Analysis and Design of Efficient Authentication Techniques for Password Entry with the Qwerty Keyboard for VR Environments

Tingjie Wan, Liangyuting Zhang, Yunxin Xu, Zixuan Guo, Boyu Gao and Hai-Ning Liang*, *Member, IEEE*

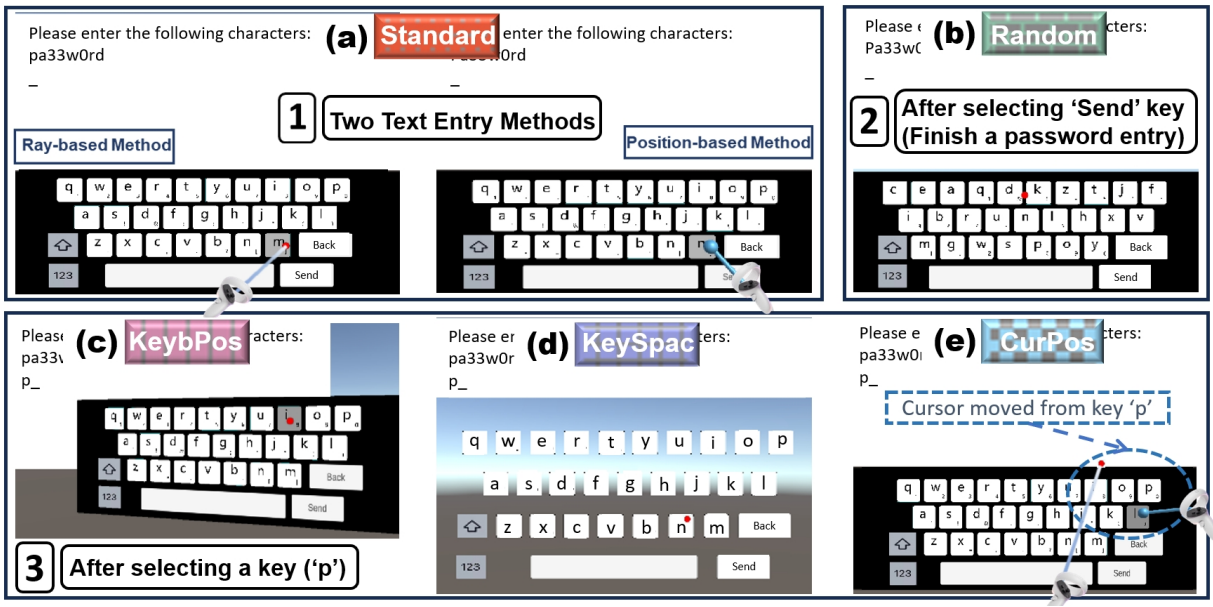

Fig. 1: (a) The Standard Qwerty keyboard (Standard, red). (b) Random layout keyboard (Random, green). The three keyboard-related position alteration strategies: (c) Altering Entire Keyboard Position (KeybPos, pink), (d) Altering Key Spacing (KeySpac, purple), and (e) Altering Cursor Position (CurPos, blue). Figure 1 shows the two text entry methods. In the ray-based method (left figure), a ray from the controller points at distant keys, with a selection made by pressing the trigger button. In the position-based method (right figure), a drumstick attached to the controller contacts keys directly for input. In Figure 2, Random features a random layout, which changes after completing the typing of a password. Figure 3 illustrates the altered state after one action of key selection in the three strategies during password entry. With KeybPos, the entire keyboard position shifts after selecting a key. KeySpac involves a one-time key spacing adjustment after one key selection. CurPos results in the cursor moving once after each key selection. To move the cursor, the direction of the ray and drum changed after one selection.

Abstract—Authentication in digital security relies heavily on text-based passwords, even with other available methods like biometrics and graphical passwords. While virtual reality (VR) keyboards are typically invisible to onlookers, the presence of inconspicuous sensors, including accelerometers, gyroscopes, and barometers, poses a potential risk of unauthorized observation and recording. Traditional defense shoulder-surfing attack methods typically involve breaking apart the Qwerty layout, which destroys the user's inherent familiarity with the layout. This research addresses the need for secure password entry in VR environments while retaining the Qwerty layout. We explore three keyboard-related position alteration strategies to ensure security while mitigating the decline in user experience. These strategies involve moving the entire keyboard, cursor, and keys. Our theoretical study assesses the effectiveness of these strategies against shoulder-surfing attacks. Two user studies, employing ray-based and position-based text entry methods, respectively, evaluate the practical effectiveness of the three strategies in resisting shoulder-surfing attacks, as well as their impact on typing performance and user experience. Our findings demonstrate that the three strategies achieve shoulder-surfing attack resistance comparable to a random layout keyboard. Moreover, compared to a random layout, the two strategies involving the movement of the entire keyboard and the repositioning of keys support faster entry rates and enhanced user experience.

Index Terms—Virtual reality; text entry; password; keyboard layout; user study

1 Introduction

*Corresponding author (email: hainingliang@hkust-gz.edu.cn)

- *Tingjie Wan, Liangyuting Zhang, Yunxin Xu, and Zixuan Guo are with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, China. Tingjie Wan and Zixuan Guo are also with the University of Liverpool, UK.*
- *BoYu Gao is with the College of Cyber Security and Guangdong Institute of Smart Education, Jinan University, China.*
- *Hai-Ning Liang is with the Computational Media and Arts Thrust, The Hong Kong University of Science and Technology (Guangzhou), China.*

- *This research was funded in part by the Suzhou Municipal Key Laboratory for Intelligent Virtual Engineering (#SZS2022004), the Natural Science Foundation of China (#62372212), and the Guangdong Province Science Foundation (#2024A1515011515).*

Received 14 March 2024; revised 17 June 2024; accepted 1 July 2024.
Date of publication 10 September 2024; date of current version 4 October 2024.
Digital Object Identifier no. 10.1109/TVCG.2024.3456195

Authentication is a fundamental aspect of digital security, and text-based passwords have long been a cornerstone of this process. While alternative authentication methods such as graphical passwords and biometric measures (e.g., fingerprints) have gained popularity, they are not without their limitations. Text passwords often serve as a fallback solution when biometric methods fail or are unavailable [9].

In the context of virtual reality (VR), where immersive experiences blur the line between the physical and digital worlds, security becomes paramount. VR introduces unique challenges to authentication. Although keyboards in VR are typically invisible to onlookers and cameras, microphones, and navigation systems like GPS are widely acknowledged as privacy-sensitive, but numerous inconspicuous sensors, including accelerometers, gyroscopes, and barometers, are less understood concerning their privacy implications and are often less protected [32, 66]. This implies that while leaking data by capturing footage in VR may be a challenge, the compromise of user interaction data from certain sensors can occur easily, such as the motion sensor on a controller. Furthermore, the prevalent use of the Qwerty layout in VR keyboards and the user's limited awareness of their physical surroundings create opportunities for unauthorized observation and analysis of password entry through means like surreptitious recording or position data collection from zero-permission sensors [22, 56, 63, 70].

Different platforms and services may have varying password requirements and policies. Therefore, retaining the Qwerty keyboard layout is necessary to accommodate different environments. Traditional defenses against shoulder-surfing attacks on the Qwerty layout typically involve randomizing key positions or secondary mapping (e.g., [35, 35, 69]). While effective in thwarting such attacks, this approach disrupts users' familiarity with the Qwerty layout, resulting in a sub-optimal user typing experience. This compromise between security and user experience has been a persistent challenge, especially in the context of VR, where typing rates are inherently slower compared to the physical world. In the previous studies (e.g., [41, 49]), non-Qwerty keyboard layouts achieved slower entry rates compared to Qwerty-based ones.

The fundamental principle of shoulder-surfing attacks lies in obtaining the user's input by capturing the movement of the input tool and aligning the keyboard layout [6]. Virtual keyboards on touchscreens, such as smartphones and smartwatches, are limited by their screen size, allowing only disorganization in the keyboard layout to counteract the alignment between layout and input actions. In expansive VR environments, there is greater flexibility, presenting an opportunity to explore alternative methods to defend the alignment of shoulder-surfing attacks without compromising the structural integrity of the Qwerty layout. This research explores the impact of modifying keyboard-related positions, including the position of the keyboard and cursor and the position of the keys, to ensure the security of text-based passwords in VR authentication while maintaining a seamless user experience. Guided by three pertinent research questions (Section 3.2), we initiated our inquiry by theoretically analyzing the effectiveness of three possible keyboard-related position modification strategies against shoulder-surfing attacks. Subsequently, two user studies were conducted to assess the efficacy of the three strategies on ray-based and position-based text entry methods initially proven effective against shoulder-surfing attacks, as well as their impact on typing performance and user experience in VR.

In short, our work presented in this paper makes the following contributions: (1) We present an analysis of the effectiveness of retaining the Qwerty key positions while modifying keyboard-related positions in countering shoulder-surfing attacks during password input in VR (Section 6). (2) We introduce and evaluate the effectiveness of three keyboard-related position strategies in mitigating shoulder-surfing attacks and reducing the reduction of entry rate with ray-based and position-based text entry methods in Study One (Section 7) and Study Two (Section 8) respectively. (3) We provide three recommendations for designing authentication measures to mitigate shoulder-surfing attacks in VR (Section 10).

2 Related Work

2.1 VR/AR Authentication

Identity authentication in VR/AR systems is based primarily on the following methods: biometric recognition, multimodal authentication, and knowledge-based authentication.

Biometric recognition techniques include behavior [42, 45], Electroencephalogram (EEG) readings [31, 34], eye-tracking [3, 21], Electrooculography (EOG) readings [40], and cranial conduction [48]. While these methods offer strong security due to their uniqueness, they are not infallible. Collecting biometric data raises privacy concerns and security vulnerabilities [9, 19]. Additionally, implementing biometrics can be challenging due to compatibility and consistency issues across platforms, often requiring extra tracking and sensing devices [2].

Multimodal biometric authentication combines multiple techniques to verify a user's identity (e.g., [45, 70]), enhancing accuracy and security by requiring attackers to bypass several systems. For instance, BlinKey [70] uses passwords and blinking rhythms. Thus, knowing the password alone is insufficient to gain illicit access. It must also be entered in a manner that matches the user's biometric gesture data. These multimodal systems help mitigate some of the shortcomings of any single authentication type but may also be susceptible to similar weaknesses.

Knowledge-based methods generally perform well and are platform-agnostic, allowing integration across different AR and VR platforms with low power consumption. These methods, like PINs and drawing patterns (e.g., [22, 35]), remain popular for VR due to their high usability. However, they are vulnerable to shoulder-surfing attacks. To mitigate this, researchers have explored randomized keys or secondary mapping (e.g., [35, 69]), but such methods can disrupt user familiarity with the Qwerty layout and potentially reduce acceptance and efficiency.

PINs and swipe patterns are commonly used across many platforms, but numerous platforms or services still require text passwords that include symbols, numbers, and letters. The traditional approach to shoulder-surfing prevention on Qwerty keyboards is to randomize the layout. Our literature review yielded two papers ([41, 49]) that have proposed methods involving randomization within rows or columns while retaining some aspects of the Qwerty structure. Their findings suggest that preserving the feature of the Qwerty layout can effectively improve entry rates. Thus, we focus on exploring retaining the Qwerty layout while enhancing resistance against shoulder-surfing attacks.

2.2 VR/AR Text Entry on Virtual Qwerty Keyboard

Natural and seamless interaction is paramount in VR. Virtual keyboards offer a promising alternative, as they can be seamlessly integrated into a VR environment and conveniently controlled using the user's hands, either through dedicated controllers or advanced hand-tracking. In VR workspaces, where user immersion is paramount, various input techniques with virtual Qwerty keyboards have been extensively evaluated. Two predominant categories have emerged: ray-based and position-based text entry methods, each with its unique merits.

Ray-based text entry, commonly known as raycasting, relies on rays or beams to interact with virtual keyboards from a distance, eliminating the need for physical contact. This method encompasses techniques such as head-pointing [7, 39, 54, 64, 68] and controller-pointing [7, 7, 13, 54, 60], offering an intuitive and efficient means of text input in VR environments, thus making it a popular choice.

Position-based text entry involves physically moving controllers or hands/fingers to specific positions corresponding to key locations on the virtual keyboard and then selecting those keys by contacting. This method simulates the physical act of pressing keys, mimicking real-world typing experiences. Researchers have explored variations of this method using controllers [7, 8, 36, 54], hands/fingers [15–18, 24], and feet [59]. Positional selection provides users with a tactile feedback-like experience, enhancing the sense of presence and bridging the gap between virtual and physical keyboards in VR environments.

Therefore, we explore the performance of two text input methods in defending against shoulder-surfing attacks by implementing two text

input techniques with a controller.

3 Preliminaries

In this section, we analyze the issue of password leakage under shoulder-surfing attacks in VR and then present our three research questions.

3.1 Shoulder-Surfing Attacks Against Password Entry in VR

Shoulder-surfing is a known method for stealing passwords, where attackers observe the user's login information. In VR, while attackers cannot see content within head-mounted displays (HMDs), they exploit indirect channels to infer possible inputs. The success of these attacks often relies on knowing the keyboard layout, typically Qwerty.

Shoulder-surfing attacks in VR can be categorized as external and internal eavesdropping. Internal eavesdropping occurs when attackers access VR sensor data without direct permissions. For example, Lu et al. [63] demonstrated that motion sensor data from VR controllers can reveal keystrokes on a Qwerty keyboard. Similarly, accelerometers and gyroscopes in devices can infer tap positions, revealing input content [38, 43, 44, 61, 65].

External eavesdropping involves exploiting channels outside the VR device to infer passwords. Attackers can observe or record the password entry process, capture controller and head movements, and use supplementary equipment to track hand and arm movements. This method allows them to reconstruct the password without direct access to the VR device. Prior research [50] has shown that video recordings of hand movements can be used to steal passwords from a touchscreen. The immersive environment of VR reduces users' awareness of their surroundings, increasing the risk of external eavesdropping like covert video recordings, making shoulder-surfing a significant threat.

3.2 Research Questions

- **RQ1** Could altering the positions related to the virtual Qwerty keyboard through three strategies effectively mitigate or prevent shoulder-surfing attacks, thereby ensuring the security of password entry?
- **RQ2** Do the strategies of keyboard-related position alteration, which demonstrated effectiveness in the analysis, apply to both text input methods?
- **RQ3** Could the keyboard modification strategies, proven effective in defending shoulder-surfing attacks, lead to improved typing performance and user experience compared to a random layout keyboard?

To help us understand the intricate interplay between security and user experience, our work represents a comprehensive research consisting of one theoretical analysis of the effectiveness of three Qwerty keyboard-related position modification strategies in defending shoulder-surfing attacks (see Figure 2) and two user studies with techniques derived from this analysis. The initial analysis serves as a foundational pillar, calculating the efficacy of three keyboard-related position alteration strategies in mitigating the risks associated with shoulder-surfing attacks (**RQ1**). These two studies explore the susceptibility to shoulder-surfing attacks, performance, and user experience implications of the three alteration strategies when inputting passwords with ray-based and position-based text entry methods, respectively (**RQ2**, **RQ3**).

4 Strategies for Keyboard-Related Position Alteration to Defend Against Shoulder-Surfing Attacks

The virtual keyboard system comprises two main components: the keyboard and a cursor. The keys on the keyboard are spaced apart. To defend against shoulder-surfing attacks, we contemplate altering three crucial positions: the position of the entire keyboard, the position of the cursor, and the position of the keys.

4.1 Altering Entire Keyboard Position (KeybPos)

The altering entire keyboard position (*KeybPos*) strategy is employed as a countermeasure against shoulder-surfing attacks, resulting in diverse displacements of the selection-making tool during interactions with keys by adjusting the position of the keyboard (Figure 1c). Following each confirmed selection of a key on the keyboard, the keyboard's position undergoes a random adjustment in the upward, downward, left, or right direction. Each change of keyboard position is relative to the original keyboard. Through the modification of the keyboard's position, this strategy introduces varied displacements in the selection-making tool, even when choosing the same key. This strategy impeccably preserves the integrity of the Qwerty layout, ensuring the user's inherent familiarity with the standard keyboard configuration.

4.2 Altering Key Spacing (KeySpac)

To maintain the Qwerty layout, the position of keys is changed by adjusting the spacing between keys, named altering key spacing (*KeySpac*) strategy (Figure 1d). That introduces variations in the displacement required for selecting a key with the selection tool, even when choosing the same key. The key spacing changes after each key selection. Changing the spacing between keys rather than altering the size of the keys themselves serves two purposes. First, it maintains the inherent familiarity users have with the key size on the standard keyboard. Second, it helps mitigate the potential for eye strain or oppression introduced by a large field of view [27]. In *KeySpac*, maintaining a fixed central position for the keyboard ensures visual stability, reducing the risk of disorientation or discomfort associated with frequent movements. This design choice supports users in developing a strong spatial awareness of key locations, facilitating quicker and more accurate key selections based on a consistent visual reference (the center of the keyboard) [14, 20].

4.3 Altering Cursor Position (CurPos)

The altering cursor position (*CurPos*) strategy corresponds to the *KeybPos* strategy, both fundamentally countering shoulder-surfing attacks through positional adjustments. Unlike *KeybPos*, *CurPos* does not change the keyboard's position but instead modifies the cursor's position corresponding to the selection tool on the keyboard (Figure 1e). For instance, if the *KeybPos* strategy shifts the keyboard to the left, the *CurPos* strategy would correspondingly shift it to the right. This also introduces varying displacements during interactions with keys. After each confirmed key selection on the keyboard, the cursor's position undergoes random adjustments in the upward, downward, left, or right directions, preventing movement of the entire keyboard. Compared to *KeybPos*, *CurPos* minimizes observable changes in the typing interface, as the keyboard remains stationary. This reduces the likelihood of users experiencing a decrease in familiarity with the fixed keyboard.

5 Ray-based and Postion-based Text Entry Methods

A controller was used for both the ray-based and position-based text entry methods. In the ray-based method (Figure 1a left), a ray emanates from the controller to point at keys on a distant keyboard. The user controls the cursor, indicating the ray's target through hand movements and wrist rotation. Keys are selected by pressing the controller's trigger button; when the ray points to a key, the key turns grey, and upon selection, it briefly turns red to confirm the input. The position-based method employs a drum-style typing technique (Figure 1a right). A drumstick is attached to the front of the controller, and the keyboard is positioned closer to the user. The key in front of the drumstick's tip turns grey, and selection occurs when the drumstick's tip physically contacts the key. In this method, the proximity of the keyboard requires direct interaction with the drumstick for input. In *CurPos*, the direction of the ray or drumstick adjusts to change the cursor position effectively.

6 Theoretical Study: Analysis of Keyboard-Related Position Alteration Impact on Shoulder-Surfing Attack Resistance (RQ1)

The objective of this study is to analyze the effectiveness of 3 keyboard-related position alteration strategies - *KeySpac*, *KeybPos* and *CurPos* in countering shoulder-surfing attacks.

6.1 Analysis Method - Naive Bayes Algorithm

We utilized the Naive Bayes machine learning method for our analysis. We randomly adjust the variation of keyboard/cursor position and key spacing within a predefined range and record all key positions on the keyboard at each variation. Specifically, we randomly altered 10,000 the keyboard (cursor) position or key spacing in each predefined range and recorded the coordinates of each key's position on each variation. Following this, we employed the Naive Bayes algorithm to create a predictive model. This model aimed to estimate the probability of predicting the key position under different variation ranges for each strategy. In this process, 5% of the key positions from the 10,000 generated layouts were designated as the test set, while the remaining 95% served as the training set. The 'naive' aspect of the classifier arises from the assumption of conditional independence between features or attributes used for classification, given the class label. Mathematically, assuming each key is independent, the probability of a particular key assignment (K) given coordinates (X, Y) can be expressed as:

$$P(X,Y|K) = P(X|K) \cdot P(Y|K)$$

$K = [k_1, k_2, .., k_n]$ is the key category set. $n = 31$, encompassing 26 letter keys, 2 mode-switching keys (to capital and to number/symbol), and 3 function keys (space, delete, send). $P(K|X,Y)$ is the joint probability distribution of rotation angles (X,Y) under the given category K. $P(X|K)$ and $P(Y|K)$ represent the margin probability distributions of horizontal rotation angle and vertical rotation angle under category K. Based on these probability distributions, we can use Bayes' theorem to calculate the posterior probability, that is, the probability that a key belongs to each letter category given the coordinates (X,Y):

$$P(K|X,Y) = \frac{P(X|K) \cdot P(Y|K) \cdot P(K)}{P(X) \cdot P(Y)}$$

6.2 Results

The *KeybPos*/*CurPos* strategies demonstrate greater resistance to shoulder-surfing compared to the *KeySpac* strategy (see Figure 2). Overall, achieving effective resistance to shoulder-surfing in *KeySpac* requires a higher range of variations compared to *KeybPos*/*CurPos*.

In the *KeybPos* and *CurPos* strategies, keyboard/cursor position adjustments involve two dimensions: horizontal and vertical. Figure 2a illustrates that while vertical position adjustments consistently yield predicted probabilities above 90%, horizontal position adjustments beyond a maximum variable position of 7 result in probabilities dropping to 18.39%. When both vertical and horizontal positions vary, probabilities fall below 16.91%, with maximum variations surpassing 2 in both directions. Probabilities drop below 9.39% when horizontal variations exceed 4 and vertical variations exceed 3. However, even with a maximum variable position of 8, probabilities remain above 5.00%.

In the *KeySpac* strategy, variations in key spacing are assessed along horizontal and vertical dimensions. As depicted in Figure 2b, adjustments to the vertical position consistently yield predicted probabilities exceeding 93%. In contrast, with a maximum horizontal position variation of 1, the probability decreases to 59.48%. Extending the range of horizontal position variation beyond this threshold does not lead to a significant reduction in probability. When both vertical and horizontal key spacing variations are introduced, predicted probabilities drop below 45.00% when the maximum position variations in the vertical and horizontal directions exceed 2. However, even with a maximum variable position of 8, the predicted probability remains above 32.00%.

7 Study One: Security and Performance Evaluation of Keyboard-Related Position Alteration Strategies with Ray-based Text Entry Method

The goal of this study is to conduct a comprehensive comparison and evaluation of three keyboard-related position alteration strategies in conjunction with the ray-based text entry method. To gauge the effectiveness and impact of these strategies on security and text entry

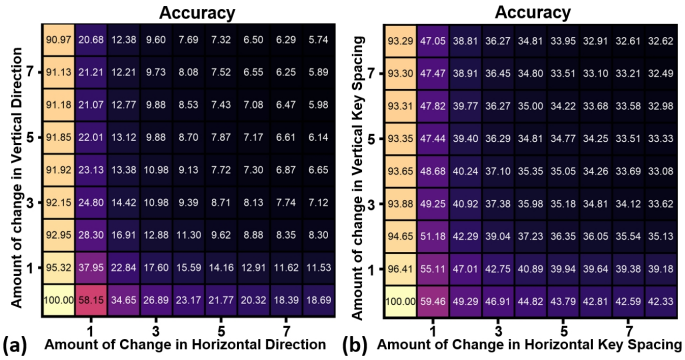

Fig. 2: (a) Effect according to the amount of change in the horizontal (x) and vertical (y) directions of keyboard/cursor position on Naive Bayes Prediction Accuracy. The key size is set to 1 unit, with row space and key space at 0. When the maximum variation value is (x,y), the variation ranges of keyboard/cursor position is ([-x, x],[-y, y]). (b) Effect according to the amount of change in the horizontal (x) and vertical (y) of the key spacing on Naive Bayes Prediction Accuracy. The key spacing is set to 0 units initially. When the maximum variation value is [x,y], the variation range of the key spacing is ([0, x],[0, y]).

performance, we use both a standard keyboard (*Standard*) and a randomly arranged keyboard (*Random*) as baseline benchmarks. The standard keyboard represents a conventional and widely used configuration, providing a reference point for text entry performance (Figure 1a). The random layout keyboard serves as a point of comparison, allowing us to evaluate the strategies in a scenario where keys lack a specific layout (Figure 1b). The two baselines help us to discern differences in security, text entry performance, and user experience across various conditions."

Based on the results of the Analysis in Section 6, we set the movement intervals for *KeybPos*/*CurPos* as ([-4, 4],[-3, 3]) and variation intervals for *KeySpac* as ([0, 2], [0,2]). These choices were made by the consideration of prediction accuracy and the occupied area size. We conducted pre-trials, following the evaluation method in Section 7.3.1, and the results showed the ICRs of all tested passwords were below 30% with these settings. The intervals of *KeybPos*/*CurPos* result in a maximum movement area of $(\text{keyboard width} + 8) \times (\text{keyboard height} + 6)$. Under the interval setting of *KeySpac*, the maximum movement area is $(\text{keyboard width} + (2 \times 9)) \times (\text{keyboard height} + (2 \times 3))$ (9 intervals in a first row and three column intervals in a keyboard). In our pre-trials, when the keyboard is altered within these intervals, it remains within the user's field of view without moving their bodies. To prevent the possibility of minor distances for each movement, we set the minimum movement distance to be greater than half of a key's width.

7.1 Participants, Materials, and Apparatus

We enrolled a total of 20 participants (10 males and 10 females) from our university campus with an age range from 19 to 26 years ($M = 22.29, SD = 2.45$). They did not receive any compensation but were provided with refreshments upon completing the experiment. While all were familiar with the Qwerty layout, as they use a laptop or computer daily, only two had no prior VR experience, five had minimal VR exposure, and thirteen were regular VR users, with ten having some typing experience in VR.

The experiments utilized a Meta Quest 2 headset connected to a Windows 10 PC with an Intel i7-7700k CPU and Nvidia GeForce GTX 1080 GPU. Unity3D (v2022.3.7f1c1) and the Oculus XR Plugin (4.0.0) packages were used for the experimental setup.

A controller facilitated ray-based text entry, with the virtual keyboard positioned strategically 10 meters in front of users. Each character key maintained a uniform size of 0.3 meters $\times$ 0.3 meters. In all three keyboard-related position alteration strategies, the keyboard moved once for each key selected by the user, and the selection was confirmed

by pressing the trigger button on the controller.

In our pursuit of genuine and contextually relevant outcomes, we utilized the RockYou password list [1]. The RockYou password list, containing over 14 million plaintext passwords from the 2009 security breach, offers a realistic sample of commonly used passwords. It includes diverse combinations of upper and lowercase letters, numbers, and symbols, devoid of inherent linguistic meaning.

7.2 Experiment Design and Procedure

This study employed a within-subjects design with the independent variable being the STRATEGY. The order of the five conditions was counterbalanced using a Latin-Square approach. In each condition, participants transcribed 12 passwords sourced from the RockYou password list. Passwords were selected randomly without any duplicates. The selected password is between 6 and 8 digits long and contains 2 types of characters (letter, symbol, or number). This is the kind of password that is commonly found in the RockYou password list. The initial two sentences were exclusively meant for training purposes, and the participants' performance on these sentences was not recorded. The subsequent ten sentences constituted the formal trials, and their outcomes were recorded for subsequent analysis. In total, 1000 trials were included in the analysis (= 20 participants $\times$ 5 strategies $\times$ 10 sentences).

Before initiating the sessions, participants completed a consent form and a demographics questionnaire. Following this, they underwent a comprehensive orientation covering the VR device, task objectives, and controls. Participants were explicitly instructed to prioritize both speed and accuracy during typing. Post-task questionnaires were administered after each condition, and upon completion of all conditions, participants engaged in semi-structured interviews to provide feedback and suggestions. To facilitate participant comfort, a five-minute break separated each session, with additional time accommodated upon request. The entire experiment spanned about 50 minutes.

7.3 Evaluation Metrics

7.3.1 Performance of Protecting Input Content

We evaluated the security effectiveness of the five strategies by simulating a shoulder-surfing attack using the objective data recorded during the experiments. Motion data from the controller was logged every time a key on the keyboard was selected, capturing pitch and roll angles, position, and the distance along the Z-axis between the keyboard and the controller. Since participants consistently concluded by pressing the "Send" key, we utilized the reconstructed 2D keyboard with a tree-based backward inference algorithm to predict the entered password. This prediction method was replicated from Wu et al.'s research [63]. For estimating the 3D cursor location and reconstructing the 2D keyboard based on the recorded data, we applied the 3D cursor estimation and K-means clustering algorithms, following the methodology used in Wu et al.'s study [63].

- **Identical Character Ratio** (ICR) calculates the proportion of characters at corresponding positions between two strings that are identical to the total number of characters. This measurement allows a direct comparison of the absolute consistency between an original string and a transcribed string. It is particularly well-suited for passwords due to their stringent character order and consistency requirement.
- **Ranking and Interview** We require participants to rank all the five conditions they are willing to use when entering the password, consider security, and provide the reasons for the ranking.

7.3.2 Performance of Text Entry

We measured the text entry performance of the five strategies under the raycasting technique using the objective data recorded during the experiments. Additionally, we gathered the subjective data collected from two questionnaires.

- **Entry Rate** was quantified in words per minute (WPM) [67]. This measure was derived by dividing the number of transcribed words by the time taken to complete the text transcription, measured in minutes. A word was defined as five keystrokes long, encompassing spaces.
- **Error Rate** [53] was evaluated employing char-level typing metrics, where the total error rate (TER) constitutes the combination of the not corrected error rate (NCER) and the corrected error rate (CER).
- **Usability** of each strategy was measured using a System Usability Scale (SUS) questionnaire [12]. This questionnaire consists of 10 questions, each rated on a 5-point scale. The analysis involved using the weighted overall score, ranging from 0 to 100, where higher scores indicate enhanced usability.
- **Workload** associated with the five strategies was evaluated using the RAW NASA-TLX workload questionnaire [25]. This questionnaire comprises six subscales that gauge various facets of workload, encompassing mental demand, physical demand, temporal demand, frustration, effort, and performance. Participants provided ratings for each subscale on a scale from 0 to 100, with 5-point intervals. Reduced scores on this scale signify decreased workload and enhanced overall performance.

7.4 Results

We used SPSS 26 for data analysis. The results of Shapiro-Wilk tests and Q-Q plots revealed that entry rate and ICR data exhibited normal distribution ($p > .05$), whereas TER, NCER, SUS, and NASA-TLX data did not follow a normal distribution ($p < .05$). Consequently, an Aligned Rank Transform [62] was applied to non-normally distributed data before conducting Repeated Measure (RM-) ANOVA tests. Given the six dimensions in NASA-TLX data, Multivariate Analysis of Variance (MANOVA) was utilized for comparison. For one-dimensional data, RM-ANOVA tests were employed. Effect sizes were reported with partial eta squared (η_p^2).

7.4.1 Identical Character Ratio

An RM-ANOVA analysis revealed a significant main effect of STRATEGY on the ICR ($F_{4,76} = 222.966, p < .001, \eta_p^2 = .018$), as illustrated in Figure 3a. The ICR of *Standard* ($M = 60.26, SD = 11.69$) was significantly higher than *KeybPos* ($M = 16.23, SD = 1.90$), *KeySpac* ($M = 18.31, SD = 3.16$), *CurPos* ($M = 16.26, SD = 1.76$) and *Random* ($M = 19.26, SD = 3.34$) (all $p < .001$).

7.4.2 User Ranking and Feedback

As shown in Figure 3b, out of the 20 participants, 10 ranked *KeySpac* as the best strategy, while the remaining 8 participants chose *KeybPos* and 2 participants selected *Standard*. Some of them thought *KeySpac* seems a little more attractive than *KeybPos*. 18 participants ranked *Standard* at the bottom because, although it was easy to use, they felt it could not ensure their security. The other 2 did not believe that information is easily disclosed in VR. 11 individuals expressed a preference for using *Random* over *CurPos* since they consistently spent a significant amount of time searching for the cursor position and the correspondence between the controller and the cursor feels somewhat awkward.

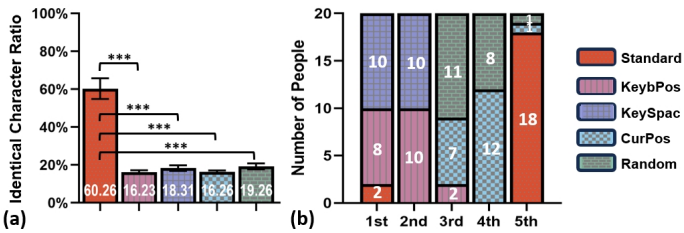

Fig. 3: (a) The means of ICR and (b) The Ranking of each strategy when typing passwords. Error bars represent 95% confidence intervals. ***, **, and * represent a .001, .01, and .05 significance level, respectively.

7.4.3 Entry Rate and Error Rate

An RM-ANOVA analysis indicated that STRATEGY had significant a main effect on entry rate ($F_{4,76} = 240.017, p < .001, \eta_p^2 = .094$), as illustrated in Figure 4a. Post-hoc pairwise comparisons revealed that the entry rate of *Standard* ($M = 16.45, SD = 1.87$) was higher than it of *KeySpac* ($M = 12.17, SD = 1.78$), *KeybPos* ($M = 11.57, SD = 1.52$), *CurPos* ($M = 7.47, SD = 1.29$) and *Random* ($M = 7.70, SD = 1.13$) (all $p < .001$). both *KeybPos* and *KeySpac* outperform *CurPos* and *Random* (all $p < .001$).

Figure 4b and c summarize the TER and NCER results, respectively. RM-ANOVAs analysis did not reveal any significant differences in STRATEGY for TER and NCER ($p > .05$).

7.4.4 Usability and Perceived Workload

An RM-ANOVA found a significant main effect of STRATEGY in SUS scores ($F_{4,76} = 37.534, p < .001, \eta_p^2 = .364$) as Figure 4d shown. Post-hoc pairwise comparisons demonstrated that *Standard* ($M = 94.90, SD = 8.93$) exhibited higher usability compared to *KeySpac* ($M = 82.45, SD = 14.43$), *KeybPos* ($M = 80.00, SD = 14.01$), *CurPos* ($M = 64.50, SD = 21.90$), and *Random* ($M = 62.80, SD = 17.50$) (all $p < .001$). The usability of *Random* was significantly lower than that of both *KeybPos* and *KeySpac* ($p = .001$ for both). *CurPos* demonstrated lower usability compared to *KeybPos* ($p = .018$) and *KeySpac* ($p = .003$).

Figure 4e shows the NASA-TLX scores for the five strategies. MANOVAs revealed a significant difference in perceived workload ($F = 16.496, p < .001, Wilks'\Lambda = .124, \eta_p^2 = .876$). For each dimension of NASA-TLX, RM-ANOVAs showed significant effects in mental demand ($F_{4,76} = 16.810, p < .001, \eta_p^2 = .469$), physical demand ($F_{4,76} = 22.156, p < .001, \eta_p^2 = .538$), temporal demand ($F_{4,76} = 18.072, p < .001, \eta_p^2 = .487$), effort ($F_{4,76} = 22.017, p < .001, \eta_p^2 = .537$), performance ($F_{4,76} = 10.448, p < .001, \eta_p^2 = .355$), and frustration ($F_{4,76} = 21.543, p < .001, \eta_p^2 = .531$). Post-hoc tests indicated that *Standard* achieved the lowest workload, followed by *KeybPos* and *KeySpac*, with *Random* and *CurPos* having the highest workload across all six dimensions.

7.5 Discussion

7.5.1 Effectiveness of Ensuring the Security of Password Entry (**RQ2**)

Our results demonstrate the effectiveness of the three strategies in resisting shoulder-surfing attacks within the ray-based text entry method, as the ICRs for all three strategies (*KeybPos*: 16.23%, *CurPos*: 16.26% and *KeySpac*: 18.31%) are significantly lower than that of *Standard* (60.26%), achieving a level of resistance comparable to *Random* (19.26%) (see Figure 3a). When entering passwords, users are often hesitant to use *Standard* due to their heightened self-security awareness. However, it is evident that when weighing usability/convenience against security, users typically prioritize usability/convenience [4, 55]. The majority of their ranking considerations were related to entry rate (see Figure 3b and Figure 4a).

In our three proposed strategies, the random nature of the changes ensures that even if attackers understand the concept used, they cannot effectively counteract this randomness. Since a single motion data point from the controller corresponds to multiple possible keys; as such, the composition of a password, typically containing multiple characters, leads to an exponential increase in potential combinations. Additionally, standard authentication mechanisms do not permit multiple attempts. This combination of randomness and limited attempts makes our concept effective in defending against shoulder-surfing attacks.

KeySpac was slightly preferred over *KeybPos* for password entry, mainly because *KeySpac* maintained the central point of the keyboard, aiding users' relative positional memory recall [37]. Users perceived *KeySpac* changes as more dynamic, potentially enhancing engagement according to prior studies on dynamism's influence on interest [47]. However, in the *KeybPos* strategy, users initially had to locate the keyboard across their field of view. Nonetheless, its advantage lies in the consistent internal distances between keys, enabling quicker target key identification once users locate the keyboard.

The *CurPos* strategy, which maintains the absolute keyboard position while preserving key-to-key distances, faced limited user preference during password input due to several factors. The small cursor size hindered quick location, impacting user experience negatively. Additionally, perceptual-motor discrepancies arose from cursor movement, causing inconsistency between its virtual position and the controller's actual direction. This inconsistency, where the cursor moved opposite to the controller's direction after selection, led to user discomfort and confusion, disrupting the expected correspondence between virtual and physical representations [10, 28]. Additionally, the movement of the cursor causes perceptual-motor discrepancies in distance, which also has detrimental effects on user interactions [11]. In the *CurPos* strategy, the cursor's position shifts after each key selection, creating a perceptual-motor discrepancy when entering the next character. The perceived movement distance is visually based on the previous cursor's position, but since the cursor moves, this results in a mismatch between the perceived and the actual required movement distance.

7.5.2 The Typing Performance and User Experience (**RQ3**)

The *KeySpac* and *KeybPos* strategies demonstrated entry rates of 12.17 WPM and 11.57 WPM, respectively, which were slower than the *Standard* configuration (16.45 WPM) but faster than *CurPos* (7.47 WPM) and *Random* (7.70 WPM), as shown in Figure 4a. These outcomes align with prior research, where non-Qwerty layouts had significantly lower entry rates [41, 49]. In our observation, participants gradually adapted to the keyboard variations, becoming more proficient, but the entry rate did not significantly increase. This suggests a training effect that leads to a reduced cognitive load over time, while the improvement in entry rates is primarily attributed to users' pre-existing familiarity with the Qwerty layout. This implies that modifying the keyboard properties with the *KeySpac* and *KeybPos* strategies also leads to a decrease in the entry rate, although the impact is less pronounced than that of the *Random* strategy.

It is important to recognize the nuanced impact of complex text on entry rates [46], with examples such as achieving over 10 WPM with raycasting on a VR virtual keyboard [58], 6.5 WPM and 7.1 WPM with hand on AR virtual keyboards [52], and 5.37 WPM and 8.16 WPM with dwell-based text entry [57]. The lower entry rate observed for password input in our results can be attributed to the introduction of various character types and necessary adjustments to the Qwerty keyboard layout.

Standard, *KeybPos*, and *KeySpac* all exhibit satisfactory usability with the mean scores surpassing 70 (70 means acceptable level [5]), and *Standard* attains the highest SUS score. Participants consistently reported workloads within the 30-point range across the three strategies, indicating a moderate level of cognitive engagement during typing tasks. In contrast, the workloads for *CurPos* and *Random* are nearly 40 or higher than 40 in six dimensions—any workload exceeding 40 is considered high [29] (see Figure 4e)—and their SUS scores are borderline with a mean score of around 60 [5].

7.5.3 Result Discrepancies Between Theory Analysis and Real Shoulder-Surfing Attacks

The disparity between the results obtained from Naive Bayes predictions (theoretical analysis) and tree-based backward inference (user study) in *KeybPos*/*CurPos* and *KeySpac* can be attributed to their inherent methodological disparities. Naive Bayes operates within an idealized theoretical framework, providing the probability distribution for all possible key positions based on prior knowledge. On the other hand, the tree-based model relies on practical experimentation, inferring the keyboard layout and pressed keys from the actual position data obtained from the controller in real-world conditions.

In the case of *KeySpac*, the higher results from Naive Bayes are influenced by the fact that the keyboard's central position remains unchanged, creating a more pronounced and consistent feature for the model. However, in real shoulder-surfing attack scenarios, this feature's value diminishes due to insufficient data to fully capture its feature.

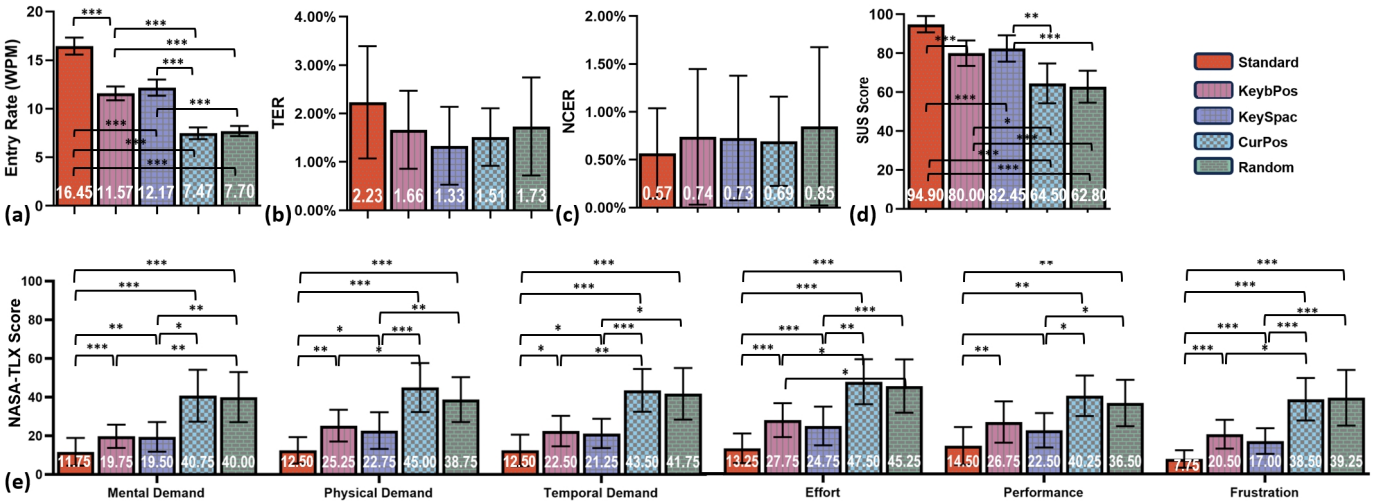

Fig. 4: The means of (a) Entry Rate: (b) TER: (c) NCER: (d) SUS Scores (the higher, the better): and (e) NASA-TLX Scores (the lower, the better). Error bars represent 95% confidence intervals.

Despite the disparity with real-world results, this theoretical analysis assists in ensuring the strategies are adapted for effectiveness in practical scenarios. Both the results of our theoretical analysis and this user study indicate that to achieve the desired resistance against shoulder-surfing, *KeybPos*/*CurPos* requires a smaller movement range compared to *KeySpac*.

7.6 Lessons from Study One

The following lessons (**L#**) were learned from this study.

- **L1.** *KeybPos*, *KeySpac* and *CurPos* strategies are effective in resisting shoulder-surfing attacks with ray-based text entry method (**RQ2**).
- **L2.** *KeybPos* and *KeySpac* consistently outperforms the *Random* on text entry performance with higher usability and lower workload (**RQ3**).
- **L3.** Consistency between perceived (visual) and actual movement, both in direction and distance, is crucial for a positive user experience. Deviations in these aspects, as observed in the *CurPos* strategy, can lead to confusion and reduced typing performance.
- **L4.** Complex text, encompassing uppercase letters, symbols, and numbers, and adjustments to the Qwerty layout both introduced a nuanced impact on the entry rate.

8 Study Two: Security and Performance Evaluation of Keyboard-Related Position Alteration Strategies with the Position-based Text Entry Method

The goal of this study is to compare and evaluate the security, text entry performance, and user experience of the three keyboard-related position alteration strategies with the position-based text entry method. *Standard* and *Random* strategies still used as the baselines in this study. The change interval settings of *KeybPos*/*CurPos* and *KeySpac* are consistent with the settings in Study One.

8.1 Participants and Materials

20 participants, comprising 10 males and 10 females, with ages ranging from 19 to 27 ($M = 22.34, SD = 3.45$), were recruited from our university campus also to take part in the study with the same compensation. Like participants in the first study, they were familiar with the Qwerty keyboard. Three participants had no VR experience before, while two had used VR a few times but had not engaged in typing activities. Fifteen participants used VR several times a month, with 12 of them indicating some experience with typing systems within VR.

A controller was employed to realize the position-based text entry method. In the three keyboard-related position alteration strategies, the keyboard moves once for every key the user selects. The confirmation of selection is by strikes with the controller. The virtual keyboard was strategically positioned 40 centimeters ahead of the users, sized at 28 centimeters in width and 13 centimeters in height. Each character key remained uniform at 2.4 centimeters $\times$ 2.4 centimeters. The visual perception of the keyboard size in VR remains consistent with the keyboard in Study One.

8.2 Experiment Design and Procedure

This study utilized a within-subjects design, with STRATEGY serving as the independent variable. The experimental design and procedures were identical to Study One in Section 7.2, with the only difference being that participants were required to use the location-based text entry method.

8.3 Results

Shapiro-Wilk tests and Q-Q plots showed that TER, NCER, SUS, and NASA-TLX data were not normally distributed ($p < .05$). Therefore, we applied the Aligned Rank Transform [62] to these data before conducting RM-ANOVA tests. For the six dimensions of NASA-TLX data, MANOVA was used for comparison. For normally distributed one-dimensional data, RM-ANOVA tests were applied.

8.3.1 Identical Character Ratio

An RM-ANOVA analysis detected a significant main effect of STRATEGY on the ICR ($F_{4,76} = 142.788, p < .001, \eta_p^2 = .088$), as depicted in Figure 5a. The ICR of *Standard* ($M = 62.14, SD = 13.46$) surpassed significantly that of *KeybPos* ($M = 16.81, SD = 5.74$), *KeySpac* ($M = 18.77, SD = 6.05$), *CurPos* ($M = 16.78, SD = 5.56$), and *Random* ($M = 18.38, SD = 5.17$) (all $p < .001$).

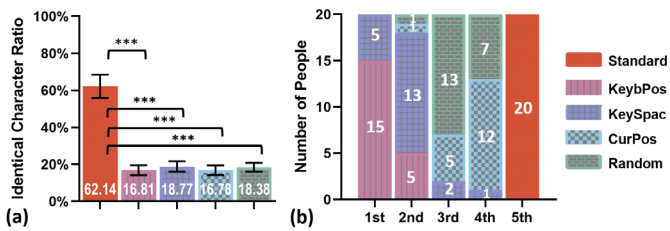

Fig. 5: (a) The means of ICR and (b) The Ranking of each strategy when typing passwords. Error bars represent 95% confidence intervals.

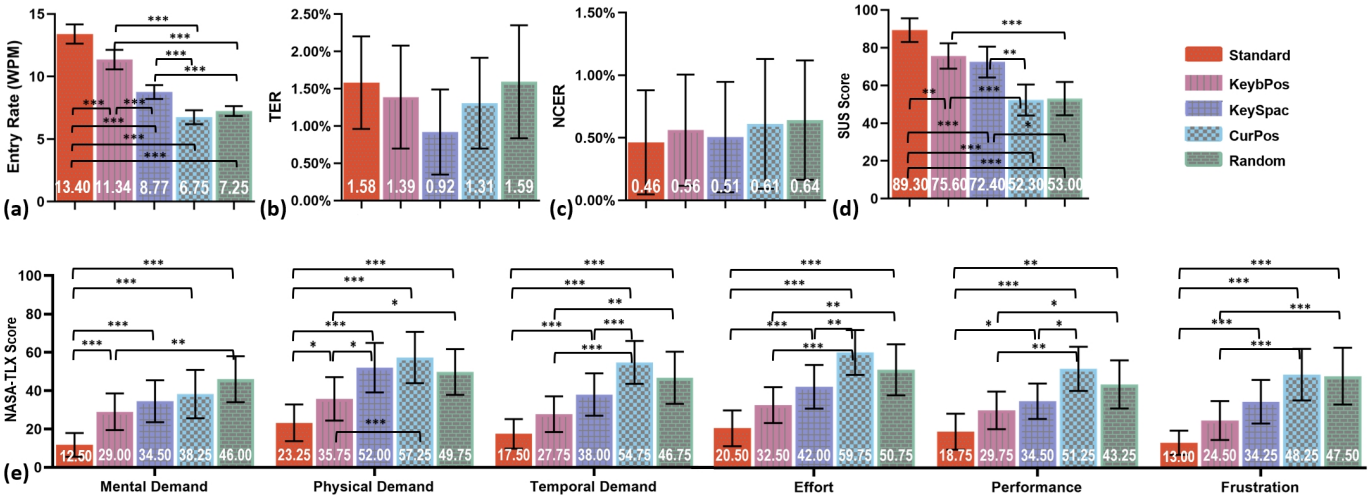

Fig. 6: The means of (a) Entry Rate: (b) TER: (c) NCER: (d) SUS Scores (the higher, the better): and (e) NASA-TLX Scores (the lower, the better). Error bars represent 95% confidence intervals.

8.3.2 User Ranking and Feedback

As depicted in Figure 5b, 15 out of the 20 participants favored *KeybPos* as the optimal strategy, while the remaining 5 participants opted for *KeySpac*. Although *KeySpac* facilitated easy identification of the target key and the change is attractive, the larger spacing between keys necessitated considerable arm movement to reach distant keys. Notably, *Standard* ranked lowest for 20 participants, as they believed that observing hand and arm movements might enable guessing the typed content. Additionally, 13 participants preferred *Random* over *CurPos* due to the perceived challenge of locating the cursor position and controlling the cursor during the typing tasks.

8.3.3 Entry Rate and Error Rate

An RM-ANOVA analysis indicated that STRATEGY showed a significant difference on entry rate ($F_{4,76} = 100.585, p < .001, \eta_p^2 = .170$), as illustrated in Figure 6a. Post-hoc tests showed *Standard* ($M = 13.40, SD = 1.64$) achieved the fastest text entry, followed by *KeybPos* ($M = 11.34, SD = 1.67$), then *KeySpac* ($M = 8.77, SD = 1.18$), with *Random* ($M = 7.25, SD = 0.83$) and *CurPos* ($M = 6.75, SD = 1.19$) being the slowest ($p < .001$ in the all pairs). Figure 6b and c present the results for TER and NCER. RM-ANOVA analyses did not indicate STRATEGY had any significant effects on TER and NCER ($p > .05$).

8.3.4 Usability and Perceived Workload

A significant main effect of STRATEGY on SUS scores was identified through RM-ANOVA ($F_{4,76} = 32.029, p < .001, \eta_p^2 = .628$), see Figure 6d. Post-hoc pairwise comparisons revealed that *Standard* ($M = 89.30, SD = 13.40$) outperformed *KeybPos* ($M = 75.60, SD = 14.37, p = .005$), *KeySpac* ($M = 72.40, SD = 17.39, p = .001$), *Random* ($M = 53.00, SD = 18.95, p < .001$), and *CurPos* ($M = 52.30, SD = 17.57, p < .001$) in usability. *KeybPos* exhibited superior usability compared to *CurPos* and *Random* (both $p = .001$), while *KeySpac* showed higher usability than *CurPos* ($p = .006$) and *Random* ($p = .011$).

Figure 6e shows the NASA-TLX scores for the five strategies. MANOVAs revealed a significant difference in perceived workload ($F = 16.106, p < .001, Wilks'\Lambda = .127, \eta_p^2 = .873$). For each dimension of NASA-TLX, RM-ANOVAs showed significant effects in mental demand ($F_{4,76} = 19.715, p < .001, \eta_p^2 = .509$), physical demand ($F_{4,76} = 20.035, p < .001, \eta_p^2 = .513$), temporal demand ($F_{4,76} = 22.084, p < .001, \eta_p^2 = .538$), effort ($F_{4,76} = 23.421, p < .001, \eta_p^2 = .552$), performance ($F_{4,76} = 13.978, p < .001, \eta_p^2 = .424$), and frustration ($F_{4,76} = 22.861, p < .001, \eta_p^2 = .546$). Post-hoc tests indicated that *Standard* achieved the lowest workload, and *Random* and *CurPos* had the higher workload across all six dimensions.

8.4 Discussion

8.4.1 Effectiveness of Ensuring the Security of Password Entry (**RQ2**)

The Identical Character Ratios (ICR defined in Section 7.3.1) of *Standard* with ray-based and position-based typing methods were over 60%, further demonstrating that it is feasible to infer user input in VR environments by analyzing sensor data. This aligns with previous research findings that accelerometers, gyroscopes, and motion sensors can reveal the position of taps, thereby compromising user input data [30,38,43,44,61,63,65]. When a position-based text entry method is employed, *KeybPos* (ICR: 16.81%), *KeySpac* (ICR: 18.77%), and *CurPos* (ICR: 16.78%) all demonstrated comparable capabilities to the *Random* strategy (ICR: 18.38%) in resisting shoulder-surfing attacks.

Prior solutions to mitigate indirect shoulder-surfing attacks in VR typically involve using randomized layouts [41], unobservable or unrecordable input methods like gaze-based input [33], or alternative authentication methods summarised by Bošnjak and Brumen [9]. Randomized keyboard layouts are effective in mitigating shoulder-surfing attacks but often sacrifice efficiency. Gaze-based input methods provide robust defenses against external eavesdropping, but their efficacy against internal eavesdropping is uncertain. Other alternative authentication methods, while valuable, generally lack the widespread applicability of password-based systems. Our proposed password protection strategies maintain the familiar Qwerty layout, addressing security concerns and mitigating the damage to the user experience. This makes our approach more advantageous, as it allows users to continue using their common input method while enhancing security against both external and internal threats.

In both *KeybPos* and *KeySpac*, users prefer *KeybPos* due to its synchronization of keyboard movement with upper body shifts observed in our study, leveraging natural coordination [51]. This minimizes the need for extensive arm movements compared to *KeySpac*, where users maintain body orientation toward the keyboard center and rely more on arm movements. Additionally, *KeySpac* entails larger spatial movements, leading to increased physical effort and fatigue. In position-based text input, user preference for *CurPos* is notably low due to factors such as small cursor size and perceptual-motor discrepancies in position and distance, as discussed in Section 7.5.1. Increased hand and arm spatial movement also contributes to user reluctance.

8.4.2 Typing Performance and User Experience (**RQ3**)

The results of our study demonstrate significant differences among the strategies in text entry performance, usability, and perceived workload. The *KeybPos* strategy (11.34 WPM), which involves altering positions of the keyboard, exhibited superior text entry rates compared

to *KeySpac* (8.77 WPM), *CurPos* (6.75 WPM), and *Random* (7.25 WPM), albeit slightly slower than the *Standard* (13.40 WPM). The higher usability score of *KeybPos* than it of *KeySpac*, *CurPos*, and *Random*, showing *KeybPos* is better on typing experience (Figure 6a).

Despite *KeySpac*'s lower text entry rate (8.77 WPM) and slightly higher perceived physical workload (52.00), it still received acceptable usability scores (over 70 in Figure 6d [5]). This suggests that users prefer *KeySpac* due to its spatial cues from the fixed keyboard center and engaging dynamics. The fixed center aids spatial awareness, and the dynamic nature adds enjoyment (see Section 7.5.1). In contrast, *CurPos* and *Random* strategies consistently resulted in slower text entry rates and higher perceived workloads, with *CurPos* being the slowest. NASA-TLX results showed that *Standard* had the lowest workload, while *CurPos* and *Random* had higher workloads across all dimensions, supporting the benefit of maintaining a standard keyboard layout [26].

8.4.3 Ray-based Method VS. Position-based Method

In ray-based and position-based text input methods, differences in users' rankings and entry rates for *KeybPos* and *KeySpac* are due to the distinct arm movements required. Ray-based input involves small movements, coordinating the wrist and forearm, while position-based input requires larger movements involving the upper arm, forearm, and wrist. Since distal movements (hand) are less fatiguing than proximal movements (upper arm) [23], strategies increasing movement space make position-based input more likely to cause physical fatigue.

8.5 Lessons Learned from Study Two

The following lessons (**L#**) were learned from this study.

- **L5.** *KeybPos*, *KeySpac* and *CurPos* strategies are effective in resisting shoulder-surfing attacks with ray-based text entry method (**RQ2**).
- **L6.** *KeybPos* consistently surpasses *Random* in text entry performance, exhibiting higher usability and reduced workload. (**RQ3**).
- **L7.** The differences in spatial requirement are the primary reason for the divergence in performance between the ray-based and position-based text entry methods when employing the *KeybPos* and *KeySpac* strategies.
- **L8.** A moving keyboard center encourages users to instinctively synchronize upper body movements with its motion, thereby reducing the space needed for hand and arm movements. While a fixed keyboard center provides a stable reference point, enhancing users' spatial awareness.

9 Potential Applications with Other Text Entry Approaches

The outcomes of our study underscore the acceptable performance exhibited by both the *KeySpac* and *KeybPos* strategies for text entry and resistance against shoulder-surfing attacks. Notably, these strategies, which exclusively modify keyboard-related positions, offer a versatile application potential across various virtual text entry techniques that use a virtual Qwerty keyboard. For instance, the principles in *KeySpac* and *KeybPos* could be introduced seamlessly in head-pointing typing, a representative form of ray-based typing (see Figure 7a). Moreover, the *KeybPos* strategy could be adapted to hand gesture typing, a characteristic position-based typing method, as the movement space during hand gesture typing is similar to that of using a controller-based position-based text entry method (an example is in Figure 7b). This adaptability positions *KeySpac* and *KeybPos* as compelling and user-friendly solutions for many VR text input scenarios.

10 Design Recommendations for Preventing Shoulder-surfing Attacks in VR

From the above results, we can distil the three design considerations.

- **Preserving the Qwerty Layout** Retaining the Qwerty keyboard layout when implementing measures to resist shoulder-surfing attacks is recommended. This preserves users' familiarity with the layout, mitigating potential decreases in typing performance caused by enhanced security measures.

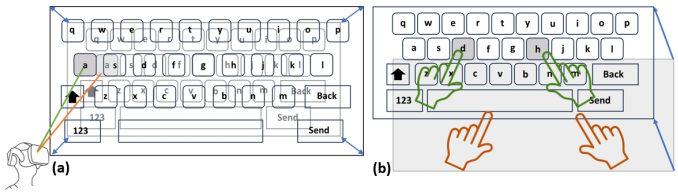

Fig. 7: (a) Head-based pointing using *KeySpac*. The orange ray indicates the 'a' key on the original keyboard, while the green ray points to the 'a' on the modified keyboard. (b) Hand gesture typing with *KeybPos*. The orange hand represents the actual position of the user's hand before the keyboard movement. After the keyboard is repositioned, the user needs to adjust their hand to the new keyboard location.

- **Adding Cues for Key Positions** Introducing cues for key positions, such as the fixed keyboard center in *KeySpac*, can assist users in quickly locating characters to mitigate potential decreases in typing performance resulting from security measures.
- **Maintaining Consistency in Perceptual-Motor Position and Distance** When designing measures to resist shoulder-surfing attacks, it is crucial to avoid introducing perceptual disparities, including position and distance. Inconsistencies in perceptual-motor position and distance can lead to user confusion, disrupting the overall user experience, as mentioned in **L3**.

11 Limitations and Future Work

Our study has two limitations that can serve as focal points for future research. First, the participant pool consisted primarily of young adults. This somewhat limited demographic diversity may affect the generalizability of the findings to broader populations. Although younger participants are the largest user groups and may be more familiar with the Qwerty keyboard, potentially providing an advantageous context for validating the effectiveness of various strategies in resisting shoulder-surfing attacks and enhancing typing performance, future studies should aim to include a more diverse participant pool. This diversity can help assess the strategies' effectiveness across a wider range of user types and demographics. Second, our study primarily focused on simulating controller data leakage as a form of shoulder-surfing attack, given its confirmed accuracy [63]. However, shoulder-surfing attack methods also include other means of unauthorized observation. Our future work will explore the performance of the three strategies under these alternative shoulder-surfing attack scenarios. Evaluating the strategies in the presence of more sophisticated shoulder-surfing methods will provide us with a further understanding of their effectiveness and further enhance the security measures for virtual text entry in VR environments.

12 Conclusion

Our work delved into the strategy to resist shoulder-surfing attacks in VR environments when entering passwords with ray-based and position-based text entry methods. Through the introduction and evaluation of three keyboard-setting alteration strategies, our goal was to fortify security and mitigate the damage to the inherent familiarity of the Qwerty layout. The findings underscore the efficacy of the *KeySpac* and *KeybPos* strategies in the ray-based text entry and the *KeybPos* strategy in the position-based text entry. Notably, *KeySpac* and *KeybPos* consistently outperformed *Random* on typing performance and are comparable to *Random* on security.

Despite the challenges associated with password input and intricate text, both *KeySpac* and *KeybPos* maintained competitive entry rates, positioning them as promising solutions for VR authentication. Our study makes a valuable contribution to the ongoing discourse on fortifying VR text input systems. We advocate for the widespread adoption of the *KeySpac* and *KeybPos* strategies, as their universality and applicability to various VR text input techniques. These strategies offer an advantageous balance between security and user experience, making them cost-effective solutions that require no additional effort to ensure robust security in VR environments.

REFERENCES

[1] Common password list (rockyou.txt). https://www.kaggle.com/datasets/wjburns/common-password-list-rockyoutxt. [Accessed 06-03-2024]. 5

[2] S. Ahmad, B. Mohd Ali, and w. A. wan adnan. Technical issues and challenges of biometric applications as access control tools of information security. *International Journal of Innovative Computing, Information and Control*, 8:7983–7999, 11 2012. 2

[3] K. Ahuja, R. Islam, V. Parashar, K. Dey, C. Harrison, and M. Goel. Eyespyvr: Interactive eye sensing using off-the-shelf, smartphone-based vr headsets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(2), jul 2018. doi: 10.1145/3214260 2

[4] M. Awad, Z. Al-Qudah, S. Idwan, and A. H. Jallad. Password security: Password behavior analysis at a small university. In *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, pp. 1–4, 2016. doi: 10.1109/ICEDSA.2016.7818558 6

[5] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, 2008. doi: 10.1080/10447310802205776 6, 9

[6] F. Binbeshr, M. Mat Kiah, L. Y. Por, and A. Zaidan. A systematic review of pin-entry methods resistant to shoulder-surfing attacks. *Computers & Security*, 101:102116, 2021. doi: 10.1016/j.cose.2020.102116 2

[7] C. Boletsis and S. Kongsvik. Controller-based text-input techniques for virtual reality: An empirical comparison. *International Journal of Virtual Reality*, 19(3):2–15, Oct. 2019. doi: 10.20870/IJVR.2019.19.3.2917 2

[8] C. Boletsis and S. Kongsvik. Text input in virtual reality: A preliminary evaluation of the drum-like vr keyboard. *Technologies*, 7(2), 2019. doi: 10.3390/technologies7020031 2

[9] L. Bošnjak and B. Brumen. Shoulder surfing experiments: A systematic literature review. *Computers & Security*, 99:102023, 2020. doi: 10.1016/j.cose.2020.102023 2, 8

[10] D. Brickler, M. Volonte, J. W. Bertrand, A. T. Duchowski, and S. V. Babu. Effects of stereoscopic viewing and haptic feedback, sensory-motor congruence and calibration on near-field fine motor perception-action coordination in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 28–37, 2019. doi: 10.1109/VR.2019.8797744 6

[11] D. Brickler, M. Volonte, J. W. Bertrand, A. T. Duchowski, and S. V. Babu. Effects of stereoscopic viewing and haptic feedback, sensory-motor congruence and calibration on near-field fine motor perception-action coordination in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 28–37, 2019. doi: 10.1109/VR.2019.8797744 6

[12] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996. 5

[13] S. Chen, J. Wang, S. Guerra, N. Mittal, and S. Prakkamakul. Exploring word-gesture text entry techniques in virtual reality. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, p. 1–6. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290607.3312762 2

[14] I. A. Clemens, L. P. Selen, M. Koppen, and W. P. Medendorp. Visual stability across combined eye and body motion. *Journal of vision*, 12(12):8–8, 2012. 3

[15] J. L. Derby, C. T. Rarick, and B. S. Chaparro. Text input performance with a mixed reality head-mounted display (hmd). *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 63(1):1476–1480, 2019. doi: 10.1177/1071181319631279 2

[16] T. J. Dube and A. S. Arif. Impact of key shape and dimension on text entry in virtual reality. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, p. 1–10. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3334480.3382882 2

[17] J. Dudley, H. Benko, D. Wigdor, and P. O. Kristensson. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 289–300, 2019. doi: 10.1109/ISMAR.2019.00027 2

[18] J. J. Dudley, K. Vertanen, and P. O. Kristensson. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Trans. Comput.-Hum. Interact.*, 25(6), dec 2018. doi: 10.1145/3232163 2

[19] R. L. Finn, D. Wright, and M. Friedewald. *Seven Types of Privacy*, pp. 3–32. Springer Netherlands, Dordrecht, 2013. doi: 10.1007/978-94-007-5170-5_1 2

[20] B. A. Galati Gaspare, Pelle Gina and C. Giorgia. Multiple reference frames used by the human brain for spatial perception and memory. *Experimental Brain Research*, 206:109–120, 2010. doi: 10.1007/s00221-010-2168-8 3

[21] C. George, D. Buschek, A. Ngao, and M. Khamis. Gazeroomlock: Using gaze and head-pose to improve the usability and observation resistance of 3d passwords in virtual reality. In L. T. De Paolis and P. Bourdot, eds., *Augmented Reality, Virtual Reality, and Computer Graphics*, pp. 61–81. Springer International Publishing, Cham, 2020. 2

[22] C. George, M. Khamis, E. Zezschwitz, M. Burger, H. Schmidt, F. Alt, and H. Hussmann. Seamless and secure vr: Adapting and evaluating established authentication systems for virtual reality. 02 2017. doi: 10.14722/usec.2017.23028 2

[23] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of motor behavior*, 19 4:486–517, 1987. 9

[24] A. Gupta, M. Samad, K. Kin, P. O. Kristensson, and H. Benko. Investigating remote tactile feedback for mid-air text-entry in virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 350–360, 2020. doi: 10.1109/ISMAR50242.2020.00062 2

[25] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006. doi: 10.1177/154193120605000909 5

[26] R. S. Hirsch. Effects of standard versus alphabetical keyboard formats on typing performance. *Journal of Applied Psychology*, 54(6):484, 1970. doi: 10.1037/h0030143 9

[27] T. Hirzle, M. Cordts, E. Rukzio, and A. Bulling. A survey of digital eye strain in gaze-based interactive systems. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Full Papers. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3379155.3391313 3

[28] J. Hochreiter, S. Daher, G. Bruder, and G. Welch. Cognitive and touch performance effects of mismatched 3d physical and visual perceptions. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1–386, 2018. doi: 10.1109/VR.2018.8446574 6

[29] B. G. Knapp and M. J. Hall. Human performance concerns for the trackwolf system. 1990. 6

[30] C. Kreider. The discoverability of password entry using virtual keyboards in an augmented reality wearable: An initial proof of concept. 2018. doi: sais2018/23 8

[31] V. Krishna, Y. Ding, A. Xu, and T. Höllerer. Multimodal biometric authentication for vr/ar using eeg and eye tracking. In *Adjunct of the 2019 International Conference on Multimodal Interaction*, ICMI '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3351529.3360655 2

[32] J. Kröger. Unexpected inferences from sensor data: A hidden privacy threat in the internet of things. In L. Strous and V. G. Cerf, eds., *Internet of Things. Information Processing in an Increasingly Connected World*, pp. 147–159. Springer International Publishing, Cham, 2019. 2

[33] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*, SOUPS '07, p. 13–19. Association for Computing Machinery, New York, NY, USA, 2007. doi: 10.1145/1280680.1280683 8

[34] S. Li, S. Savaliya, L. Marino, A. M. Leider, and C. C. Tappert. Brain signal authentication for human-computer interaction in virtual reality. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 115–120, 2019. doi: 10.1109/CSE/EUC.2019.00031 2

[35] Y. Li, Y. Cheng, Y. Li, and R. H. Deng. What you see is not what you get: Leakage-resilient password entry schemes for smart glasses. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, p. 327–333. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3052973.3053042 2

[36] Y. Li, S. Sarcar, K. Kim, H. Tu, and X. Ren. Designing successive target selection in virtual reality via penetrating the intangible interface with handheld controllers. *International Journal of Human-Computer Studies*, 165:102835, 2022. doi: 10.1016/j.ijhcs.2022.102835 2

[37] X. Liu, M. J. C. Crump, and G. D. Logan. Do you know where your fingers have been? explicit knowledge of the spatial layout of the keyboard in skilled typists. *Memory & Cognition*, 38:474–484, 2010. doi: 10.3758/MC

.38.4.474 6

[38] 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*, CCS '15, p. 1273–1285. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2810103.2813668 3, 8

[39] X. Lu, D. Yu, H.-N. Liang, W. Xu, Y. Chen, X. Li, and K. Hasan. Exploration of hands-free text entry techniques for virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 344–349, 2020. doi: 10.1109/ISMAR50242.2020.00061 2

[40] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan. Oculock: Exploring human visual system for authentication in virtual reality head-mounted display. *2020 Network and Distributed System Security Symposium (NDSS)*, 2020. doi: 10.14722/ndss.2020.24079 2

[41] A. Maiti, M. Jadliwala, and C. Weber. Preventing shoulder surfing using randomized augmented reality keyboards. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 630–635, 2017. doi: 10.1109/PERCOMW.2017.7917636 2, 6, 8

[42] R. Miller, N. K. Banerjee, and S. Banerjee. Within-system and cross-system behavior-based biometric authentication in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 311–316, 2020. doi: 10.1109/VRW50115.2020.00070 2

[43] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, p. 323–336. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2307636.2307666 3, 8

[44] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang. Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, HotMobile '12. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2162081.2162095 3, 8

[45] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt. Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300340 2

[46] D.-M. Pham and W. Stuerzlinger. Hawkey: Efficient and versatile text entry for virtual reality. In *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3359996.3364265 6

[47] M. B. Rolf Ploetzner, Sandra Berney. When learning from animations is more successful than learning from static pictures: learning the specifics of change. *Instructional Science*, 49:497–514, 2021. doi: 10.1007/s11251-021-09541-w 6

[48] S. Schneegass, Y. Oualil, and A. Bulling. Skullconduct: Biometric user identification on eyewear computers using bone conduction through the skull. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 1379–1384. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858152 2

[49] D. Schneider, A. Otte, T. Gesslein, P. Gagel, B. Kuth, M. S. Damlakhi, O. Dietz, E. Ofek, M. Pahud, P. O. Kristensson, J. Müller, and J. Grubert. Reconviguration: Reconfiguring physical keyboards in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 25(11):3190–3201, 2019. doi: 10.1109/TVCG.2019.2932239 2, 6

[50] D. Shukla and V. V. Phoha. Stealing passwords by observing hands movement. *IEEE Transactions on Information Forensics and Security*, 14(12):3086–3101, 2019. doi: 10.1109/TIFS.2019.2911171 3

[51] L. Sidenmark and H. Gellersen. Eye, head and torso coordination during gaze shifts in virtual reality. *ACM Trans. Comput.-Hum. Interact.*, 27(1), dec 2019. doi: 10.1145/3361218 8

[52] Z. Song, J. J. Dudley, and P. O. Kristensson. Efficient special character entry on a virtual keyboard by hand gesture-based mode switching. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 864–871, 2022. doi: 10.1109/ISMAR55827.2022.00105 6

[53] R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, p. 113–120. Association for Computing Machinery, New York, NY, USA, 2003. doi: 10.1145/642611.642632 5

[54] M. Speicher, A. M. Feit, P. Ziegler, and A. Krüger. Selection-based text entry in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3174221 2

[55] N. Sulaiman. A study on password security awareness in constructing strong passwords. In A. Khanna, D. Gupta, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, eds., *International Conference on Innovative Computing and Communications*, pp. 421–429. Springer Singapore, Singapore, 2022. 6

[56] K. Viswanathan and A. Yazdinejad. Security considerations for virtual reality systems, 2022. 2

[57] T. Wan, R. Shi, W. Xu, Y. Li, K. Atkinson, L. Yu, and H.-N. Liang. Hands-free multi-type character text entry in virtual reality. *Virtual Reality*, 28(1):8, 2024. doi: 10.1007/s10055-023-00902-z 6

[58] T. Wan, Y. Wei, R. Shi, J. Shen, P. O. Kristensson, K. Atkinson, and H.-N. Liang. Design and evaluation of controller-based raycasting methods for efficient alphanumeric and special character entry in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2024. doi: 10.1109/TVCG.2024.3349428 6

[59] T. Wan, L. Zhang, H. Yang, P. Irani, L. Yu, and H.-N. Liang. Exploration of foot-based text entry techniques for virtual reality environments. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3613904.3642757 2

[60] T. Wan, L. Zhang, X. Yunxin, L. Yu, and H.-N. Liang. Design and evaluation of controller-based raycasting methods for secure and efficient text entry in virtual reality. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2024. 2

[61] H. Wang, T. T.-T. Lai, and R. Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, p. 155–166. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2789168.2790121 3, 8

[62] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. *The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures*, p. 143–146. Association for Computing Machinery, New York, NY, USA, 2011. 5, 7

[63] 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)*, pp. 3382–3398, 2023. doi: 10.1109/SP46215.2023.10179301 2, 3, 5, 8, 9

[64] W. Xu, H.-N. Liang, A. He, and Z. Wang. Pointing and selection methods for text entry in augmented reality head mounted displays. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 279–288, 2019. doi: 10.1109/ISMAR.2019.00026 2

[65] Z. Xu, K. Bai, and S. Zhu. Taplogger: inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, p. 113–124. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2185448.2185465 3, 8

[66] Z. Xu and S. Zhu. Semadroid: A privacy-aware sensor management framework for smartphones. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, p. 61–72. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2699026.2699114 2

[67] H. Yamada. *A historical study of typewriters and typing methods, from the position of planning Japanese parallels*. Journal of Information Processing, 1980. 5

[68] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 4479–4488. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025964 2

[69] R. Zhang, N. Zhang, C. Du, W. Lou, Y. T. Hou, and Y. Kawamoto. Augauth: Shoulder-surfing resistant authentication for augmented reality. In *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017. doi: 10.1109/ICC.2017.7997251 2

[70] H. Zhu, W. Jin, M. Xiao, S. Murali, and M. Li. Blinkey: A two-factor user authentication method for virtual reality devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), dec 2020. doi: 10.1145/3432217 2