking assumption. VK-Eve, WiKi-Eve both perform well in the static condition.

Case study on the dynamical scenario. We further implement these attacks in dynamical scenario. In particular, it is well known that the virtual keyboard in VR screen is in a wide angle, due to the immersive experience service. As mentioned in Section II-A1, the user will pay more effort to search keys and move the headset more frequently, which will introduce the new domain information (e.g., different positions, orientations). Note that, this dynamical condition is more close to the real world scenario. According to the results in Fig. 6, when the new domain information is involved, the features in WindTalker, and WiKi-Eve are obviously messy and unstable compared to the static scenario. However, VK-Eve still works well in the stability of extracted features, thanks to the designation of the proposed UKH and UMP, which can overcome the challenges of user's motion in VR.

III. DESIGN OF VK-EVE

In this section, we introduce the technical details of VK-Eve, which consists of 4 modules as illustrated in Fig. 7, the data collection module, the pre-processing module, the feature extraction module, and the keystroke output module.

A. Data Collection Module

In this module, the workflow can be divided into 2 steps: attacking time determination, and the BFI extraction.

Attacking time determination. Firstly, VK-Eve should get the knowledge of the MAC address [20], which can be acquired by analyzing the victim's activities and the corresponding traffic simultaneously. To be specific, VK-Eve can map the keystroke typing behavior of the user to the traffic [18] and derive the MAC address with the assist of other hints (e.g., the victim inspects the request made to the payment server or game center). After that, for a given keystroke activity when the user playing VR (e.g., conducting payment, logging in accounts), the non-encrypted IP address to the service center can be captured by VK-Eve. As the experimental results prove that the IP address is not dynamic for a certain time period [18], [20], the attacker constructs an IP address dataset, which can map the IP address to specific application. It is known that users need to authenticate before using VR applications. When opening a VR application, the corresponding IP address associated with the application appears. This allows us to determine the time of attack when the sensitive IP address appears. Additionally, VK-Eve can gather relevant BFI data

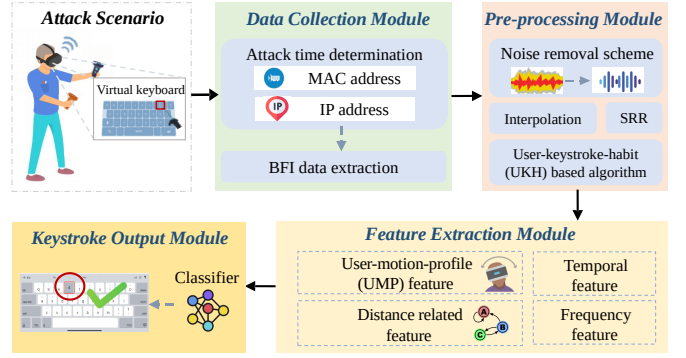


Fig. 7: Overview of the attack: VK-Eve.

related to the user's typing actions by filtering MAC and IP addresses effectively⁴.

BFI extraction. According to the introduction in Section II, the broadcasting BFI can be captured by a monitor smart devices (e.g., personal computers) with Wireshark [31]. It can locate the BFI in "VHT beamforming report" frame, and extract it following the antenna numbers in transmitting side (Tx) and the receiving side (Rx). Thus, based on the two aforementioned steps, VK-Eve can collect BFI which is prepared for attacking.

B. Pre-processing Module

In this module, we design the noise removal scheme, and the interpolation algorithm in VK-Eve for data pre-processing. In particular, targeting on the user habit in VR, we separate the processed data of VR headset and the handheld controller for feature extraction.

1) Noise Removal and Interpolation: In this part, we utilize the moving average filter and temporal convolutional network (TCN) for data de-noising, and leverage the interpolation algorithm (e.g., spline, super-sampling-rate-reconstruction) to optimize the data quality.

Noise removal schemes. Despite we utilize the moving average filter to mitigate the noise, the residual environmental noise still exists. Thus, we leverage the temporal convolutional network (TCN) to fix this issue. TCN contains dilated convolutional layers, which can analyze and filter the noise in time series. We take the first layer as an example. For a collected input BFI denoted as $D[n]$, the layer will conduct the convolution calculation with the kernel vector p using a dilation elements χ .

$$[D *_{\chi} p]_i = \sum_{j=1}^P p_j \cdot D_{i-\chi \cdot j}. \quad (1)$$

Before using TCN, we should optimize the network. We exploit the semi-supervised schemes, which can reduce the cost and user burden for labeling. When implementing, VK-Eve first pre-collects the environmental noise and the "clean"

⁴According to [20], the specific IP address is unique and will be kept relatively stable for one or two weeks, which can be identified clearly even with running multiple background applications simultaneously. Thus, the IP based attacking time determination is reliable for deployment in real world.

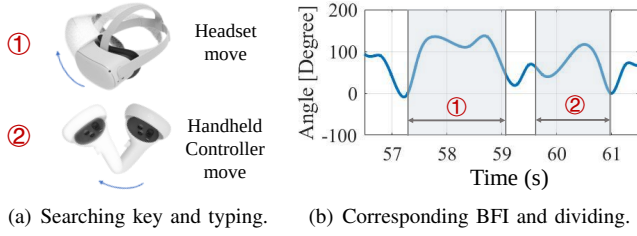


Fig. 8: Keystroke searching and typing process in VR.

BFI of keystroke typing in VR, and constructs the two datasets for training. Then, we exploit the mixed data of BFI with environmental noise and label the “clean” part. After training, the TCN of VK-Eve can use the Adam algorithm and minimize the mean-squared-error (MSE) loss to generate the final output as the filtered BFI samples, coined as $D_F[n]$.

Interpolation algorithm. Since the sampling rate of BFI is usually limited, which is less than 20 samples per second [18], we exploit the interpolation algorithm and the super-sampling-rate-reconstruction to improve the quality of the collected BFI. For the implementation of the interpolation algorithm, instead of the classic linear schemes (i.e., connecting the separated points linearly), VK-Eve uses the spline scheme for interpolation [32]. Additionally, VK-Eve designs a super-sampling-rate-reconstruction (SRR) to add more useful details for the collected BFI curves, which is to reconstruct the signal by using more samples from other time periods [33]. Note that, the processed data after the noise removal and interpolation is notated as $D_\phi[n]$.

2) *UKH Based Signal Separation for VR:* As the virtual keyboard is in wide angle on VR screen due to the immersive experience, the user tends to move the headset to search for the key and type the keystroke. It will result in a mixed aliasing of two signals (e.g., wireless signal affected by the headset, the handheld controller). Thus, we design a novel user-keystroke-habit (UKH) based algorithm to separate the collected BFI. By analyzing the user’s habit, we can find two issues: as shown in Fig. 8, the user usually searches the key with headset motion and then types with the handheld controller; the moving distance of headset and handheld controller is proportional. In this study, we define the searching time of the user (e.g., headset motion time) as t_h , and the typing time (e.g., handheld controller motion time) as t_c , while the moving speed of headset and the controller is denoted as v_h , v_c respectively. Based on the findings, we define a variable named as the proportional ratio (r_δ), which can be derived as $r_\delta = \frac{v_c}{\eta v_h + v_c}$. Note that, the factor η , v_c , v_h can be collected from the training data in pilot study.

According to the proportional ratio (r_δ), VK-Eve can divide the mixed BFI in Fig. 9(a). As the effect of handheld controller is more obvious, we take the BFI of handheld controller as an example, which is notated as $D_{C_i}[n]$. For the first keystroke $D_{C_1}[n]$, VK-Eve divides it as follows:

$$D_{C_1}[n] = \overbrace{D_\phi[1] \oplus \dots D_\phi[\lceil r_\delta L_1 \rceil]}^{v_c} \oplus \overbrace{D_\phi[\lceil r_\delta L_1 \rceil + 1] \oplus \dots D_\phi[L_1]}^{\eta v_h}, \quad (2)$$

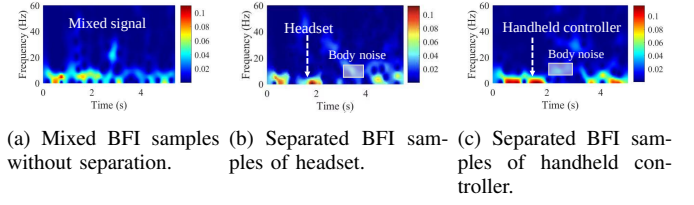


Fig. 9: Spectrogram of the collected BFI by VK-Eve with/without separation.

where L_1 is the length of the whole typing process in the first. It is observed in Fig. 9, compared to the mixed signal, VK-Eve can finally get a separated BFI by our schemes, which is calculated as $D_{C_\phi}[n] = [D_{C_1}[n], \dots, D_{C_N}[n]]$ with N keystrokes in total.

C. Feature Extraction Module

After we obtain the processed BFI, to find the key characteristics of different keystrokes, we extract some effective features, including the temporal features, the frequency features, the user-motion-profile (UMP) features, and the distance related features.

1) *Features Based on Time-frequency Analysis:* In this part, we conduct the time-frequency analysis to extract the temporal features and frequency features.

Temporal features. In this study, we design a temporal feature F_T . Specifically, we divide the collected data in time series. For a given processed BFI $D_{C_\phi}[n]$ with N keystrokes in total, we focus on each component $D_{C_i}[n]$, and extract the temporal features F_T from it, which contains the following metrics: peak value (D_P), the valley value (D_V), area under the curve (D_{AUV}), the average value (D_{AVE}), the roll-off rate (D_{RF}). To ensure the calculation speed, we resample F_T as 128 points, which can be shown as $F_T = [D_P, D_V, D_{AUV}, D_{AVE}, D_{RF}]$.

Frequency features. Besides temporal features, VK-Eve also extracts features in frequency domain to improve the diversity of BFI. Specifically, we exploit the Short Time Fourier Transform (STFT) [34] on the processed data $D_{C_i}[n]$. When implementing STFT, a Hanning window is utilized for processing, while the window size and the overlap is set as 60, and 59 respectively. The cutoff of frequency band is limited as 15 Hz. At last, VK-Eve averages the spectrogram by sampling side and generates a 128 dimension feature F_F .

2) *Characterizing VR Motion Based on UMP Feature:* Inspired by [35], we propose a novel cross-domain invariant feature, namely the user-motion-profile (UMP) feature. In particular, it is to remap the user’s motion in a new coordinate, and exploits the prominent velocity for estimation. The advantage of UMP is that, it can keep stable and resist the effects of motion regardless of the other domain influence, which includes the user’s positions, and orientations in VR interaction. Then, we introduce the extraction of UMP features in detail. Firstly, we define an operator as follows:

$$\mathcal{Q} \otimes \mathcal{P} \triangleq \sum_{i=1}^M \sum_{j=1}^N \mathcal{Q}_{(i,j,*)} \cdot \mathcal{P}_{(i,j)}, \quad (3)$$

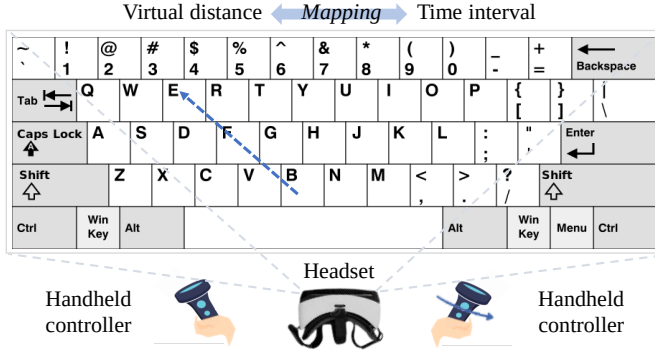


Fig. 10: Impact of distance when typing keystroke in VR.

where $\mathcal{Q} \in R^{M \times N \times L}$ and $\mathcal{P} \in R^{M \times N}$. The operator results are equivalent to multiplying the corresponding element between $\mathcal{Q}$ and $\mathcal{M}$, and calculating the summarized results. After that, VK-Eve defines the designed features UMP as follows:

$$[UMP] = \min_{\Phi} \sum_{i=1}^L |Func_1(M^{(i)}(\Phi), [\Lambda]^{(i)})| + \eta \|\Phi\|_0, \quad (4)$$

where $M^{(i)}(\Phi)$ is the reconstructed data of the i -th link, and Λ is the aggregated data refined from $D_{C_i}[n]$. L and N are the link numbers in total, and the possible velocity components on the axis of x and y . η and $\|\cdot\|_0$ are the sparsity coefficient and the non-zero elements in UMP features respectively. Empirically, we set the optimization function $Func_1$ as the Earth Mover's Distance (EMD) [35].

In the real world implementation, the orientation of the user is always similar, which means most of the body parts (including the headset, and the handheld controller) move along the same direction (the facing direction). Thus, we can derive the sub-part ($M^{(i)}(\Phi)$) as follows:

$$M^{(i)}(\Phi) = Func_2(C^{(i)} * Angle^{(\theta)}) \otimes \Phi, \quad (5)$$

where $C^{(i)}$ is the coefficient matrix when mapping UMP features and $[\Lambda]^{(i)}$, and $Func_2$ is to pick up the most relevant elements among the matrix to optimize the calculation speed. The angle matrix $Angle^{(\theta)}$ is shown as follows:

$$Angle_{i,j}^{(\theta)} = \begin{cases} 1, & |angle(<v_i, v_j>) - \theta| < \zeta \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $<v_i, v_j>$ is the velocity vector of the $<i, j>$ -th element in the UMP features. ζ is the pre-defined threshold according to the motion habits in VR, usually set as 0.05 rad empirically. Finally, by analyzing the velocity weight of the user and optimizing the equation 4, VK-Eve can calculate the UMP features vector, and resample it as 128 dimensions.

3) *Distance Related Feature*: According to the description in the pre-processing module, we can know the moving time interval can reveal the information of typing keystroke. In other words, for a given user, a longer time interval usually shows a longer distance on the virtual keyboard between the last typing key and the next key. Based on these findings, we propose the distance related feature F_D in this subsection.

As shown in Fig. 10, for the j -th user, we target on the typical QWERTY keyboard in VR, and construct the virtual-distance array S_{d_j} , which is derived as follows:

$$S_{d_j} = \omega_j \times \begin{bmatrix} \phi & d_{key_1, key_2} & \cdots & d_{key_1, key_N} \\ d_{key_2, key_1} & \phi & \cdots & d_{key_2, key_N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{key_N, key_1} & d_{key_N, key_2} & \cdots & \phi \end{bmatrix}, \quad (7)$$

where d_{key_i, key_j} represents the virtual-distance of keyboard layout between i -th and j -th keystroke (key_i, key_j), which is adjusted by the user's angle factor ω_j . Besides, VK-Eve also calculates an average time interval threshold as TH_j . Note that, S_{d_j} , ω_j , TH_j can be derived by analyzing the training data in pilot study. After that, when the testing data comes, VK-Eve extracts the tested time interval T_{C_i} by conducting temporal analysis on each D_{C_i} , and normalizes it as $T_C[n] = \frac{1}{TH_j} [T_{C_1}[n], \dots, T_{C_{N-1}}[n]]$. VK-Eve leverages the cosine similarity function S_{cos} to generate the similarity score F'_D as follows:

$$F'_D = S_{cos}(T_C[n], S_{d_j}), \quad (8)$$

Finally, VK-Eve averages F'_D along the x axis and get a 128-dimension vector as the distance related feature F_D .

After generating all the necessary features, VK-Eve aggregates the aforementioned features as $F = [F_T, F_F, UMP, F_D]$. Additionally, to improve the performance, VK-Eve leverages the FEAST toolbox [36] for ranking. Finally, the top k ($k = 50$) features are input to the keystroke output module for classification.

D. Keystroke Output Module

In this study, we leverage the multi-head self attention layers [37] to leverage the inter-dependency among each keys. The advantage of this layer is that it allows the model to jointly combine the information from different representations in various positions, which is suitable for the inference in keystroke. In particular, the designed layer will receive the aggregated features $F = [F_T, F_F, UMP, F_D]$ from the feature extraction module, and generate the attention score z_k as follows:

$$z_k = Func_3 \left(\frac{(F\Omega_Q)(F\Omega_K)^T}{\sqrt{Z}} \right) (F_n\Omega_V), j \in [1, N], \quad (9)$$

where F represents the matrix composed of F_n , and $Func_3$ is the selected function for processing with its result referred as attention scores. $\Omega_Q, \Omega_K, \Omega_V$ represents the learning matrices of query, key, and value. At last, VK-Eve inputs the derived attention score to the final fully connected layer and generates the output result of the predicted keystroke.

IV. EVALUATION

In this section, we will elaborate on the implementation of VK-Eve, and evaluate the performance. To demonstrate the effectiveness of VK-Eve, we evaluate the impact of environmental factors (e.g., typing speed, user, distance) followed by real world implementation on some popular VR applications (e.g., VRChat [38]).

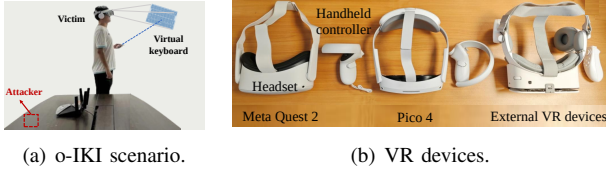


Fig. 11: Scenario of o-OKI and the involved VR devices.

A. Implementation and Settings

System Setup. When implementing VK-Eve, we leverage an Acer TravelMate laptop [39] with an Intel AX210 Wi-Fi NIC [40], which supports the following Wi-Fi protocol: 802.11b/g/n/ac. We set the laptop into the monitor mode while the channel index and the bandwidth is set as 52 and 80 MHz respectively. Then, we utilize the Wireshark to collect the BFI, which is the “VHT beamforming report” in No-ACK frames. After the capturing, Python and Matlab are used for data processing and keystroke classification.

10 users (6 males and 4 females) are recruited to type the password on virtual keyboard in VR. The included VR devices contain 4 VR devices in 2 types: screenless viewer devices driven by smartphones (i.e., enhanced iPhone 13, enhanced Xiaomi 13 Pro) and standalone devices (i.e., Meta Quest 2, Pico 4) as shown in Fig. 11⁵. For the detailed settings of experiments, we verify the performance of VK-Eve in the lab at 9:00 AM. The user is instructed to stand and type on the keyboard in front of his/her headset orientation with a low typing speed as 1 character per second. The attacking device (e.g., personal computer for monitoring BFI) is set with a distance of 1 m. Furthermore, for password and word eavesdropping, we generate a random password list in various lengths with 4, 6, and 8 digits in the appendix, and select the most high frequency word list from Wiki [41]. In summary, 13248 attacking trials are collected and constructed a large dataset for training and testing. The split ratio of the dataset is 8:2.

Evaluation metrics and ethics consideration. Similar to existing works, VK-Eve utilizes the following metrics to measure the performance. For a single keystroke, we utilize the accuracy and Fscore for measurement, and define it as $Accu_k$, and $Fscore_k$. Concerning the password or word eavesdropping, as the attacker can try multiple times, we adopt the top-20, top-50, and top-100 accuracy as the metrics, which is denoted as Top_{p20} , Top_{p50} , and Top_{p100} . Specifically, Top_{pN} accuracy denotes the likelihood of the correct password being included in the candidate password pool. To construct this pool, we compute the product of similarity scores for each key in the password (see equation 9), and subsequently choose the top-N passwords according to these aggregated scores. As for the ethics consideration, the experiments are under the approval of the institutional review board (IRB) of our institutions. In this study, the user is required to perform the password input action on VR devices, which is free of health and privacy risk.

⁵We evaluate the performance of VK-Eve on the standalone VR headset in out-of-band (OKI) scene and it is believed to be useful to be implemented later.

TABLE I: Performance of VK-Eve for password inference.

Digit	Top_{p20}	Top_{p50}	Top_{p100}	Time overhead (s)
4	63.4%	80.6%	91.8%	4.54
6	65.8%	76.4%	88.4%	6.77
8	52.1%	62.7%	78.2%	8.29

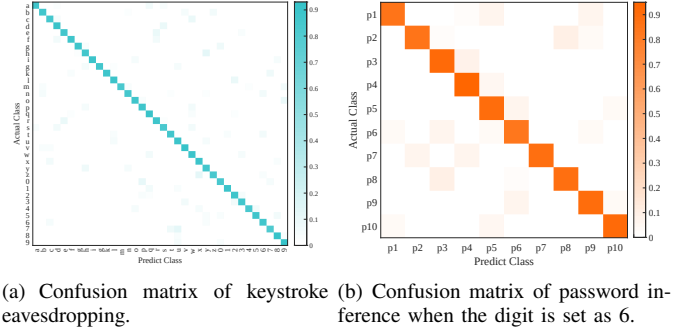


Fig. 12: Confusion matrix of VK-Eve on two tasks.

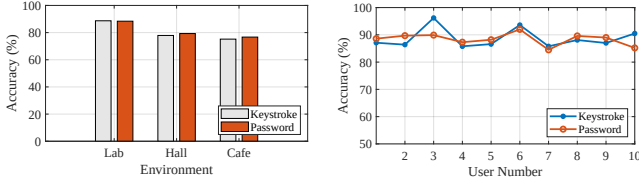
B. Overall Performance

For the overall performance, VK-Eve achieves an accuracy of 88.7% and Fscore of 0.89 on the task of the keystroke eavesdropping. To visualize the results, the confusion matrix of VK-Eve is shown in Fig. 12(a). It is observed that the wrong classification results usually caused by the proximity of key positions in virtual keyboard in VR, like “a” and “s”. For the password stealing task, it is observed in Table I that VK-Eve can achieve an average top-100 accuracy of 86.1%, while the confusion matrix for 10 candidate passwords in Fig. 12(b) also performs well, which proves the effectiveness of our proposed attack VK-Eve.

C. Impact of Various Factors

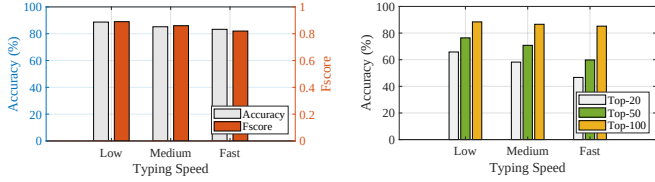
In this subsection, we evaluate the effects of various impact factors on VK-Eve to show the robustness of our proposed attack. Except for the evaluation of the given factors, all the experiments follow the same configurations in Section IV-A.

1) Impact of Environments and Users: As different environment and various users will have an effect on the transmission of BFI, we conduct experiments for measuring. In this study, we set the environment as lab, hall, and cafe, and recruit 10 users. As shown in Fig. 13(a), the performance of VK-Eve will decrease in a noisy environment, like the cafe. It is because there is more noise which may interrupt the communication quality, thereby the BFI will also be affected. However, the performance can still maintain an accuracy of 75.2% and a top-100 accuracy of 76.7% for keystroke eavesdropping and password inference, which demonstrate the usability of our attack. According to the results in Fig. 13(b), when the data from more users are involved for training, VK-Eve can perform better, which means eavesdropping the keystroke more precisely, thanks to more diverse input. In summary, when facing the challenge of various environments and users, the proposed attack still works well, which shows the robustness of the proposed attack VK-Eve.



(a) Accuracy in various environments. (b) Accuracy under different users.

Fig. 13: VK-Eve's performance under various environments and users.



(a) Keystroke eavesdropping results. (b) Password inference results.

Fig. 14: VK-Eve's performance under various typing speed.

2) *Impact of Typing Speed:* Since the collected BFI is highly related to the user's motion, it is believed the typing speed of user can seriously effect the performance of VK-Eve. To investigate this factor, we set the typing speed as 3 conditions: low, medium, and fast, which is 1 character, 2 characters, and 3 characters per second respectively. According to the results in Fig. 14(a) and Fig. 14(b), when the typing speed increases, the performance of VK-Eve decreases about 2.0% on average for the two tasks (i.e., keystroke, password), which is caused by the aliasing issue in BFI. However, VK-Eve can still reach an accuracy of 83.3% for keystroke eavesdropping, and achieve a top-100 accuracy of 85.2% in stealing the password, which proves VK-Eve is robust against the typing speed.

3) *Impact of Device Type:* In this study, we also measure the impact of the device type of VR. As mentioned in Section IV-A, we implement VK-Eve on 4 VR devices: screenless viewer devices with iPhone 13, with Xiaomi 13 Pro, Meta Quest 2, and Pico 4. It is observed in Table II that VK-Eve can get better performance on the screenless viewer devices compared to the standalone device. The reason is that the BFI quality will be better for a "participant"-centered VR devices in o-IKI scene. In summary, VK-Eve can reach an average accuracy of 87.1% and a top-100 accuracy of 86.1% when eavesdropping the keystroke and inferring the password.

4) *Impact of Attacking Distance:* It is well known that the distance between the attacking device and the victim's VR headset is a significant impact factor on BFI [18]. Thus, we set various attacking distances from 0.5 m to 5.5 m with a step of 0.5 m and measure the performance of VK-Eve. According to the results in Fig. 15, when the attacking distance increases from 1 m to 5.5 m, the performance of VK-Eve decreases simultaneously due to the transmission quality. Besides, the measured metrics can still reach above 60.0% for keystroke eavesdropping and password inference even in the

TABLE II: Impact of device type.

Device Type	Task	Accuracy	Fscore	Time overhead (s)
iPhone 13	Keystroke	89.2%	0.89	2.79
	Password	89.7%	0.90	6.74
Xiaomi 13 Pro	Keystroke	88.6%	0.89	2.77
	Password	88.4%	0.88	6.85
Meta Quest 2	Keystroke	88.7%	0.87	2.98
	Password	88.1%	0.88	7.00
Pico 4	Keystroke	88.3%	0.88	2.62
	Password	87.4%	0.86	6.54

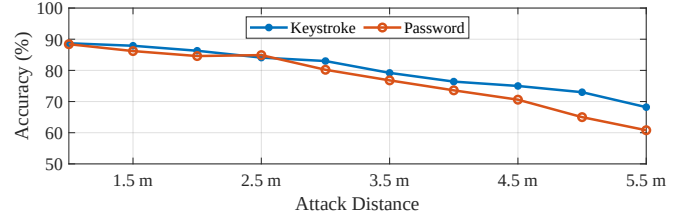


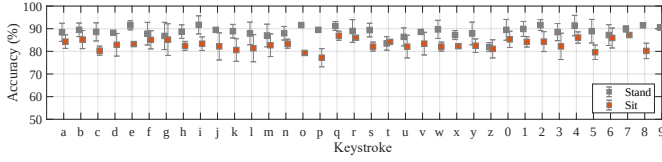
Fig. 15: Impact of attacking distance on VK-Eve.

worst case (i.e., 5.5 m) when deploying VK-Eve. In summary, our proposed keystroke eavesdropping attack is robust and stable when changing the attacking distances.

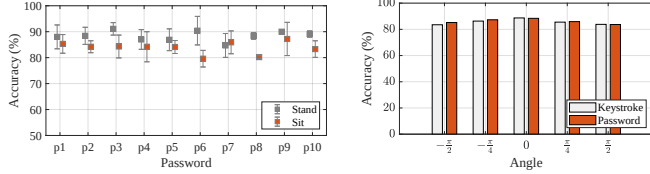
5) *Impact of the User's Posture and Head Orientation:* According to the discussion in Section III-C, the user's posture and head orientation have a great influence on the collected BFI. In this part, we evaluate the two factors as follows.

Firstly, we let the user type keystroke in VR in different postures, standing or sitting. As shown in Fig. 16(a), we can find that VK-Eve performs better when the user types the keystroke in VR while keeping standing. The results in Fig. 16(b) are similar to the password inference task, whose measuring metrics can reach both above 80.0% on average. Then, we set various head orientations for VR headset (i.e., the angles between the orientation and the front), which include: left ($-\frac{\pi}{2}$), left-front ($-\frac{\pi}{4}$), front (0), right-front ($\frac{\pi}{4}$), and right ($\frac{\pi}{2}$). It is observed from Fig. 16(c) that our proposed attack will face a decrease of 5.1% and 4.0% in accuracy when the user changes the angles from the front. Nevertheless, VK-Eve can eavesdrop the keystroke in an accuracy of 85.6%, and infer the password with a top-100 accuracy of 86.1%, when the user changes the head orientations. In a word, the experimental results prove VK-Eve is robust thanks to the design of UKH and UMP, when facing the effects of postures and orientations.

6) *Impact of the Registration Cost:* To investigate the effect of the registration size, the proportion of the training dataset is set as 50.0%, 66.7%, and 80.0%. As shown in Table III, VK-Eve can reach an accuracy of 88.7% for keystroke eavesdropping and a top-100 accuracy of 88.4% for the password inference task, when the proportion is set as 80.0%. Meanwhile, when the proportion increases from 50.0% to 80.0%, the performance gets improved with more inputs from the victim. Additionally, the performance of VK-Eve can still maintain 79.6% and 77.8% for the two tasks even in the worst case (i.e., 50.0%), which is also acceptable. In summary, when facing the challenge of lacking training data, VK-Eve is still



(a) Results of keystroke eavesdropping when user typing in VR with standing or sitting postures.



(b) Results of password inference when user types with standing/sitting. (c) Impact of different head orientation angles.

Fig. 16: VK-Eve's performance when the user changes posture, motion and head orientation.

TABLE III: Impact of registration cost.

Registration Cost	Task	Accuracy	Fscore	Time overhead (s)
50.0%	Keystroke	73.4%	0.73	1.58
	Password	77.8%	0.78	4.82
66.7%	Keystroke	85.7%	0.86	2.42
	Password	84.3%	0.84	5.59
80.0%	Keystroke	88.7%	0.89	2.63
	Password	88.4%	0.88	6.77

useful for keystroke and password inference, which threatens the security of user's privacy in VR.

7) *Impact of the Temporal Stability*: To evaluate the impact of the temporal factor, we implement VK-Eve on 3 different times (i.e., morning, afternoon, evening) in a period of a month. Note that, the training data is all collected in the first time. As shown in Fig. 17, VK-Eve achieves an average accuracy of 73.2% and 72.4% for the two tasks, which include the keystroke eavesdropping and the password inference task. The experimental results prove that our proposed attack is stable and effective over a long time period.

D. Real World Implementation of VK-Eve

After evaluating the performance of VK-Eve with the potential impact factors, we conduct some experiments to show the feasibility of VK-Eve in real world, which contains 4 cases: the word prediction, the unlock patterns, the VR games, and the VR meetings.

1) *Word Prediction*: Considering the user may type and chat with other users on VR devices, which can also be attacked by VK-Eve. According to the settings in Section IV-A, we instruct the recruiters to type the most common 200 english words [41] in VRChat app with the handheld controllers, whose length vary from 1 to 8 digits. Then, we implement VK-Eve to eavesdrop the input words. It is observed in Table IV that VK-Eve achieves an average top-100 accuracy of 86.7% on the most common words when the user types in VR, which shows the attack is still useful

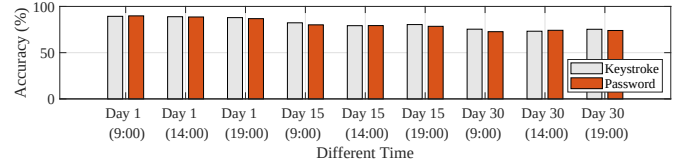


Fig. 17: VK-Eve's performance in different times.

TABLE IV: Results of VK-Eve when predicting words.

Word Digit	Top_{p20}	Top_{p50}	Top_{p100}	Time overhead (s)
1	92.4%	100.0%	100.0%	2.29
2	82.3%	94.2%	96.7%	3.98
3	70.9%	86.5%	93.2%	4.82
4	62.5%	81.3%	90.4%	5.38
5	59.2%	78.9%	86.3%	7.25
6	60.7%	68.8%	80.2%	9.23
7	55.3%	63.1%	76.2%	10.16
8	50.4%	59.9%	70.4%	12.07

in word prediction. Furthermore, it also poses the threat of privacy leakage when chatting in VR.

2) *Unlock Pattern*: When wearing the VR devices, it is common to set an unlock pattern (e.g., graphical unlock pattern, password unlock pattern) for the device itself or some sensitive applications to authenticate the user [42]. Since the password entry is already discussed, we mainly discuss and evaluate the performance of VK-Eve on the graphical unlock pattern in this subsection.

Concerning the graphical locking pattern, the user is instructed to conduct the "point and pinch" operation on the starting dot of the lock pattern, draw the pattern with the handheld controller pinched, and finally release when it is completed. As illustrated in Fig. 18(a), we define 12 locking patterns on a given VR device (i.e., Meta Quest 2) in total, which contains 4 different patterns of 3, 4, and 5 points respectively. According to the results in Fig. 18(b), VK-Eve can achieve an average top-100 accuracy of 87.9% on all unlock patterns. Besides, when the length of unlock pattern increases, it is harder for VK-Eve to attack. Otherwise, the attack performance can maintain 83.1% even in the worst case. Thus, the experiments show that VK-Eve can also be extended successfully in inferring the unlock patterns in VR.

3) *VR Games*: Except for the implementation of VK-Eve on the aforementioned cases, we also deploy the designed attack in VR games, such as the Minecraft. As shown in Fig. 19, the user may type to chat with each other in Minecraft. Following the same settings in Section IV-A, we use VK-Eve to attack and steal the chatting content in Table IV on this occasion. Experimental results in Fig. 20 show that VK-Eve can reach a top-100 accuracy of 88.6% for chatting content inference, which proves the attack is also useful when the user types in popular VR games.

4) *VR Meetings*: Recently, VR meetings are popular with the advantages of convenience and security. We also measure the performance of VK-Eve when attacking popular VR meeting application, Horizon Workrooms. As shown in Fig. 19, in this app, the user can draw the characters on the virtual table in front of it, and the content will be shown on the whiteboard to

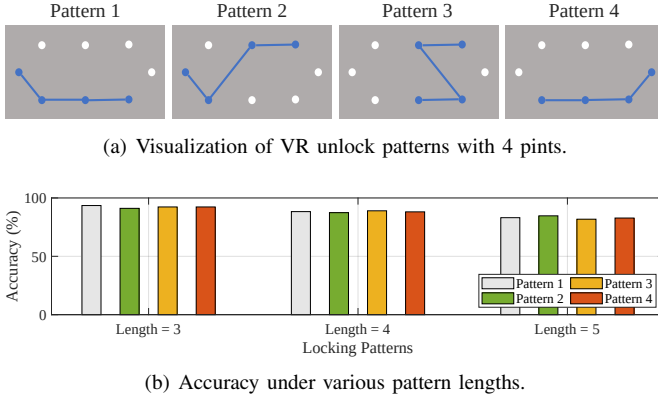


Fig. 18: VK-Eve's performance on inferring unlock patterns.

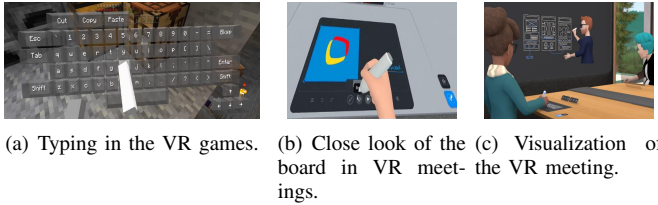


Fig. 19: Typing in VR games and VR meetings.

everyone in the meeting room. For the settings of this situation, we recruit 10 users to write the words in Table IV on the virtual table. VK-Eve is leveraged to infer the user's chatting words. As illustrated in Fig. 20, VK-Eve can eavesdrop the chatting word in a top-100 accuracy of 85.3%, which demonstrates the proposed attack can be generalized into more fields in VR, like the popular VR meetings.

V. COUNTERMEASURE AND DISCUSSION

In this section, we review the state-of-the-art defense schemes and present a novel countermeasure for VK-Eve, named as VK-Defender, followed by the discussion and limitation of our attack VK-Eve.

A. Designation of Defense Schemes

1) *Existing Defense Schemes*: Since the key insight of VK-Eve is to eavesdrop the user's keystroke according to the ubiquitous BFI, existing countermeasures mainly pay attention to covering and protecting the collected BFI. Specifically, the defense schemes include encrypting the traffic, obfuscating the Wi-Fi signal, and randomizing the keyboard.

Firstly, for the traffic encryption [43], it can defense against the attacker to directly capturing the "clean text" of BFI. However, it is not suitable for a Wi-Fi system, which supports multiple users, due to the high cost of deployment. Secondly, considering the signal obfuscation, [43] uses the intelligent reflecting surfaces to scramble the CSI signal and defense against the attack. However, this scheme cannot be deployed due to the regulation of evolving Wi-Fi towards integrated sensing and communications (ISAC), which claims that the Wi-Fi should meet all the user's sensing requirements. Lastly,



Fig. 20: VK-Eve's performance on VR games and meetings.

TABLE V: VK-Eve's performance with/without VK-Defender.

Defense	Task	Accuracy	Fscore	Time overhead (s)
With VK-Defender	Keystroke	23.1%	0.24	3.08
	Password	20.8%	0.20	7.68
Without VK-Defender	Keystroke	88.7%	0.89	2.63
	Password	88.4%	0.88	6.77

considering the defense schemes of keyboard randomization [44], since the position of key is randomly changed, it can obfuscate the collected wireless signal (e.g., CSI, BFI) obviously, which can protect the user's privacy.

2) *Our Countermeasures: VK-Defender*: To defense against VK-Eve, we design a randomization based scheme named VK-Defender, and conduct the experiments as follows. Specifically, VK-Defender will generate 3 random keyboard layouts adaptively for each user, and the user can select one as he/she likes. Then, VK-Eve is deployed to eavesdrop the keystroke with and without the protection of VK-Defender.

It is observed in Table V that the attack performance of VK-Eve decreases 65.0% and 67.6% for keystroke eavesdropping and password inference, which demonstrate the effectiveness of our proposed countermeasures. In this study, we also conduct the security analysis of our schemes under an adaptive attacker. Assuming that the attacker knows the password content but not the randomized keyboard mapping, the attacker must infer the mappings of all distinct characters appearing in the password. Let n denote the number of distinct characters and $|\Sigma|$ denote the keyboard alphabet size. The attack success probability can be derived as follows:

$$\Pr[\text{success}] = \frac{1}{P(|\Sigma|, n)} = \frac{(|\Sigma| - n)!}{|\Sigma|!}, \quad (10)$$

where $P(|\Sigma|, n)$ denotes the number of possible character to key mappings. The above result indicates that the attack success probability decreases rapidly as the number of distinct characters increases, providing strong protection against adaptive attackers.

Furthermore, to ensure the cost of our defense is light weighted, we set a questionnaire for the victim for investigation. As listed in Table VI, we prepare 4 questions for user study. We collected answers from 10 volunteers, which contain 5 males and 5 females. Note that, 80.00% of the participants is familiar with VR devices and have played VR devices for more than 1 month. The score of ranking is set from 1 to 5, in which 1 is for "strongly disagree" and 5 represents "strongly agree".

We define the following metrics to measure the results in questionnaire, which are the average value λ_{ave} and the median value λ_{med} . As for the voting result, we use $\Lambda =$

TABLE VI: Questionnaire in use study.

Num	Questions
1	Is VK-Defender of high security?
2	Is it comfortable when using VR devices with VK-Defender deployed?
3	Is it acceptable for the time delay introduced by VK-Defender?
4	Do you like VK-Defender in VR devices?

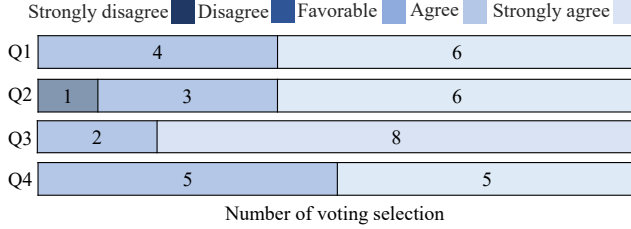


Fig. 21: User study results from questionnaire.

$[\lambda_1, \dots, \lambda_4]$ for representation, where λ_i denotes the overall results from all the recruiters for the i -th question. Concerning each λ_i , it consists of a 2-tuple $\lambda_i = [\lambda_{ave_i}, \lambda_{med_i}]$, where λ_{ave_i} , λ_{med_i} represent the aggregated averaging value and median value from all participants respectively. The statistical results is : $\lambda_1 = (4.6, 5.0)$, $\lambda_2 = (4.5, 5.0)$, $\lambda_3 = (4.8, 5.0)$, $\lambda_4 = (4.5, 4.5)$, which shows that most users agree with using VK-Defender is acceptable to the introduced cost. In summary, experimental results in Table V and the user study from questionnaire in Fig. 21 prove this defending approach VK-Defender can raise the entry barrier for keystroke eavesdropping attack, while it is also user-friendly and practical to be deployed in the real world.

B. Discussion and Limitations

1) *Discussion of Different Inputting Manner:* Different inputting manners can also affect the performance of VK-Eve. Thus, we mainly consider two impact factors in this study, which is the various keyboard layout, and the popular hand-typing mode in VR.

Different keyboard layouts. Although the most widely used keyboard in VR is the typical QWERTY keyboard in Fig. 22, there are still many different layout in other VR systems. As illustrated in Fig. 22 from Pico 4, it is a separate keyboard with the number and character input on two sides. To show the usability of our work, we verify the performance of VK-Eve on the separate keyboard with the same settings. It is observed in Table VII that VK-Eve can reach a good performance when eavesdropping the keystroke and password when the layout changes, which shows the generality of VK-Eve.

Hand-typing mode in VR. As shown in Fig. 22(c), to release the user burden of holding the handheld controller, there has been a trend that commercial VR devices support the hand-tracking mode for typing keystrokes [45]. We also test the performance of VK-Eve on this occasion. Experimental results in Table VII show that it will face a decrease due to more complex hand-typing gestures, which are confusing to identify just by the BFI signal. However, it can still maintain an

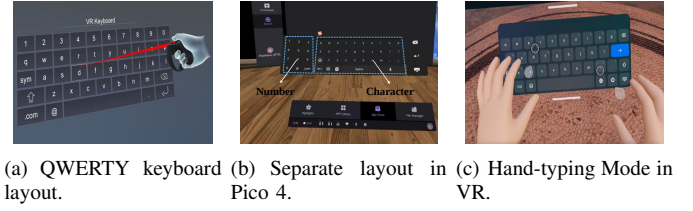


Fig. 22: Different keyboard layouts and hand-typing mode in VR devices.

TABLE VII: VK-Eve's performance on different layouts and hand-typing mode.

Inputting Factors	Task	Accuracy	Fscore	Time overhead (s)
QWERTY	Keystroke	86.4%	0.90	2.17
	Password	89.2%	0.88	6.52
Separated	Keystroke	84.1%	0.83	2.70
	Password	80.9%	0.77	7.08
Hand-typing Mode	Keystroke	76.3%	0.75	2.42
	Password	71.4%	0.72	6.88

accuracy of 76.3% and 71.4% in the keystroke and password tasks, which demonstrates VK-Eve is practical and workable to the challenges of different inputting manners.

2) *Discussion of Unseen Users and Devices:* Similar to existing works [20], [18], VK-Eve requires the training process to involve the samples from all the users and devices. However, considering that this training process is not always feasible, we evaluate the performance of VK-Eve when some users or devices do not participate in the training process. It is observed in Fig. 23(a) and Fig. 23(b) that when facing unseen users and devices, the performance of VK-Eve when eavesdropping the keystroke will decrease to 73.5% and 66.8%, respectively. We find that VK-Eve can achieve a higher cross user transfer performance when users share similar body posture, interaction habits, and movement patterns. We also observe that the transfer performance is higher within the same device category. This is because smartphone driven VR devices and standalone VR headsets share more similar deployment settings and interaction mechanisms. However, the performance remains acceptable in deployment, which proves that our attack is feasible and robust when handling unseen users and devices.

3) *Discussion of Realistic constraints:* In practical deployments, repeated login failures usually trigger account lockout mechanisms, limiting the number of online password attempts. Such account lockout mechanisms are common in mainstream VR applications (e.g., VRChat, Rec Room, Meta Horizon Workrooms, Spatial, Beat Saber, and Minecraft), where users are typically allowed only 5 to 10 consecutive login attempts before temporary lockout is triggered, and the lockout duration may progressively increase from several minutes to tens of minutes. Therefore, besides the top-100 results, we further evaluate VK-Eve using top-5 and top-10 metrics. The results show that VK-Eve achieves attack success rates of 40.6% and 51.8% for top-5 and top-10 predictions, respectively, demonstrating a considerable threat even under

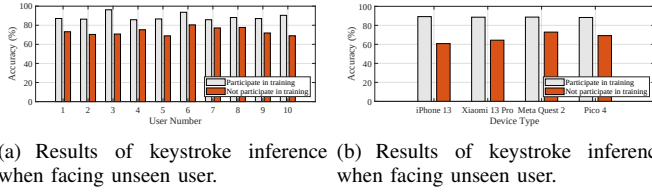


Fig. 23: Performance of VK-Eve when handling unseen user and device.

realistic rate-limiting policies. Furthermore, many real-world systems enforce password composition policies, and attackers may obtain dictionary priors regarding password structures. When the attacker knows whether each password position is a letter or a digit, the attack performance further improves to 50.2%, 62.8%, and 74.5% for top-5, top-10, and top-20 accuracy, respectively. These results indicate that VK-Eve remains effective under practical online attack constraints and can benefit from realistic password policy priors.

4) *Limitations of VK-Eve*: The limitations of VK-Eve mainly include the following parts: limited device types, the stability of BFI, the signal obfuscation, the sampling rate, and the user habit.

Limited device types and keystroke types. In the evaluation part, we mainly consider 4 types of VR devices, which contain screenless viewer devices (i.e., enhanced iPhone 13, enhanced Xiaomi 13 Pro) and the standalone devices (i.e., Meta Quest 2, Pico 4). However, there are still other types of VR devices, like the Apple Vision Pro, and HTC Vive. For further study, we will explore the performance of VK-Eve on other commercial VR devices. Besides, as the functional key's typing will introduce more noise to the BFI signal's quality, we only consider the single keystroke from "a" to "z", and "0" to "9", and the coherent password, while the evaluation with functional key (e.g., "Tab", "Ctrl", "Backspace", "Shift", "Enter", "Space") will be left for further study.

BFI signal's quality. The quality of the BFI signal is also a drawback of our attack, which mainly reflects on the temporal stability, signal obfuscation and sampling rates. Concerning the temporal stability, as shown in Section IV-C, VK-Eve can still be effective in a time period of a month, but it also faces a decrease in attacking performance. To fix this issue, we consider updating the training data for attacking regularly at an acceptable cost. For the signal obfuscation, it can come from multiple users, multiple access points, especially other noise signals. Specifically, when the noise signal affects the channel of the monitoring devices, the quality of BFI will significantly decrease, thereby the performance of VK-Eve decreases. Besides, due to the limitation of the protocol [18], the limited sampling rate of BFI is also a problem. For an extreme consideration, if the user types the keystroke really quickly (e.g., 3 characters per second), the collected BFI will mixed in the time domain resulting by a limited sampling rate. In this study, VK-Eve uses the interpolation algorithm and the super-sampling-rate-reconstruction as a solution, but it still adds the cost of processing. Thus, the open questions will be left for further study.

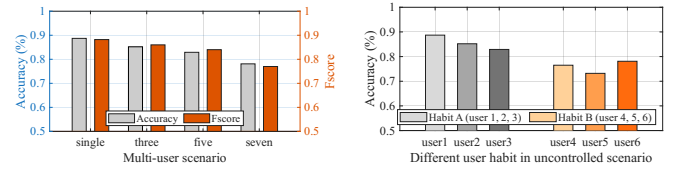


Fig. 24: Performance of VK-Eve when handling uncontrolled VR scenarios.

Multi-user and different user habits in uncontrolled scenarios. To further validate the robustness of VK-Eve beyond controlled conditions, we additionally evaluate two impact factors, the multi-user crowded environments and uncontrolled user habits, which are common in real-world VR deployments. The detailed results are presented in Fig. 24(a) and Fig. 24(b). For the multi-user crowded evaluation, we vary the number of simultaneous users from one to seven to measure the impact of multiple users. The results in Fig. 24(a) show the performance of VK-Eve decrease, mainly due to accumulated wireless noise and body shadowing effects. However, VK-Eve maintains an average accuracy of 83.7% with an Fscore of 0.83 across all configurations, and even with seven concurrent users, the accuracy remains above 78.1%. For different user habits in the uncontrolled VR scenarios, we recruit six users with distinct keystroke movement styles, including straight and curved trajectories styles. According to the results in Fig. 24(b), the curved motion patterns cause more signal fluctuation and lead to lower accuracy compared to straight movements, yet VK-Eve still achieves an average accuracy of 75.9% under this challenging condition. In summary, the experimental results demonstrate that VK-Eve maintains effective performance under diverse and practical uncontrolled conditions.

VI. RELATED WORKS

In this section, we show existing works about the keystroke eavesdropping in vision, sensor, and Wi-Fi, followed by VR keystroke attacks.

A. Vision Based Attacks

Vision based attack is the most intuitive way to eavesdrop the keystroke. [7], [6] directly attack the screen display to analyze and eavesdrop the keystroke. Based on this pilot study, researches exploit the side-channel scheme to make the attack more practical, which analyzes the physical changes instead of directly observing the screen. For instance, the attacker can eavesdrop the keystroke by capturing the shadow changes [8]. Except for directly observing the screen, existing attacks can also predict the keystroke based on the biometric related videos. For example, the eye tracking and the finger motion [46], [5], [47] reveal the keystroke information, which can be used as a side-channel data. In summary, the state-of-the-art works need to observe the video for the whole keystroke process of the victim, which can not be deployed stealthy and practically in real world.

B. Sensor Based Attacks

Researches also shows that the sensor data can also be leveraged as a side-channel information to eavesdrop the keystroke. The sensor based attacks can be categorized into the intra attack and the inter attack. Intra attack [10], [13] exploits the acoustic data, the accelerometer and gyroscope data in the smart devices (e.g., smartphones) to predict the typing keystroke on itself. Similarly, inter attack [9], [15], [14], [48] can collect and use the sensor data on smart devices (e.g., smartphones, smartwatches) to detect the targeted keystroke on a nearby physical keyboard. However, these works both require the hacking of the smart device, and need to attack in a limited distance (i.e., line-of-sight region).

C. Wi-Fi Based Attacks

Wi-Fi based attack is served as the mainstream manner for keystroke eavesdropping, which is based on the unique advantages like stealthiness, convenience, and effectiveness. It contains the electromagnetic signal based attack, and the radio signal based attack.

Concerning the electromagnetic signal based attack, [49] utilizes the electromagnetic emissions during communication to predict the typing keystroke. Additionally, [11], [50] also show that the emitted electromagnetic signal can be used to eavesdrop the keystroke on touchscreen. Besides, the electromagnetic changes in charging power [51] is also vulnerable to be eavesdropped and leveraged to predict the typing keystroke. As for the radio signal based attack, [19], [20] present the CSI based keystroke eavesdropping method for the in band scenario, and out of band scenario respectively. Then, [52] utilizes the linguistic analyzing system to enhance the CSI attacking system. Also, [53] exploits the spatio temporal analysis to improve the performance of keystroke eavesdropping. Besides the typical CSI, [18], [54] shows a BFI based attacking scheme to infer victim's keystroke on the smartphones.

D. Keystroke Inference Attack in VR

With the development of VR techniques, researches point out the security vulnerabilities, especially the keystroke eavesdropping in VR. [55] utilizes the computer vision schemes to collect the side-channel information, which can infer the password input on VR. [45] leverages the video clips of the victim's hand gestures on the virtual screen to decipher the typed text. [56], [57], [58], [59], [60] capture the sensor data from the VR headset and handheld controller to analyze the user's activities (e.g., head motion, gesture, keystroke). [16] leverages subtle head movements to infer users' keystrokes on virtual keyboards through cross-modal analysis. [17] combines hand-tracking sensor data with side-channel information to infer air-pinch keystrokes on virtual keyboards. [61], [62] leverage the controller acoustic emanation and the website traffic to conduct the keystroke inference attack in VR. [25], [24] leverage the CSI samples to eavesdrop the user input, like gestures, keystrokes. [21] exploits unintended WiFi packet leakage from VR devices to perform the cross-building keylogging attacks. To the end, the state-of-the-art works

face the challenge of strong assumption, hardware hacking requirement, line-of-sight limitation. Our attack VK-Eve is the first BFI based cross-domain keystroke eavesdropping attack for VR, which is more practical, suitable, and effective.

VII. CONCLUSION

In this paper, we propose VK-Eve as the first BFI based cross-domain keystroke eavesdropping attack for VR. Specifically, VK-Eve exploits the Wi-Fi embedded in VR headset to capture the keystrokes behavior on invisible keyboards in VR: we collect the beamforming feedback information, design user-keystroke-habit (UKH) algorithm and cross-domain invariant features called user-motion-profile (UMP) features in VK-Eve. Various experiments are conducted to evaluate the impact of various factors to prove the usability of our work. Moreover, we deploy VK-Eve for the specific scenarios (e.g., VR games, VR meetings) in real world implementation. In conclusion, VK-Eve is a practical, suitable, and effective attack to steal the password in VR, which pose a serious security threat for the further research.

REFERENCES

- [1] G2, "Virtual reality user statistics," <https://www.g2.com/articles/virtual-reality-statistics>, 2024.
- [2] G. view research, "Virtual reality market," <https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market>, 2023.
- [3] J. M. Jones, R. Duezguen, P. Mayer, M. Volkamer, and S. Das, "A literature review on virtual reality authentication," in *Human Aspects of Information Security and Assurance: 15th IFIP WG 11.12 International Symposium, HAISA 2021, Virtual Event, July 7–9, 2021, Proceedings 15*. Springer, 2021, pp. 189–198.
- [4] J. Sun, X. Jin, Y. Chen, J. Zhang, Y. Zhang, and R. Zhang, "Visible: Video-assisted keystroke inference from tablet backside motion," in *NDSS*, 2016.
- [5] 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.
- [6] Q. Yue, Z. Ling, X. Fu, B. Liu, W. Yu, and W. Zhao, "My google glass sees your passwords," *Proceedings of the Black Hat USA*, 2014.
- [7] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, "A fast eavesdropping attack against touchscreens," in *2011 7th International Conference on Information Assurance and Security (IAS)*, 2011, pp. 320–325.
- [8] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 1403–1414.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] A. Maiti, O. Armbruster, M. Jadhwal, and J. He, "Smartwatch-based keystroke inference attacks and context-aware protection mechanisms," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 2016, pp. 795–806.
- [13] L. Cai and H. Chen, "{TouchLogger}: Inferring keystrokes on touch screen from smartphone motion," in *6th USENIX Workshop on Hot Topics in Security (HotSec 11)*, 2011.

- [14] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "When good becomes evil: Keystroke inference with smartwatch," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1273–1285.
- [15] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 551–562.
- [16] S. Lee and W. Choi, "Eyes on your typing: Snooping finger motions on virtual keyboards," in *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025, pp. 4340–4355.
- [17] L. Zhou, J. Luo, X. Gong, J. Pei, W. Liu, and H. Wen, "Airsy: An air-pinch keystroke inference attack on virtual keyboards in vr systems," *IEEE Transactions on Consumer Electronics*, 2026.
- [18] J. Hu, H. Wang, T. Zheng, J. Hu, Z. Chen, H. Jiang, and J. Luo, "Password-stealing without hacking: Wi-fi enabled practical keystroke eavesdropping," *Proc. of the 30th ACM CCS*, pp. 1–14, 2023.
- [19] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 90–102.
- [20] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1068–1079. [Online]. Available: <https://doi.org/10.1145/2976749.2978397>
- [21] C. Khalili and N. S. O. Abari, "Xr devices send wifi packets when they should not: Cross-building keylogging attacks via non-cooperative wireless sensing," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2026.
- [22] M. Speicher, "Shopping in virtual reality," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 1–2.
- [23] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, 2011.
- [24] Z. Fu, J. Xu, Z. Zhu, A. X. Liu, and X. Sun, "Writing in the air with wifi signals for virtual reality devices," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 473–484, 2019.
- [25] 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.
- [26] J. Hu, T. Zheng, Z. Chen, H. Wang, and J. Luo, "MUSe-Fi: Contactless MUti-person SEnsing Exploiting Near-field Wi-Fi Channel Variation," in *Proc. of the 29th ACM MobiCom*, 2023, pp. 75:1–15.
- [27] Meta, "Meta quest 2." <https://www.meta.com/quest/products/quest-2/>, 2023.
- [28] Pico, "Pico 4." <https://www.picoxr.com/sg/products/pico4/>, 2023.
- [29] C. Wu, X. Huang, J. Huang, and G. Xing, "Enabling ubiquitous wifi sensing with beamforming reports," in *Proceedings of the ACM SIGCOMM 2023 Conference*, ser. ACM SIGCOMM '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 20–32. [Online]. Available: <https://doi.org/10.1145/3603269.3604817>
- [30] K. F. Haque, M. Zhang, F. Meneghello, and F. Restuccia, "Beamsense: Rethinking wireless sensing with mu-mimo wi-fi beamforming feedback," *arXiv preprint arXiv:2303.09687*, 2023.
- [31] M. Stute, A. Heinrich, J. Lorenz, and M. Hollick, "Disrupting continuity of apple's wireless ecosystem security: New tracking,{DoS}, and {MitM} attacks on {iOS} and {macOS} through bluetooth low energy,{AWDL}, and {Wi-Fi}," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3917–3934.
- [32] H. Behjat, Z. Doğan, D. Van De Ville, and L. Sörnmo, "Domain-informed spline interpolation," *IEEE Transactions on Signal Processing*, vol. 67, no. 15, pp. 3909–3921, 2019.
- [33] Y. Huang, K. Chen, Y. Huang, L. Wang, and K. Wu, "Vi-liquid: unknown liquid identification with your smartphone vibration," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 174–187.
- [34] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-based practical smartphone eavesdropping with built-in accelerometer," in *NDSS*, vol. 2020, 2020, pp. 1–18.
- [35] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3.0: Zero-effort cross-domain gesture recognition with wi-fi," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8671–8688, 2022.
- [36] J. Zhang, Z. Wang, X. Ji, W. Xu, G. Qu, and M. Zhao, "Who is charging my phone? identifying wireless chargers via fingerprinting," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2992–2999, 2020.
- [37] Y. Xu, H. Huang, C. Feng, and Y. Hu, "A supervised multi-head self-attention network for nested named entity recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 16, 2021, pp. 14 185–14 193.
- [38] D. Saffo, C. Yildirim, S. Di Bartolomeo, and C. Dunne, "Crowdsourcing virtual reality experiments using vrchat," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–8. [Online]. Available: <https://doi.org/10.1145/3334480.3382829>
- [39] A. Corporation, "Acer travelmate laptops for business," <https://www.acer.com/gen/laptops/travelmate/>, 2023.
- [40] I. Corporation, "Intel wi-fi 6 ax201." <https://www.intel.sg/content/www/xa/en/products/sku/130293/intel-wifi-6-ax201-gig/specifications.html>, 2023.
- [41] Wikipedia, "Most common words in english." https://en.wikipedia.org/wiki/Most_common_words_in_English, 2023.
- [42] S. Stephenson, B. Pal, S. Fan, E. Fernandes, Y. Zhao, and R. Chatterjee, "Sok: Authentication in augmented and virtual reality," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 267–284.
- [43] P. Staat, S. Mulzer, S. Roth, V. Moonsamy, M. Heinrichs, R. Kronberger, A. Sezgin, and C. Paar, "Irshield: A countermeasure against adversarial physical-layer wireless sensing," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1705–1721.
- [44] I. Griswold-Steiner, N. Diarrassouba, S. Arangath, and A. Serwadda, "User perceptions of defensive techniques against keystroke timing attacks during password entry," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–6.
- [45] S. R. K. Gopal, D. Shukla, J. D. Wheelock, and N. Saxena, "Hidden reality: Caution, your hand gesture inputs in the immersive virtual world are visible to all!" in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 859–876. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity23/presentation/gopal>
- [46] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, "Beware, your hands reveal your secrets!" in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 904–917.
- [47] Z. Yang, Y. Chen, Z. Sarwar, H. Schwartz, B. Y. Zhao, and H. Zheng, "Towards a general video-based keystroke inference attack," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 141–158.
- [48] 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*, ser. MM '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1239–1247. [Online]. Available: <https://doi.org/10.1145/3474085.3475539>
- [49] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *Proceedings of the 18th Conference on USENIX Security Symposium*, ser. SSYM'09. USA: USENIX Association, 2009, pp. 1–16.
- [50] H. Shan, B. Zhang, Z. Zhan, D. Sullivan, S. Wang, and Y. Jin, "Invisible finger: Practical electromagnetic interference attack on touchscreen-based electronic devices," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1246–1262.
- [51] P. Cronin, X. Gao, C. Yang, and H. Wang, "Charger-Surfing: Exploiting a power line Side-Channel for smartphone information leakage," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 681–698. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity21/presentation/cronin>
- [52] S. Fang, I. Markwood, Y. Liu, S. Zhao, Z. Lu, and H. Zhu, "No training hurdles: Fast training-agnostic attacks to infer your typing," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1747–1760.
- [53] E. Yang, Q. He, and S. Fang, "Wink: Wireless inference of numerical keystrokes via zero-training spatiotemporal analysis," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 3033–3047.
- [54] H. Wang, J. Hu, T. Zheng, J. Hu, Z. Chen, H. Jiang, Y. Zheng, and J. Luo, "Muki-fi: Multi-person keystroke inference with bfi-enabled wi-fi sensing," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2024.
- [55] 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.
- [56] J. Lee, H. Kim, and K. Lee, "Vrkeylogger: Virtual keystroke inference attack via eavesdropping controller usage pattern in webvr,"

Computers & Security, vol. 134, p. 103461, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404823003711>

- [57] S. Luo, X. Hu, and Z. Yan, “Holologger: 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.
- [58] C. Slocum, Y. Zhang, N. Abu-Ghazaleh, and J. Chen, “Going through the motions: AR/VR keylogging from user head motions,” in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 159–174. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity23/presentation/slocum>
- [59] 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)*, 2023, pp. 3382–3398.
- [60] Ü. Meteriz-Yıldiran, N. F. Yıldiran, 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)*, 2022, pp. 765–774.
- [61] S. Luo, A. Nguyen, H. Farooq, K. Sun, and Z. Yan, “Eavesdropping on controller acoustic emanation for keystroke inference attack in virtual reality,” in *The Network and Distributed System Security Symposium (NDSS)*, 2024.
- [62] Z. Su, K. Cai, R. Beeler, L. Dresel, A. Garcia, I. Grishchenko, Y. Tian, C. Kruegel, and G. Vigna, “Remote keylogging attacks in multi-user VR applications,” in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 2743–2760. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity24/presentation/su-zihao>



Jiachun Li (Member, IEEE) received the B.Sc. degree from Huazhong University of Science and Technology, Wuhan, China, in 2020, and the Ph.D. degree from the NSEC Lab, Shanghai Jiao Tong University, Shanghai, China, in 2025, supervised by Haojin Zhu. He was a Visiting Scholar with Nanyang Technological University, Singapore, from 2023 to 2024, supervised by Jun Luo. He is currently an associate professor with Guizhou Provincial Laboratory of Big Data, State Key Laboratory of Public Big Data, Guizhou University, Guiyang, China. His

research interests include embodied intelligence security, smart home security, and metaverse security. He has published papers in top-tier conferences and journals such as USENIX Security, IEEE INFOCOM, IEEE TIFS, IEEE TDSC, and IEEE TMC.



Yan Meng (Member, IEEE) is an Associate Professor in Shanghai Jiao Tong University, China. He received his Ph.D. degree in Computer Science and Technology from Shanghai Jiao Tong University in 2021. He received his B.S. degree in Electronic and Information Engineering from Huazhong University of Science and Technology in 2016. His research interests include wireless network security and IoT security. He published more than 60 papers, including IEEE TDSC, IEEE TMC, IEEE TIFS, ACM CCS, and USENIX Security. He received the 2022

ACM China Doctoral Dissertation Award and the Young Elite Scientists Sponsorship Program by CAST.



Yuling Chen received the B.S. degree in mathematics and applied mathematics from Taishan University, Taian, China, in 2006, and the M.S. degree in applied mathematics and the Ph.D. degree in software engineering from Guizhou University, Guiyang, China, in 2009 and 2021, respectively. She is currently a Professor with Guizhou Provincial Laboratory of Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, China. Her recent research interests include federated learning, blockchain, cryptography and information safety, and artificial intelligence security.



Jun Luo (Fellow, IEEE) received his BS and MS degrees in Electrical Engineering from Tsinghua University, China, and the Ph.D. degree in Computer Science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. From 2006 to 2008, he has worked as a postdoctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of Nanyang Technological University in Singapore, where he is currently an Associate

Professor. His research interests include mobile and pervasive computing, wireless networking, machine learning and computer vision, applied operations research, as well as security. More information can be found at <http://www.ntu.edu.sg/home/junluo>.



Haojin Zhu (Fellow, IEEE) received his B.Sc. degree (2002) from Wuhan University (China), his M.Sc.(2005) degree from Shanghai Jiao Tong University (China), both in computer science and the Ph.D. in Electrical and Computer Engineering from the University of Waterloo (Canada), in 2009. He is currently a professor and the Vice Dean of the School of Computer Science at Shanghai Jiao Tong University. His current research interests include network security and privacy enhancing technologies.

He published more than 70 international journal papers, including JSAC, TDSC, TPDS, TMC, TIFS, and 90 international conference papers, including IEEE S&P, ACM CCS, USENIX Security, NDSS, ACM MOBILECOM. He received a number of awards including SIGSOFT Distinguished Paper of ESEC/FSE (2023), ACM CCS Best Paper Runner-Ups Award (2021), IEEE TCSC Award for Excellence in Scalable Computing (Middle Career Researcher, 2020), IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award (2014), Top 100 Most Cited Chinese Papers Published in International Journals (2014), Distinguished Member of the IEEE INFOCOM Technical Program Committee (2015, 2020), best paper awards of IEEE ICC (2007) and Chinacom (2008), WASA Best Paper Runner-up Award (2017). He is serving as the editorial board for IEEE Trans. on Wireless Communications, ACM Transactions on Privacy and Security (TOPS), and program committees for top conferences such as USENIX Security, ACM CCS, NDSS, and IEEE INFOCOM.