De-Anonymizing Avatars in Virtual Reality: Attacks and Countermeasures

Yan Meng, *Member, IEEE*, Yuxia Zhan, *Student Member, IEEE*, Jiachun Li, *Student Member, IEEE*, Suguo Du, Haojin Zhu, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—By providing users with an immersive visual and acoustic experience, virtual reality (VR) serves as a foundational technique for the emerging metaverse. One of the most promising aspects of VR is its ability to protect users' identities by transforming their physical appearances into avatars with arbitrary appearances in the virtual world. However, the increasing threat of de-anonymization attacks that seek to reveal users' identities poses significant privacy risks. We propose AvatarHunter, a non-intrusive and user-unaware de-anonymization attack leveraging victims' inherent movement signatures. AvatarHunter discreetly collects the avatar's gait information by recording videos in the VR scenario without requiring any permissions. Notably, we designed a Unity-based feature extractor that maintains the avatar's movement signature while enabling AvatarHunter to be resistant to changes in the avatar's appearance. We conduct real-world experiments on VRChat to evaluate AvatarHunter's effectiveness. The results demonstrate that in commercial settings, AvatarHunter achieves attack success rates (ASR) of 92.1% and 66.9% in closed-world and open-world avatar scenarios, respectively, significantly surpassing existing benchmarks. Additionally, simulations using an open-source dataset confirm that AvatarHunter can attain over 78% ASR in full-body tracking scenarios. Finally, we discuss several countermeasures and implement an obfuscation mechanism during the avatar rendering phase, significantly reducing the ASR.**

***Index Terms*—De-anonymization attack, identity inference, movement signature, virtual reality.**

I. Introduction

COMPARED with traditional human computer interfaces (e.g., keyboard, mouse, touch screen), virtual reality (VR) offers users an immersed experience by leveraging advanced equipment, including head-mounted displays (HMDs), handheld controllers, eye tracking, full-body trackers, and ambisonics. The metaverse, driven by VR, is regarded as a disruptive technology poised to reshape how humans connect with each other [2]. According to the report published by Grand View Research, the global market size of VR was valued at USD 59.96 billion in 2022 and is expected to reach USD 435.36 billion in 2030, with a compound annual growth rate (CAGR) of 27.5% [3].

In VR applications, the user's real identity could be naturally masked by an *avatar*, a digital representation of the user in the virtual world [4]. Essentially, the avatar has the shape of a human, which could be configured by the users or developers themselves. For instance, in VR games such as VRChat [5], the user's physical appearance is transformed into avatars with cartoonish and fantastic shapes, enabling users to communicate and interact with each other anonymously. Compared with traditional privacy protection methods (e.g., pseudonym user's nickname, customizing voiceprint), the avatar is regarded as an enhanced privacy-preserving solution since the user's appearance in the virtual world is completely hidden. Thus, the avatar gives users a sense of being *unlinkable* to their real identities, leading them to believe they can avoid tracking by potential external adversaries.

Most existing research on the security issues of VR primarily focuses on the functional integrity of VR devices [6], [7] and VR users' access control [8], [9], with privacy concerns receiving less attention [4]. A limited number of studies explore privacy leakage in VR from the perspectives of traffic analysis [10] and side-channel-based de-anonymization [11]. Specifically, the former analyzes traffic payloads from Meta Quest devices, revealing that a victim's identity could be exposed to malicious VR developers [10], while the latter attempts a de-anonymization attack in VR by using a malicious app to capture acceleration and gyroscope data from the HMD for user identification [11]. It is crucial to note that the aforementioned state-of-the-art works on privacy issues have not considered avatars. Furthermore, they all depend on strong adversary models/assumptions, such as malicious app developers or the requirement for data collection from VR devices.

Research motivation: In this study, we propose a de-anonymization attack targeting VR avatars, coined as AvatarHunter. Taking *Ready Player One*, a well-known science fiction film, as an example, the protagonist, Wade Watts, conceals his real identity by altering his avatar's appearance, while the adversary attempts to uncover his true identity. The

Manuscript received 29 February 2024; revised 2 July 2024; accepted 7 July 2024. Date of publication 10 July 2024; date of current version 5 November 2024. This work was supported in part the National Natural Science Foundation of China under Grant 62302298, Grant 62132013, Grant 62325207, and Grant 72171145, and in part by the Startup Fund for Young Faculty at SJTU (SFYF at SJTU) under Grant 23X010502192. A preliminary version titled "De-anonymization Attacks on Metaverse" was published in the IEEE INFOCOM 2023 [DOI: 10.1109/INFOCOM53939.2023.10229062]. Recommended for acceptance by Y. Liu. *(Yan Meng and Yuxia Zhan contributed equally to this work and are co-first authors.) (Corresponding author: Haojin Zhu.)*

Yan Meng, Yuxia Zhan, Jiachun Li, and Haojin Zhu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yan_meng@sjtu.edu.cn; dabeidouretriever@sjtu.edu.cn; jiachunli@sjtu.edu.cn; zhu-hj@cs.sjtu.edu.cn).

Suguo Du is with the Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: sgdu@sjtu.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TMC.2024.3426046

objective of de-anonymization attacks can thus be formalized as follows: given a set of VR avatars and real-world identities, *can an external attacker link a user's real identity to her avatar, even when the avatars observed may be arbitrarily altered or modified?* Such de-anonymization scenarios are not limited to tracking users in an avatar-changeable VR game like VRChat [5] or an anonymous VR meeting [12]. The underlying principle is that in VR environments, regardless of the chosen avatars, the victim's inherent and unique behavior patterns (e.g., gait information leveraged by AvatarHunter) remain relatively stable and can be leveraged to associate the user's avatar with her real identity.

Challenges of de-anonymization: De-anonymizing VR avatars is not a trivial task due to the following challenges. *1) Insight validation.* It is essential to investigate whether the avatar generation process will preserve the user's behavior signatures (e.g., gait information) in the physical world, and whether these signatures remain unique in the virtual world. *2) Requirements for practicality.* Considering most VR applications are closed-sourced, unlike existing de-anonymization schemes [13], [14], [15], [16], our de-anonymization attack should not require intruding into the victim's network nor injecting malicious apps. *3) Variability of avatars.* When extracting a victim's unique behavior signature in the virtual world, the most commonly used schemes (i.e., computer vision-based recognition algorithms [17], [18], [19]) extract both movement and appearance signatures, with the latter being susceptible to changes in the avatar's appearance. Thus, it requires us to extract an effective feature that remains stable when unknown avatars are used by the victim.

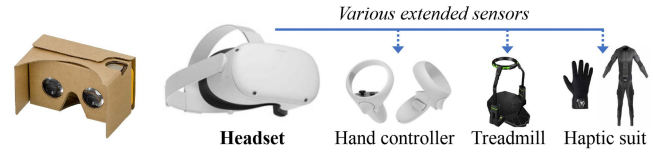
To address the above mentioned three challenges, we conduct following steps. First, we consider a **white-box** VR environment, which means the data from VR sensors (e.g., acceleration) can be obtained, and recruit volunteers to engage this environment with using various avatars. It is observed that the sensor data from different users are quite different, which successfully validate our insight. *Second*, to implement de-anonymization in a non-intrusive manner, we leverage the observation from white-box setting that the video clips of different users contains user behavior signatures. In our proposed AvatarHunter, which only needs to record the video clips when the target victim walks in the open virtual world to infer the identity behind the victim's avatar. Considering there is no existing video-based user recognition schemes for VR scenario, we leverage Unity, the largest platform for developing VR applications [20], to automatically generate abundant gait video from avatars, and constructing the feature extractor based on it. *Finally*, considering existing video-based user recognition schemes, which concentrate on gait samples in the physical world, are sensitive to distortions caused by avatars' changeable appearances, we design a feature extractor enhancement module when extracting the movement signature. The insight is to let the extractor own prior knowledge about various avatars. To achieve it, we leverage Unity to design various avatars and generate the corresponding gait video clips. Then, the feature extractor are enhanced via retraining procedure based on videos originating from various avatars. After that, AvatarHunter is immune to the avatar's appearance changes.

Since AvatarHunter is the pioneering non-intrusive and user-conscious de-anonymization attack in VR, there is no publicly available dataset designed for VR scenarios. Therefore, we have constructed a dataset, named VRC-dataset, utilizing the VRChat application, which is among the most popular VR platforms with a user base of 4 million [21]. We deployed 4 VR accounts acting as *cameras* to record gait video trials, encompassing 100 volunteer-avatar pairs (i.e., 10 recruited volunteers with 10 different avatars each). Participants were required to use the commercial VR device Meta Quest 2, comprising one HMD and two hand controllers. The experimental outcomes reveal that, with the commercial VR device, AvatarHunter can achieve attack success rates (ASR) of 92.1% and 66.9% in closed-world and open-world avatar scenarios, respectively. The experiments also demonstrate AvatarHunter's robustness against various factors, including the number of cameras, input video lengths, and prior knowledge about target victims. To further investigate AvatarHunter's performance in the full-body tracking setting, we build the FBT-dataset, simulated on the Unity platform using the open-source motion dataset CMU-MoCap [22]. The FBT-dataset comprises 100 user-avatar pairs, with avatars animated by body movement data encompassing 31 joints from CMU-MoCap participants. Evaluation results indicate that AvatarHunter can achieve an ASR of 78% in the open-world avatar setting.

Finally, to counteract AvatarHunter, we introduce an obfuscation-based defense mechanism. By integrating noise during the physical-to-virtual conversion process in avatar generation, it becomes challenging for AvatarHunter to accurately extract movement behaviors from the victim's gait video clips. Experiments conducted on the FBT-dataset demonstrate that the ASRs of AvatarHunter in both closed-world and open-world settings drop to 58% and 50%, respectively, when the added noise is subtly controlled to remain unnoticed by users. Our main contributions are summarized below:

- *New attack paradigm.* We introduce AvatarHunter, a non-intrusive and user-unaware de-anonymization attack targeting avatars in VR scenarios. AvatarHunter operates without necessitating access to the victim's VR device or obtaining any permissions within the device's software layer.
- *Novel de-anonymization method.* We propose an innovative approach to extract identity-related movement signatures from avatar video clips. By leveraging numerous gait samples originating from various avatars in the Unity platform, AvatarHunter demonstrates robustness against the avatars' mutable appearances.
- *Open-source dataset.* We provide a dataset contains video trials from 10 users, each with 10 different avatars, collected in real-world VR scenarios. This dataset is accessible to researchers, vendors, and developers to evaluate privacy risks in the metaverse and design countermeasures against de-anonymization attacks.[1]
- *Obfuscation-based countermeasure.* We introduce a defense mechanism that incorporates noise during the

[1]The dataset will be provided upon email request.



(a) Cardboard. (b) Standalone VR.

Fig. 1. VR device types: Cardboard and standalone devices.

avatar movement rendering process. Experimental evidence shows that the proposed countermeasure can significantly diminish AvatarHunter's efficacy.

The remainder of this paper is organized as follows. Section II introduces the necessary preliminary knowledge. Section III describes the threat model and motivation. Section IV presents the system design of AvatarHunter, followed by evaluations, countermeasures, and discussion in Sections V, VI, and VI, respectively. Finally, Section VIII concludes this work.

II. Preliminaries and Related Works

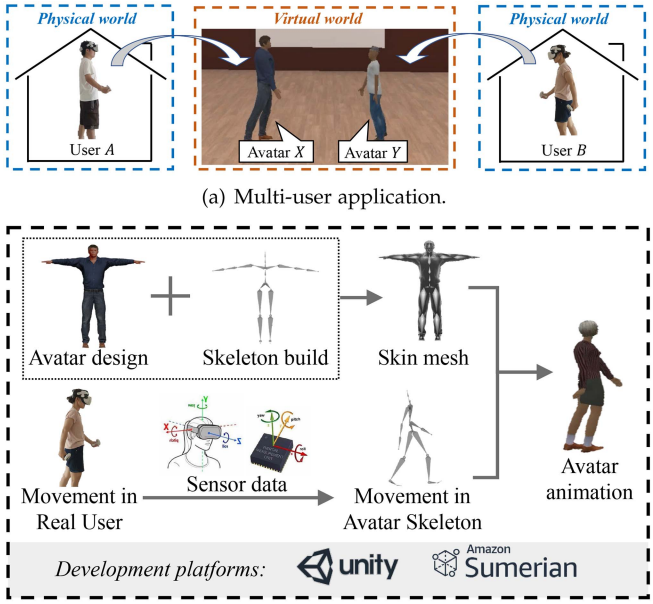
In this section, we first introduce mainstream VR devices and current multiple-user VR application platforms. We then elaborate the anonymization mechanism (i.e., avatar) in VR applications. Finally, we review existing de-anonymization attacks in VR scenarios.

A. Devices and Applications in VR

VR devices: In consumer markets, VR devices can be divided into two types: *cardboard devices* and *stand-alone devices*. As illustrated in Fig. 1(a), the former acts only as a screen displaying 3D immersive visual contexts, while the latter, which is the research target of our study, not only provides immersive visual contexts but also embeds various sensors to enhance the user's interactive experiences. As shown in Fig. 1(b), current popular VR devices (e.g., Meta Quest 2 [23], Apple Vision Pro [24], HTC VIVE Pro 2 [25], PICO [26]) are equipped with general-purpose sensors (e.g., gyroscope, accelerometer) and also work in conjunction with various human-interactive sensors (e.g., hand controllers, treadmills, haptic suits) to provide a more immersive experience for users.

In this study, we choose a stand-alone device named Meta Quest 2 as our study object.[2] Quest series VR devices are developed by Meta, which owns 49% of the global VR headset market in Q3 2023 [27]. Additionally, other popular VR devices including the PICO series and HTC VIVE series have a similar architecture to Quest. It consists of a system on chip (SoC) for computation, an LCD for displaying visual contents, and a microphone-loudspeaker pair for voice channel interaction. Quest not only supports VR applications in its official store but also games from third-party VR application platforms (e.g., Steam VR, SideQuest). Furthermore, Quest can be connected to a laptop/desktop to provide the developer with a view of its

[2]In the following sections, we utilize the terminology *Quest* to refer to Meta Quest 2.



(a) Multi-user application.

(b) Avatar generation process.

Fig. 2. Overview of VR application and avatar.

LCD screen (i.e., the view watched by the user) for game video recording or application debugging.

Multi-user VR applications: VR applications attract an increasing number of users to participate and enjoy various functions. As illustrated in Fig. 2(a), in the multi-user VR application scenario, users in the physical world log into the VR application with the assistance of VR devices. After that, users can perform various actions including chatting, exercising, and playing games in the virtual world. To protect the user's identity privacy and increase entertainment effects, the VR platform allows users to transform themselves into various *avatars* to hide their real-world appearances. Additionally, users can also change their voices to further enhance their anonymity.

B. The Avatar Mechanism in VR

As illustrated in Fig. 2(b), users can leverage avatars to change their appearance and hide their real-world identities in the virtual world. In this subsection, we introduce the avatar generation procedure, movement mechanism, and development platforms.

Avatar generation procedure: In most VR applications, the avatar is generated based on the skeleton animation mechanism [28] consisting of three steps. *i) Designing the avatar model:* The designer first creates the static character sketch of the avatar and then models it with abundant mesh structures in 3D virtual space. *ii) Building the skeleton structure:* For the generated 3D model of the avatar, a skeleton is defined for action execution. The avatar's skeleton, having a parent-child structure, acts as the avatar's bones and joints during movement. Specifically, when the parent node moves, the child node executes the corresponding movement according to mapping rules. *iii) Mapping skeleton and skins:* The skin represents the avatar's external appearance in the virtual world. After constructing the skeleton,

the properties of the skin (e.g., positions, colors, materials, ornaments) related to the skeleton are determined. Movements of the avatar's skeleton lead to corresponding movements in the skin.

Converting sensor data from VR devices to avatar movements: Unlike traditional desktop applications where user input devices are the keyboard and mouse, in VR scenarios, the avatar's movement is closely tied to the user's physical movement behaviors. During the user playing in the VR application, the data collected from the equipped VR helmet and controller handles are mapped to the avatar's skeleton. The skeleton, driven by motion data, moves accordingly, and its related skins are rendered in the virtual world.

VR avatar/application development platforms: Several platforms assist VR developers in quickly generating and activating avatars. Unity is a typical platform, with 60% of popular VR games developed using it [29]. Unity also enables developers to set up virtual scenarios for VR applications and easily construct and control the virtual appearance of avatars in C#-based programs [30]. We utilize the Unity platform for extracting the victim's features as described in Section IV-C.

C. Existing Privacy Inference Attacks in VR

In this section, we review existing privacy inference attacks (e.g., de-anonymization, sensitive information inference) in VR scenarios. We begin with an overview of user recognition schemes from both sensor-based and video-based perspectives. Subsequently, we discuss other privacy breach issues.

Sensor-based user recognition: A bunch of works utilize sensor data for user recognition, which aids in designing secure and efficient authentication schemes in VR scenarios [8], [31], [32], [33], [34], [35], [36]. OcuLock [8] and GaitLock [32] employ users' eye movements triggered by immersive 3D visual content and gait signatures recorded by onboard IMUs to respectively recognize logging-in users. Additional interactions, such as pointing, grabbing, typing with controllers, bowling, and shooting arrows [14], [37], [38], can also be harnessed to construct authentication schemes. Notably, Nair et al. demonstrated the feasibility of inferring a user's identity with over 90% accuracy among a candidate pool of 50,000+ individuals based on motion data in VR games lasting for 100 seconds [39]. However, these user recognition schemes heavily rely on accurate sensor data extracted from built-in sensors in VR devices, limiting their application in remote attacking scenarios.

Video-based user recognition: The research community has also explored computer-vision-based user recognition tasks using videos of users walking in different settings, such as lab environments [40], markets [41], and university campuses [42]. Researchers [17], [18], [19] have achieved significant performance on these specific tasks. However, as analyzed in Section IV-C, these videos (image sequences) are captured in the physical world and contain both the appearance and movement signatures of users. This characteristic poses challenges for their application in VR scenarios, especially when users can alter their appearances by adopting different avatars. ReAvatar [43] leverages videos in VR for user recognition. However, ReAvatar's practicability is limited as it requires the adversary to entice the user into explicitly performing actions, which can easily arouse the user's suspicion.

Other privacy breaches in VR: Recent studies point out that privacy information, including text inputs, locations, and speeches, can be compromised by attacks. Arafat et al. [44] leverage the channel state information (CSI) in VR device wireless communication to track user gestures and infer text inputs. Kotaru et al. [45] also use CSI for tracking user locations in VR scenarios. Shi et al. [11] infer user speeches using vibration data collected by sensors in the VR device. However, these methods either necessitate intruding into the target victim's network or luring victim to install malicious applications.

III. Attack Insight and Threat Model

In this section, we first demonstrate the feasibility of AvatarHunter, the de-anonymization attack proposed in this study, through two case studies. Then, we formally define the threat model and the capabilities of the adversary. Finally, we elaborate on the detailed attack settings encountered by AvatarHunter.

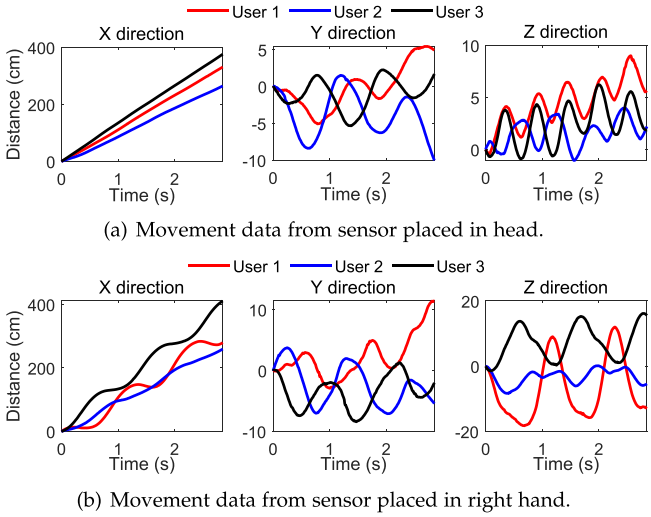
A. Basic Insight of AvatarHunter

In this subsection, we elaborate on the insight of AvatarHunter by answering the following key research question: *When victims use avatars with different appearances to hide their identities in VR applications, is it possible to link their movement signatures to their identities?* We present two case studies and make the following observations.

Observation 1. Movement data from sensors of VR devices clearly reveal the user's identity.

It is well known that different users exhibit unique movement patterns (e.g., gait information) during their movement phases (e.g., walking) due to their individual habits. To investigate the feasibility of leveraging gait information to identify user identities in VR, we first analyze the movement data collected by sensors of a VR device. Considering a commercial VR device as described in Section II-A, which comprises a headset and two hand controllers, we examine a third-party dataset named CMU-MoCap [22]. CMU-MoCap includes movement data from sensors placed on various body parts of participants, effectively simulating VR scenarios. Fig. 3 illustrates the sensor data from three different participants during walking, with sensors located on the head and the right hand. It is important to note that the X direction corresponds to the participant's walking direction.

It is observed from Fig. 3 that movement patterns between different users are quite distinctive. More specifically, the movement data from the participant's right hand show more fluctuation than those from the head. For example, in the Z direction, which is vertical to the ground, the range of hand movement data as depicted in Fig. 3(b) is significantly larger than that of head movement as shown in Fig. 3(a). This discrepancy arises because a user can move her hand more freely while maintaining a relatively stable head posture. In summary, the movement data, which can be collected by a VR device and mapped to an avatar's



(a) Movement data from sensor placed in head.

(b) Movement data from sensor placed in right hand.

Fig. 3. The relationship between VR sensor data and user identities.

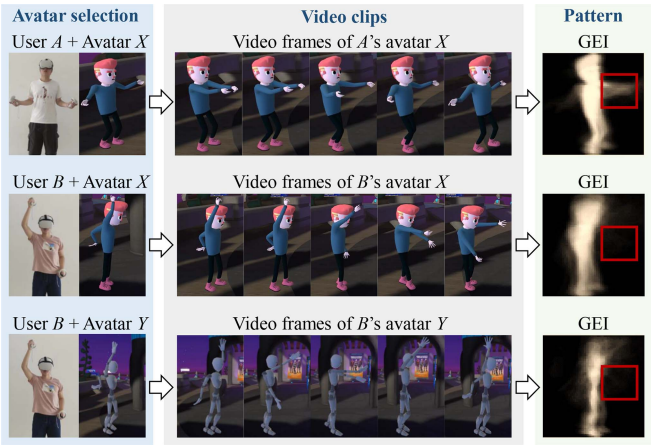


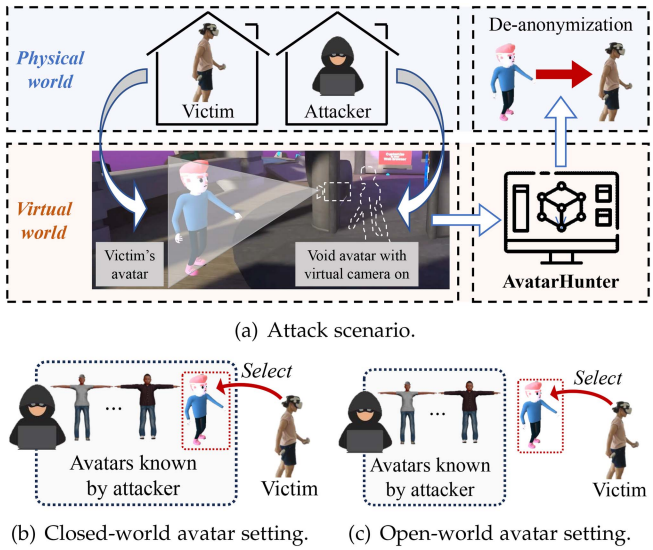
Fig. 4. The relationship between avatar movements and user identities.

movements, contains distinct user identities and opens a door for de-anonymization attacks.

Observation 2. Even when different avatars are deployed, the video clip of the gait still preserves the user's identity.

Directly leveraging the aforementioned observation 1 to launch the de-anonymization attack encounters practical issues, as obtaining sensor data from a remote victim's VR devices may not be feasible. Therefore, we conduct a case study to demonstrate how video clips containing the gait of a victim's avatar could be utilized for identity recognition. In this case study, two volunteers (i.e., user A and user B) are recruited and required to log into a VR application (i.e., VRChat). Initially, these two volunteers use the same avatar (i.e., avatar X) to walk for several seconds while we deploy a virtual camera to record their gait video simultaneously. Subsequently, user B uses another avatar (i.e., avatar Y) and walks again.

Fig. 4 illustrates the collected video frames and their extracted gait energy images (GEIs), which are representations used to characterize human walking properties [46]. It is observed that



(a) Attack scenario.

(b) Closed-world avatar setting. (c) Open-world avatar setting.

Fig. 5. Attack model.

even though two users have the same appearance in VR, their distinct gait behaviors result in significant differences in their GEIs, as highlighted in red rectangles. In the case where user B uses avatar Y, it is observed from the red rectangles in Fig. 4 that the GEI is similar (although slightly different) to that when user B uses avatar X, but quite different from that of user A. Therefore, it is feasible for AvatarHunter to reveal the avatar's identity based on the observed gait video clips during walking.

B. Attack Formulation

1) Threat Model: As illustrated in Fig. 5(a), while launching AvatarHunter, the victim is engaged in playing a VR game (e.g., VRChat [5]) and has transformed her appearance into an avatar. Meanwhile, the adversary logs into the same VR application in which the victim is playing and begins monitoring the victim's behaviors by recording the video displayed on the VR device's screen. Since this recording procedure can occur without any participation from the victim, the victim remains unaware of it and believes her identity is securely hidden behind the avatar. However, from the adversary's perspective, after completing the collection of the victim's video clips, the adversary can initiate the de-anonymization attack based on the extracted biometric signature related to the victim's identity.

2) Adversary's Capabilities: We assume the adversary has the following capabilities during the attack preparation and implementation phases.

Attack preparation phase: acquiring avatar's video with identity label (i.e., gallery construction): It is well known that in de-anonymization attacks [11], [43], the adversary has to obtain the victim's prior knowledge such as video clips and identity labels in advance. Note that, in practice, there are several ways to obtain the victim's gait video clips. For instance, when the target victim shows her identity in the nickname, the adversary can actively record the video of the victim's avatar walking movement. The attacker can also obtain such video clips from

the victim's social network contents. We regard the pre-collected video clips as *gallery* in this study.

Attack implementation phase: recording avatar's video without identity label: After constructing the gallery, the adversary enters the same VR application as the target victim. For any suspicious avatar, the adversary starts recording the video and tries to identify whether the real identity behind such an avatar is the victim. Note that recording a video does not require any permission or approval from the victim. There is also no requirement for the background scene displayed in the video. Take a popular VR chatting application, VRChat, as an example. The adversary can focus her sight on the suspicious avatar's movement and record the sight view (i.e., recording the contents of the LCD screen of Quest or the remote synchronization on the desktop). Furthermore, as illustrated in Fig. 5(a), to avoid raising any suspicion from the victim, the adversary can transform herself into an invisible camera by employing an avatar with a void appearance (i.e., the adversary's avatar becomes invisible).

3) Attack Settings: We divide AvatarHunter's attack scenarios into following attack settings according to the adversary's capabilities and VR device types. On the one hand, according to whether the adversary has the knowledge about the avatar deployed by the victim, AvatarHunter has the following attack settings:

Closed-world avatar setting: Before launching the attack, the adversary collects the victim's gait videos (gallery) in which a set of several avatars are utilized. Then, as shown in Fig. 5(b), the victim hides her external information (e.g., nickname) and changes her avatar into an avatar inside the gallery. For the adversary, the goal is to identify the victim's identity after observing the new arriving gait video clip.

Open-world avatar setting: As illustrated in Fig. 5(c), the victim can hide her external information and utilize a *novel* avatar outside the gallery in the closed-world setting. Compared with scenario-1, it is harder for AvatarHunter to infer the real-world identity since the pre-collected gallery contains no prior knowledge about the victim's gait behaviors in this novel avatar. Note that, as described in Section III-B, we assume the adversary can learn the details about the novel avatar used by the victim after investigating the collected video clips.

Besides, based on the data granularity of the victim's VR devices, we can further categorize AvatarHunter's attack scenarios into **commercial device setting** and **full-body tracking setting**. In the commercial device setting, VR devices only track partial body movements of the victim, resulting in avatar motions that are not always consistent with the victim's real-world physical movements. For instance, the Quest device, comprising only three individual parts (i.e., headset, left-hand controller, and right-hand controller), causes some movements of avatars corresponding to Quest users, such as leg movements, to be rendered by the VR application itself without actual sensor data, failing to preserve the user's precise physical movement signature. Conversely, in the full-body tracking setting, sensors placed on various body parts of the victim (e.g., using a haptic suit) are utilized, and the avatar's movement encapsulates more detailed information from the victim's real-world body movement.

IV. System Design of AvatarHunter

In this section, we introduce the detailed system design of AvatarHunter. As illustrated in Fig. 6, AvatarHunter comprises four modules: *Attack Initialization Module*, *Data Preprocessing Module*, *Gait-based Feature Extraction Module*, and *Identity Recognition Module*.

A. Attack Initialization Module

Before launching the de-anonymization attack, AvatarHunter constructs a *gallery* containing prior knowledge of the victim. Then, while the victim engages in VR applications, AvatarHunter imperceptibly collects video clips.

1) Gallery Construction During Attack Preparation: As described in Section III-B, constructing a gallery is crucial for de-anonymization attacks because, without it, AvatarHunter lacks prior knowledge of the target victim's identity. To build the gallery in this study, a common method for the adversary is to log into the VR application where the victims reveal their identities, to collect the gait video clips and identity labels. Consequently, AvatarHunter compiles a gallery comprising several users, including the target victim, before initiating de-anonymization attacks. The ith element of the gallery G_i is denoted as a 3-tuple $\langle V_{G,i}, A_{G,i}, I_{G,i}\rangle$, where $V_{G,i}$ represents the collected video clips, $A_{G,i}$ is the avatar used in this video, and $I_{G,i}$ is the identity of the gallery member.

2) Recording the Victim's Video in the Virtual World: After constructing the gallery for the targeted victim, AvatarHunter deploys *virtual cameras* to collect her movement video clips within VR applications. As illustrated in Fig. 7, the victim enters the VR application, hides her identity by anonymizing her external information (e.g., nicknames, user profiles), and employs various avatars. AvatarHunter uses N_V accounts to log into the VR application to monitor the victim's behaviors from N_V perspectives, as described in Section III-B and Fig. 5(a). Additionally, to record videos imperceptibly, AvatarHunter configures the appearances of N_V avatars as invisible. As shown in Fig. 7, the collected videos feature N_V perspectives, which can be denoted as *gait video trial* $V = \{V_1, V_2, \ldots, V_{N_V}\}$, where V_i is the video from the ith invisible camera. Furthermore, V_i comprises $M = T \times S$ frames, where T is the recording duration and S is the frame rate.

B. Data Pre-Processing Module

To enhance the de-anonymization performance, AvatarHunter pre-processes the collected video trial V before forwarding it to the subsequent module.

1) Background Elimination: For a given video trial V containing $M \times N_V$ frames, to mitigate the distortion caused by the VR application's background images on the extraction of movement signatures, AvatarHunter initially removes the background from these frames, preserving only the information pertinent to avatar movement. First, for the video from a specific view, AvatarHunter captures a background image B_I at a moment when the victim's avatar is absent during data collection. Then, utilizing the acquired B_I, AvatarHunter applies the background

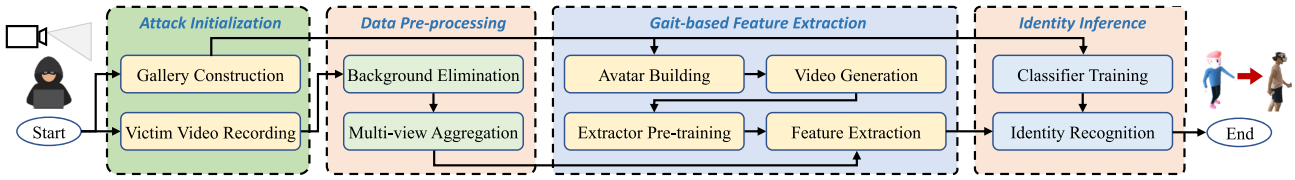


Fig. 6. System overflow of AvatarHunter.

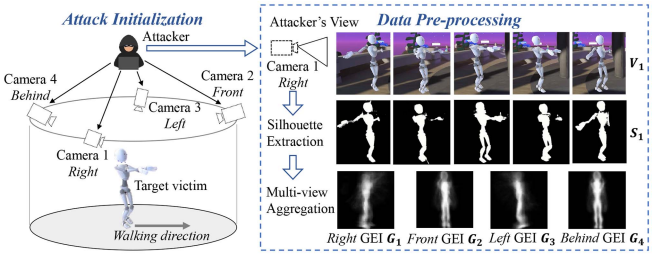


Fig. 7. An illustration of attack initialization and data pre-processing.

matting scheme proposed by Lin et al. [47] to each frame in the video, obtaining its corresponding *silhouette*. Consequently, for the collected V, AvatarHunter generates a sequence of silhouettes S with dimensions $M \times N_V$, where each element is:

$$s_{i,j} = B_E(v_{i,j}, B_I), \tag{1}$$

with $v_{i,j}$ representing the ith frame from the jth view's video V_j, $s_{i,j}$ denoting $v_{i,j}$'s corresponding silhouette, and $B_E(\cdot)$ is the background elimination operation as described in [47]. As illustrated in Fig. 7, the victim avatar's movement behaviors become markedly evident after performing background elimination.

2) Multi-View Data Aggregation: As mentioned in Section IV-A, AvatarHunter utilizes multiple cameras to collect a diverse set of data to enhance the performance of the de-anonymization attack. Consequently, AvatarHunter aggregates data from multiple views as input for the subsequent module.

Given that multiple cameras are positioned at different locations, offering varied sight views and angles, AvatarHunter standardizes silhouettes from different cameras. For each silhouette $s_{i,j}$, AvatarHunter first calculates its centroid $c_{i,j}$. Then, AvatarHunter trims the margins of $s_{i,j}$ to derive a new silhouette $s'_{i,j}$ where $c_{i,j}$ is centered. To demonstrate that utilizing multiple cameras can provides more detailed information, for the ith camera, we compute the GEI G_i from silhouettes $\{s'_{1,i}, s'_{2,i}, \ldots, s'_{M,i}\}$ as follows:

$$G_i = \frac{1}{M} \sum_{j=1}^{M} s'_{j,i}. \tag{2}$$

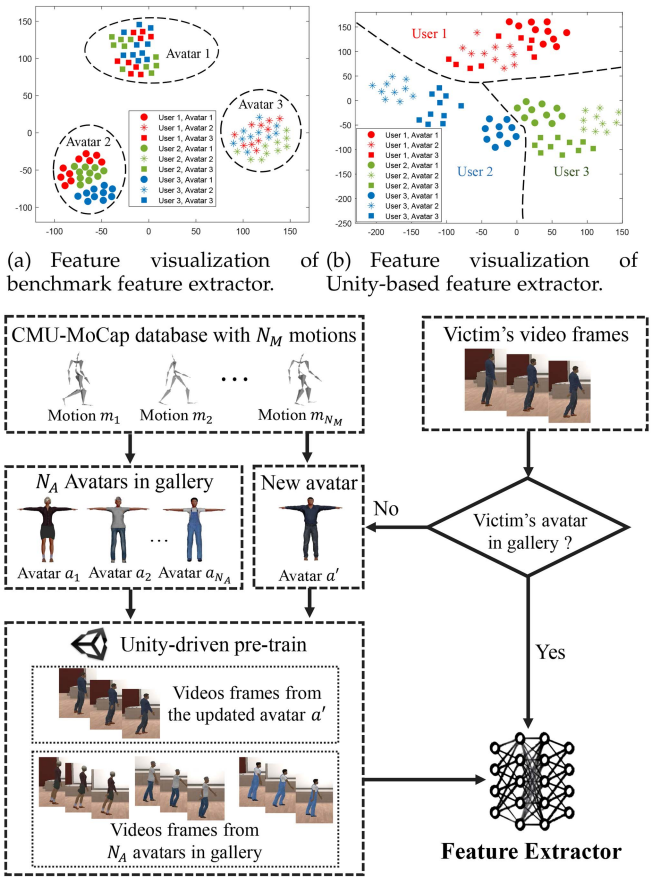
As illustrated in Fig. 7, deploying multiple cameras can provide diversified information reflected by avatar movement. For instance, with $N_V = 4$ cameras in use, the front perspective reveals the existence of leg movements, but the amplitude of these movements is distinctly visible only from the side perspectives (i.e., GEIs from the left and right cameras). Ultimately, for the collected video trial V with dimensions $M \times N_V$, AvatarHunter processes the pre-processed silhouettes S' as the input for the feature extraction module, with S' represented as follows:

$$S' = \begin{bmatrix} s'_{1,1}, s'_{1,2}, \ldots, s'_{1,N_V} \\ s'_{2,1}, s'_{2,2}, \ldots, s'_{2,N_V} \\ \cdots \\ s'_{M,1}, s'_{M,2}, \ldots, s'_{M,N_V} \end{bmatrix}. \tag{3}$$

C. Gait-Based Feature Extraction Module

After pre-processing the collected video trial, AvatarHunter must extract features from the victim avatar's gait for final identity inference. Existing video-based gait feature extractors (e.g., GaitSet [19]) typically utilize deep learning models pre-trained on a public gait video dataset (CASIA-B gait dataset [40]). It is noteworthy that the video trials in the training dataset used to pre-train the feature extractor differ from those in the gallery. In real-world user identification tasks, it is hard to obtain a gallery with a vast number of video clips to fine-tune the deep learning model. Moreover, the gaits used for pre-training existing extractors (e.g., GaitSet) are gathered in the physical world (i.e., without employing avatars, and the appearances of different real human volunteers vary significantly) rather than in the virtual world. Therefore, when applying the existing feature extractor to the input S', the extracted biometric signature includes both appearance and movement signatures, where the former becomes unstable with avatar changes. Fig. 8(a) illustrates the features extracted by GaitSet, the benchmark feature extractor in this study, when three volunteers use three different avatars. Features are projected into 2D space using the t-Distributed stochastic neighbor embedding (t-SNE) method [48]. It is observed that features extracted by the benchmark extractor cluster based on the avatars' appearances rather than the volunteers' movement signatures.

To address the challenges posed by the avatar's changeable appearances, as depicted in Fig. 8(c), we adjust the feature extractor's pre-training procedure. Instead of relying on the public gait dataset collected in the physical world, we utilize a virtual world gait dataset that we have created. In our VR gait dataset, various avatars are designed and animated based on users' movement data within the Unity platform, adhering to the procedures outlined in Section II-B. This retraining approach allows the scheme to concentrate on the identity behind the avatar, leveraging prior knowledge of the user's movement signature across different avatar appearances. The detailed steps are presented below.



(a) Feature visualization of benchmark feature extractor. (b) Feature visualization of Unity-based feature extractor.

(c) The pipeline of constructing the Unity-based feature extractor.

Fig. 8. Illustrations the gait-based feature extraction module.

1) Avatar Construction in Unity Platform: To make AvatarHunter immune to the avatar's changeable appearances, we manually construct various avatars in the Unity platform. Specifically, AvatarHunter constructs N_A avatars occurring in the *gallery*, denoted as $A = \{a_1, a_2, \ldots, a_{N_A}\}$. For the ith target avatar a_i, if it is open-source, we directly import its source file (e.g., .FBX or .MHX2 file) into the Unity platform. Otherwise, we utilize a popular 3D character making tool, MakeHuman [49], to design the closed-source avatar a_i in Unity. Note that, by using MakeHuman, for a given avatar template, we can adjust the parameters and construct avatars with various appearances, such as tall, strong, fat, and child, as illustrated in Fig. 9(a).

2) Gait Video Clips Generation: To animate avatars in Unity based on real human movements, we employ Blender [50] to map sensor data collected during human walking onto the avatar's skeleton. Each avatar features a predefined skeleton in Unity. For example, as shown in Fig. 9(b), an avatar might have a skeleton with 31 nodes. The sensor data for this study come from the third-party CMU-MoCap dataset [22], which uses 41 sensors to track participants' movements. Since AvatarHunter focuses on the victim's gait information, we select sensor data from $N_M = 136$ walking samples involving 29 participants from CMU-MoCap. These motion data samples are represented as $M = \{m_1, m_2, \ldots, m_{N_M}\}$. For avatar animation, we use the Rokoko plugin to map the motion data from 31 sensors to the skeleton, creating animations for the avatars [51].

To record video clips from multiple perspectives, we deploy eight cameras with angles between the avatar and cameras set at $0°$, $72°$, $90°$, $144°$, $180°$, $216°$, $270°$, and $288°$, respectively. For automatic tracking of avatar movements in Unity, we employ the Cinemachine plugin [52]. Following background elimination, we compile an image dataset suitable for gait recognition, as illustrated in Fig. 9(c). Finally, after applying the N_M sensor data on N_A avatars in the gallery, we totally generate the $N_M \times N_A$ eight-view videos P_{pre} as below:

$$P_{pre} = \Phi(A, M) = \begin{bmatrix} P_{1,1}, P_{1,2}, \ldots, P_{1,N_A} \\ P_{2,1}, P_{2,2}, \ldots, P_{2,N_A} \\ \cdots \\ P_{N_M,1}, P_{N_M,2}, \ldots, P_{N_M,N_A} \end{bmatrix}, \quad (4)$$

where $P_{i,j}$ is the video with eight views when applying ith motion m_i data on jth avatar a_j.

3) Feature Extractor Pre-Training: We consider the generated $N_M \times N_A$ eight-view videos as the pre-training dataset. Notably, each video in the dataset lasts for several minutes, significantly longer than those in the gallery (e.g., typically just a few seconds). The pre-training dataset is divided into training and validation sets following a 9:1 ratio. During pre-training, video clips are transformed into silhouettes as described in Section IV-B, and the model structure of the feature extractor is aligned with that of GaitSet. The feature extractor of AvatarHunter is capable of processing silhouettes from any camera view and outputs features with a uniform dimension.

4) Extracting Feature From Input: For a given collected gait video trial V and its pre-processed silhouettes S', if the avatar in V is present in the gallery (i.e., closed-world avatar setting), AvatarHunter uses S' as input to obtain feature F. Alternatively, in the open-world avatar setting, we introduce the novel avatar a' into the Unity platform and replicate the aforementioned three steps. The updated pre-training dataset can be represented as:

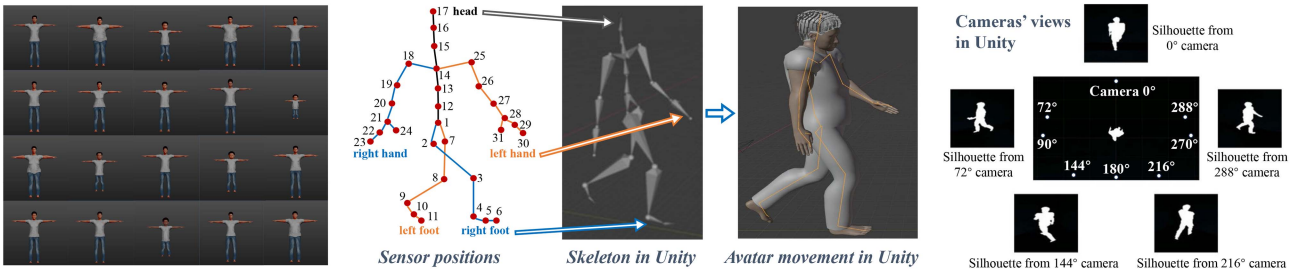
$$P'_{pre} = \Phi(A \cup a', M). \quad (5)$$

Ultimately, AvatarHunter extracts feature F from the updated feature extractor pre-trained with P'_{pre}. Fig. 8(b) illustrates the features generated by our Unity-based feature extractor. Compared with the benchmark feature extractor (i.e., GaitSet) as shown in Fig. 8(a), AvatarHunter is robust to avatar appearance changes and effectively characterizes the user's inherent movement signature.

D. Identity Inference

Finally, after obtaining the feature F that characterizes the inherent movement signature of the target user, AvatarHunter generates a classifier Ψ based on the gallery G and then determines the identity of F.

1) Gallery-Based Classifier Generation: Given the gallery G containing N_U users' identities, we denote the ith element as $G_i = \langle V_{G,i}, A_{G,i}, I_{G,i} \rangle$, where $V_{G,i}$ is the video clip, $A_{G,i} \in \{a_1, a_2, \ldots, a_{N_A}\}$ represents the avatar, and $I_{G,i} \in$



(a) Avatar's Multiple appearances. (b) Mapping motion sensor data to avatar movement. (c) Recording gait video in Unity.

Fig. 9. Generating various avatars' movements in the Unity platform.

$\{u_1, u_2, \ldots, u_{N_U}\}$ means the identity label. To construct the classifier, AvatarHunter transforms G into G', where the ith element G_i is represented as:

$$G'_i = H(G_i) = \langle E(V_{G,i}), I_{G,i} \rangle, \tag{6}$$

with $H(\cdot)$ being the transformation operation and $E(\cdot)$ the feature extraction function as defined in Section IV-C. Subsequently, a random forest model is employed, using G' as the input, to generate the classifier Ψ.

2) Identity Recognition: Finally, for the F extracted from V, AvatarHunter utilizes Ψ to determine its identity J as below:

$$\Psi(F) = J, \tag{7}$$

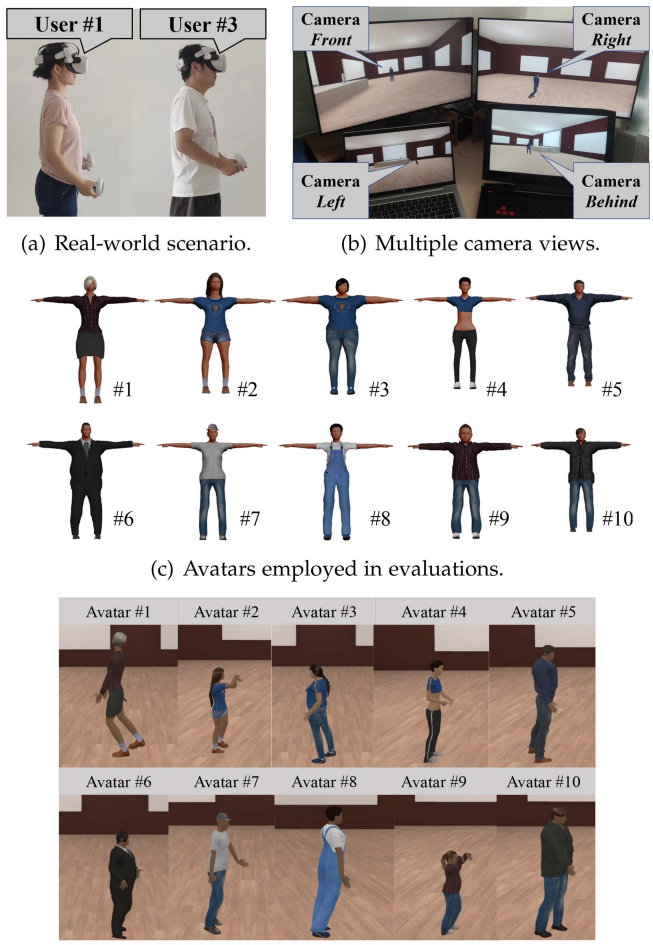
where $J \in \{u_1, u_2, \ldots, u_{N_U}\}$. If J is the victim's real identity, AvatarHunter conducts the de-anonymization attack successfully.

V. Evaluation

In this section, we deploy AvatarHunter on a widely-used real-world VR application to evaluate its overall attack performance. Subsequently, we examine the influence of various factors on AvatarHunter's effectiveness. Finally, using a third-party open-source dataset and the Unity platform, as detailed in Section IV-C, we investigate AvatarHunter's efficacy in a full-body tracking context.

A. Real-World Evaluation Setup

Considering there is no public gait dataset collected in real-world VR scenarios, this study constructs the first dataset for evaluating AvatarHunter. We primarily assess the de-anonymization performance of AvatarHunter in VRChat, a popular VR application developed for real-time social interactions. During the experiments, 10 volunteers are recruited to construct the dataset. As shown in Fig. 10(a), each volunteer is required to wear the Oculus Quest 2 device and then enter VRChat. To initiate the attack, as the volunteer navigates the virtual environment, four cameras are positioned in the front, left, right, and behind the volunteer to record the video simultaneously. Fig. 10(b) displays frames from four different perspectives at a specific moment. Throughout the data collection phase, each volunteer is instructed to use 10 avatars, as depicted in Fig. 10(c),

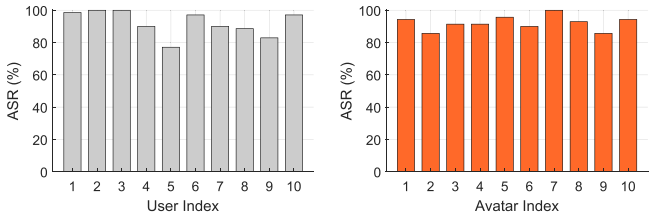


(a) Real-world scenario. (b) Multiple camera views. (c) Avatars employed in evaluations. (d) Various avatars' video clips.

Fig. 10. Real-world evaluations setup.

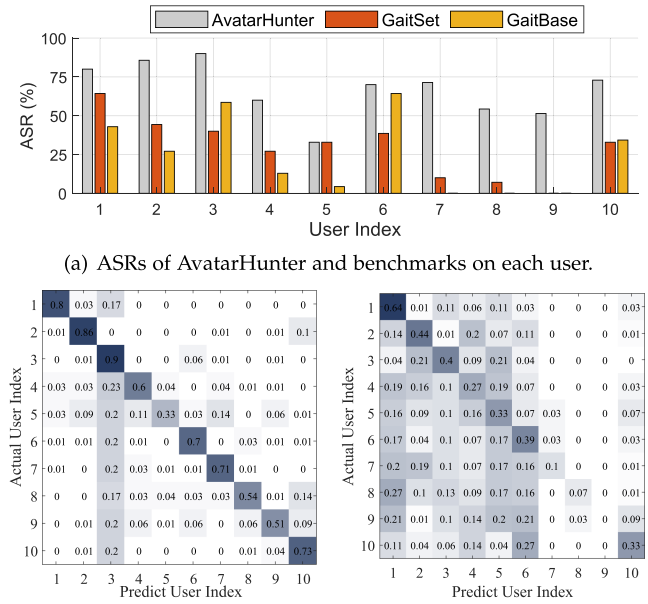
and walk 10 times using each avatar. Fig. 10(d) presents the collected video gait clips for various avatars.

Dataset Description and Evaluation Metric: In total, we have collected 10 users $\times$10 avatars $\times$10 trials $=$ 1000 gait video trials, each containing 4 videos from different perspectives, as illustrated in Fig. 10(b). For each perspective, the video lasts at least 4.5 seconds at 30 frames per second. During the evaluation, for each avatar used by each volunteer, we randomly select



(a) ASRs on each user. (b) ASRs on each avatar.

Fig. 11. Performance on closed-world avatar setting.



(a) ASRs of AvatarHunter and benchmarks on each user.

(b) AvatarHunter's confusion matrix. (c) GaitSet's confusion matrix.

Fig. 12. Performance on open-world avatar setting.

3 trials as the gallery and consider the remaining 7 trials as AvatarHunter's testing dataset, dubbed the VRC-dataset. Thus, the gallery and testing dataset comprise 300 and 700 trials, respectively. In our experiments, we adopt the attack success rate (ASR) as the evaluation metric, defined as $\frac{N_S}{N_T}$, where N_T represents the total number of attack attempts and N_S is the number of times AvatarHunter successfully identifies the victim's identity.

B. Overall Performance of AvatarHunter

In this subsection, we introduce AvatarHunter's performance under closed-world and open-world avatar settings.

1) Performance Under Closed-World Avatar Setting: In this scenario, all avatars present in the testing dataset are included in the gallery. To assess AvatarHunter's efficacy, we train the classifier using the gallery with 300 trials and perform identity inference on the testing dataset comprising 700 trials. For each trial with four views, 30 frames (*i.e.*, equivalent to 1 s) from each view are selected as AvatarHunter's input. Fig. 11(a) shows the ASR across volunteers. The overall ASR stands at 92.1%, with ASRs among different volunteers ranging from 77.1% for user #5 to 100.0% for user #2 and user #3. Even in the least favorable scenario, the ASR significantly exceeds the random guess (*i.e.*, 10% in this experimental setup). Analyzing AvatarHunter's performance across different avatars, as depicted in Fig. 11(b), reveals variability in effectiveness depending on the avatar selected. ASRs among different avatars range from 85.7% for avatar #2 and avatar #9 to 100% for avatar #7. In the least favorable case, the ASR still greatly surpasses the random guess. This underscores AvatarHunter's capability in de-anonymizing user identities in VR scenarios.

2) Performance Under Open-World Avatar Setting: In the open-world setting, the victim's avatar is not represented in the gallery. Thus, for each avatar, we omit its corresponding 30 trials from the gallery and retrain AvatarHunter's classifier with the revised gallery. The classifier is then applied to the remaining 70 trials for that avatar. The overall ASR observed is 66.9%, with ASRs among different volunteers ranging from 51.4% for user #9 to 90.0% for user #3. This performance, though inferior to that in the closed-world setting (i.e., an overall ASR of 92.1%), occurs because the gallery lacks prior knowledge about the avatar used by the victim.

We also compare AvatarHunter with an existing popular gait-based identity inference schemes. Besides GaitSet [19] introduced in Section IV-C, we further consider a state-of-the-art gait recognition model called GaitBase [53]. GaitBase is a flexible and efficient gait recognition codebase that achieves 96.5% rank-1 accuracy on the CASIA-B dataset, outperforming GaitSet and other gait recognition models like GaitPart [54]. As illustrated in Fig. 12(a), under the open-world avatar setting, AvatarHunter achieves an average ASR of 66.9%, which is significantly larger than that of GaitSet (29.7%) and GaitBase (24.4%). Among the 10 employed users, AvatarHunter achieves the highest ASR of 90.0% for user #3 and the lowest ASR of 32.9% for user #5. Meanwhile, the best and worst ASRs for the benchmarks are 64.3% for user #1 and 0.0% for user #9, respectively. Furthermore, Fig. 12(b) and (c) display the confusion matrices for AvatarHunter and GaitSet. It is noted that, compared with the benchmark, AvatarHunter features most of its darker areas along the diagonal of the matrix, indicating a higher accuracy in correct identity prediction. AvatarHunter outperforms the benchmark because its feature extractor is pre-trained using diverse avatar videos generated on the Unity platform, enhancing its robustness to appearance changes. These results convincingly demonstrate the superiority of our proposed Unity-based feature extractor in improving de-anonymization performance.

3) Time Overhead: The time overhead comprises three components: the feature extractor pre-training time, classifier training time, and real-time inference time. On a server equipped with 2 GPUs (GeForce RTX 2080 Ti), running Ubuntu 18.04.6 LTS OS, powered by an Intel Xeon E5-2678 v3 CPU, and with 128 GB RAM, the time overheads for these components are 8.05 hours, 2.3 seconds, and 123.5 ms respectively. It is noteworthy that, as the training procedures are performed prior to the attack,

TABLE I
PERFORMANCE UNDER VARIOUS GALLERY SIZE

Number of samples in gallery	100	200	**300**	400	500	600	700	800	900
Closed/open-world ASR (%)	78.8/63	91.5/64	**92.1/66.9**	93.5/65.8	94.6/64	96.3/68.5	94/62.7	93.5/65	98/66

TABLE II
PERFORMANCE WHEN CHANGING CAMERA COMBINATION

View	F	B	R	L	FB	FR	RL	BR	BL	FL	FBR	FRL	BRL	FBL	**FBRL**
Open-world ASR (%)	72.9	83.4	80.4	78.9	90.1	91.9	93.3	90.6	90.4	89.3	93.4	88.4	91.3	92.1	**92.1**
Closed-world ASR (%)	51.3	54.3	60.0	58.7	62.3	68.6	69.0	69.6	67.3	67.0	64.3	66.4	66.6	64.7	**66.9**

the real-time inference time of 123.5 ms is deemed acceptable for real-world scenarios.

C. Impact of Various Factors on AvatarHunter

In this subsection, we evaluate the impact of various factors on the performance of AvatarHunter.

1) Gallery Size: In this study, the gallery functions as the "training dataset" for the classifier. The size of the "training samples" significantly impacts the final inference performance. We explore the effect of gallery size by varying the proportion of trials included in the gallery. The findings, summarized in Table I, reveal that in the closed-world setting, the ASR escalates from 78.8% to 98.0% as the number of samples in the gallery increases from 100 to 900. In the open-world setting, the ASR is 63.0% and 66.0% for gallery sizes of 100 and 900 samples, respectively. Notably, beyond a gallery size of 300 samples, further increases in gallery size yield minimal performance gains. Setting the gallery size at 300 samples, given that each trial captures only 1 s of the victim's behavior, constructing a gallery with 30 trials per victim is feasible, thereby imposing a minimal burden on the adversary.

2) The Number of Cameras: As mentioned in Section IV-B, to enhance de-anonymization performance, AvatarHunter utilizes four cameras positioned at the front (F), left (L), right (R), and behind (B) of the victim. The robustness of AvatarHunter with different camera combinations is evaluated. As indicated in Table II, the ASR increases in the open-world avatar setting with the addition of more cameras. Moreover, different angles contribute to varying ASRs. For example, the performance using front and behind cameras (FB) is lower than that using front and right cameras (FR). AvatarHunter attains the highest ASR of 93.3% with the right and left camera combination. Nevertheless, effective performance is achievable even with a single camera (e.g., an ASR of 72.9% in the closed-world avatar setting using only the front camera). In conclusion, employing multiple cameras enhances AvatarHunter's performance, which remains robust even in single-camera scenarios.

3) Length of Recording Video: In Section V-B, for each view of a given trial, 30 frames (*i.e.*, 1 s) are selected as the input for AvatarHunter. We assess the impact of frame length on AvatarHunter's performance. As presented in Fig. 13, we adjust the frame length from 1 to 128, in increments of 1. It is noted that with the increase in frame length, the ASRs for both open-world and closed-world avatar settings incrementally rise.

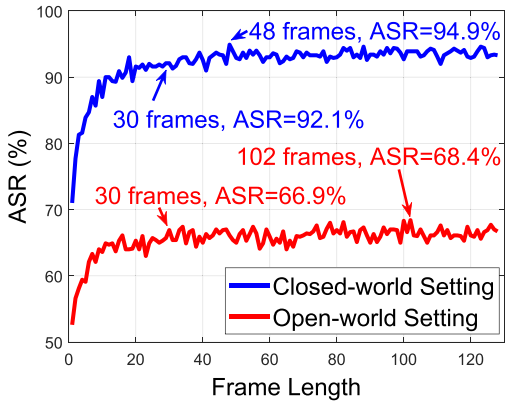


Fig. 13. Performance under various frame lengths.

Notably, even with just 10 frames (0.33 seconds), the ASRs in closed-world and open-world scenarios exceed 87% and 63% respectively, significantly outperforming the random guess (*i.e.*, an ASR of 10%). The optimal performance for AvatarHunter in the closed-world setting reaches 94.9% with a frame length of 48, and in the open-world setting, it achieves 68.4% with a frame length of 102. These experimental results demonstrate that AvatarHunter is effective even with short video lengths.

D. Performance in Full-Body Tracking Setting

Following the evaluation of various factors affecting AvatarHunter, we investigate its performance in full-body tracking scenarios, as detailed in Section III-B-3. Given that current commercial devices like Quest do not support full-body sensor connectivity, we conduct a simulation based on the public dataset CMU-Mocap. We follow the same setup as described in Section V-A, where 10 avatars are utilized. Considering most of data from CMU-Mocap are utilized to train the feature extractor as described in Section IV-C, we select ten users (i.e., user #2, #5, #6, #10, #37, #43, #45, #46, #49, #55 in CMU-Mocap) not involved in feature extractor training for the simulation. We generate the dataset, called FBT-dataset which also contains 100 user-avatar pairs. The feature extractor remains unchanged from the commercial device setting, with only the gallery being reconstructed. As depicted in Fig. 14, in the open-world setting, the ASRs for AvatarHunter, GaitSet, and GaitBase are 78%, 43%, and 49%, respectively. Notably, the ASR differs from that in the VRC-dataset due to the different volunteers and the more

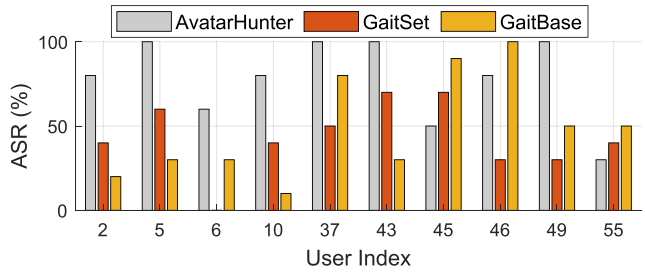


Fig. 14. Performance of AvatarHunter and benchmarks under full-body tracking and open-world setting.

variable walking patterns in CMU-Mocap, as it is not specifically designed for gait recognition. Nonetheless, AvatarHunter outperforms benchmarks, underscoring the effectiveness and robustness of our proposed feature extraction solution.

VI. Countermeasures

AvatarHunter reveals a severe privacy threat for the stakeholders of VR-related areas. In this section, we discuss several methods that can defend against AvatarHunter.

A. Intuitive Countermeasures

We first introduce some intuitive countermeasures that can defend against the proposed AvatarHunter.

Detecting Suspicious Users in VR: For AvatarHunter, recording the victim avatar's behaviors is crucial during the attack phase. Therefore, detecting suspicious users who consistently focus on a specific user or select an avatar with a void appearance can mitigate privacy leakage risks. By identifying and excluding these suspicious users, the likelihood of victim privacy breaches can be significantly reduced.

Restricted access control: Another straightforward solutions is enhancing the access control of the VR application. For the user who requires high demands of anonymity, they can only share their VR scenarios with the users they trust. However, this countermeasure only works when users can control their VR applications (e.g., the user is the administrator of the chatting room of VRChat).

B. Obfuscation-Based Defense Mechanism

In this subsection, we present an obfuscation-based mechanism to defend against AvatarHunter. We first introduce its principle, followed by implementation and simulation results.

Principle of obfuscation: As described in Section IV, the key reason why AvatarHunter can identify the victim's identity by analyzing the collected video clips V is that AvatarHunter can extract the feature F corresponding to the victim's gait characteristics. According to Section II-B, since the avatar movement of the victim is correlated with the motion data captured by the sensors of the VR device, intentionally introducing noise into the sensor data during the avatar movement generation process causes the videos to inaccurately reflect the victim's real-world gait characteristics. Consequently, the extracted F is no longer

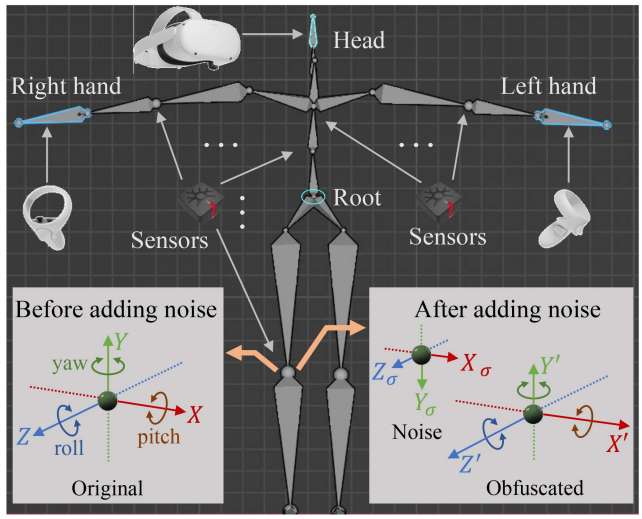


Fig. 15. The principle of adding noise to avatar's skeleton.

accurate. Thus, for the same classifier Ψ, the probability that its predicted result $J = \Psi(F)$ matches the real identity will decrease, thereby mitigating the risk of de-anonymization attacks.

Implementation: During the avatar rendering phase, the skeleton's nodes are activated by sensors attached to the user's body. We utilize sensor data from CMU-Mocap, consistent with the setup in Section V-D. Typically, VR sensors have six degrees of freedom (6-DOF), as shown in Fig. 15, with each node capturing rotation data for pitch, roll, and yaw, corresponding to the sensor's X, Y, and Z axes. For a given time t and the ith node n_i of the skeleton, the measured data are represented as $(x_i(t), y_i(t), z_i(t))$. We add noise to this data, resulting in obfuscated rotation angles $(x_i'(t), y_i'(t), z_i'(t))$ for the rendering phase. The approach for adding noise is as follows:

$$x_i'(t) = x_i(t) + n_x, \tag{8}$$

$$y_i'(t) = y_i(t) + n_y, \tag{9}$$

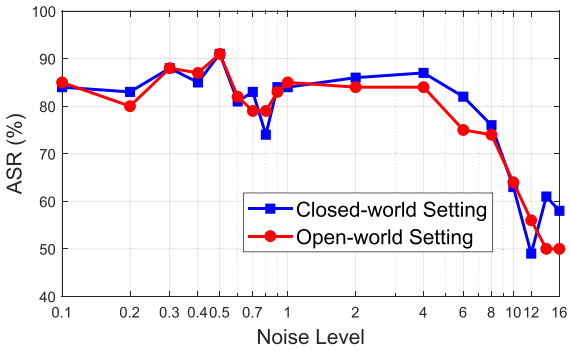
$$z_i'(t) = z_i(t) + n_z, \tag{10}$$

where n_x, n_y, and n_z follow a Gaussian distribution with the probability density function $f(x)$ given by:

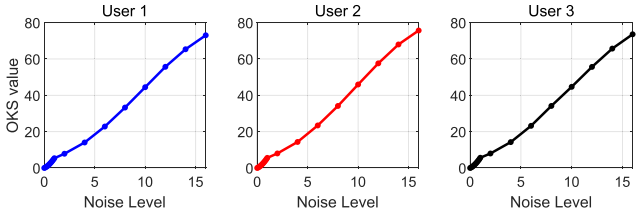
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{11}$$

with mean μ determining the distribution's location and standard deviation σ its magnitude. In this study, we set $\mu = 0$ and vary σ. Note that, during the noise adding phase, we do not add noise to the head and left, right hands in order to reduce the impact of noise on the user's experience.

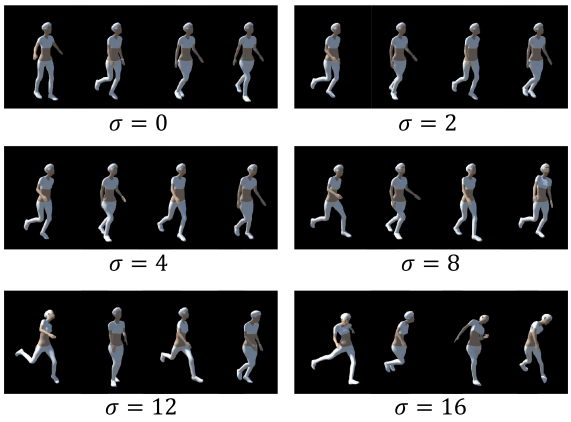
Performance analysis: We present a simulation to evaluate the efficacy of this countermeasure. We choose the full-body tracking scenario as described in Section V-D, where the avatar movements are driven and video clips are collected in the Unity platform. During the simulation, the noise level σ varies from 0.1 to 16. Fig. 16(a) presents the ASR of AvatarHunter under various noise levels. Notably, in the absence of noise, the ASRs for AvatarHunter in closed-world and open-world settings are



(a) ASRs under different noise levels.



(b) OKS values under various noise levels.



(c) Video clips under various .

Fig. 16. Simulation results about defense mechanism.

85% and 78%, respectively. Interestingly, the introduction of low-level noise does not decrease the ASR; rather, it increases. This phenomenon occurs because minor noise fails to mislead AvatarHunter and may even accentuate the gait differences between users. However, when the noise level is sufficiently high (i.e., greater than 6), the ASR decreases significantly. Therefore, the obfuscation-based mechanism is demonstrated to be an effective countermeasure against AvatarHunter.

To measure the fluctuations in avatar movement caused by adding noise, we choose the object keypoint similarity (OKS) metric, which is used in the field of pose detection to calculate the similarity between ground truth and the algorithm's predicted human keypoints [55], [56]. In this paper, OKS is used to calculate the degree of deviation of the noise-added motion compared to the original motion. Fig. 16(b) presents the OKS of three users. Fig. 16(c) shows the frames for video clips under various σ levels, with the time interval between two consecutive frames being 0.25 seconds. It is observed that as σ increases, the fluctuation becomes more pronounced. Notably, when σ is less than 12, the fluctuations in avatar movements are not significant. Thus, in practice, it is essential for users to choose an appropriate noise level to achieve a balance between privacy preservation and user experience.

VII. Limitations and Future Works

Currently, AvatarHunter is designed for conducting de-anonymization attack based on the victim avatar's gait information. To apply AvatarHunter to more broad VR scenarios, the following limitations need to be addressed.

A. Limited Size and Diversity of Dataset

Enlarging dataset including avatars with non-humanoid appearance: In this study, only humanoid avatars as shown in Fig. 10(a) are studied. However, for the avatars with the appearance of non-humanoid appearance, since their skeletons are different from that of humans, it is hard for the feature extractor to extract human movement signatures. Involving the animal-like avatars into the Unity-based feature extraction module, or proposing more advanced techniques are potential solutions and we leave them for future works.

Enlarging more users: Although AvatarHunter achieves an ASR of 92.1% in the closed-world setting, which is significantly higher than random guessing (i.e., 10%), the number of user-avatar pairs is still limited. In this study, only 100 user-avatar pairs were collected in each scenario when conducting experiments on VRChat and full-body tracking settings. Additionally, in Section IV-C, data from 29 users in the CMU-MoCap dataset and 69 avatars with different body shapes were selected to fine-tune the feature extractor. To further demonstrate AvatarHunter's efficiency, more volunteers will be recruited and included in future versions of our dataset.

B. Requirements for Gallery

In this study, AvatarHunter builds a gallery to store the victim's avatar gait information in VR scenarios, as acquiring prior knowledge about the target user is an essential prerequisite for user identification. In this subsection, we discuss several concerns regarding the gallery.

Victims not in the gallery: It is not always feasible to collect the victim's video clips before launching attacks. In such cases, AvatarHunter will fail to extract gait information from the target victim in VR scenarios. Extracting gait features from the physical world is a potential solution, which we will explore in future work.

Impact of background scenes: There is no requirement for the adversary to obtain video clips with the same background scene as those in the gallery. AvatarHunter is designed to focus on the silhouette of the avatar, rather than the avatar's colors or background scenes. As described in Section IV-B-1, AvatarHunter employs a background matting scheme to eliminate the background of each frame in the collected video clips. We acknowledge that the performance of background elimination is not perfect (i.e., sometimes there are artifacts in the silhouette). However, the attack performance detailed in

Section V demonstrates the feasibility of AvatarHunter without relying on background scene information.

Other action types: Currently, AvatarHunter mainly focuses on the action of *walking* for two reasons. First, walking is a common and frequent action when users play VR applications, making it easy for adversaries to obtain such footage. Second, compared with other movement features, gait information extracted from walking behavior is relatively stable, and extensive research on gait-based user identification exists, which can serve as the foundation of AvatarHunter. Our previous work [43] considers other types of actions, including waving, picking up objects, and throwing objects. However, luring victims into performing these actions is not always feasible in real-world attack scenarios.

C. VR Device Types

Although the gait video clips used to train the feature extractor are generated based on CMU-MoCap, the features we extract are from the frames of gait video clips, not from the control points data in the human body. Therefore, regarding input data modality, the difference in the number of control points does not affect the results. However, in terms of result accuracy, more control points allow the video clips to retain more accurate movement features from the human, making de-anonymization easier. This is demonstrated by Figs. 12(a) and 14, where AvatarHunter achieves a higher ASR in the full-body tracking setting (78%) than in the VRChat setting (66.9%). For other VR device types such as HTC Vive Pro 2 and Valve Index, AvatarHunter should also work since they have the same working principles as the Quest used in our evaluation. This will be studied in future work.

VIII. Conclusion

In this study, we have introduced AvatarHunter, a non-intrusive and user-unconscious de-anonymization attack in VR environments. AvatarHunter discreetly gathers gait videos of victims within VR applications and employs a Unity-based feature extractor to identify the victim's movement signature, which remains unaffected by changes in avatar appearance. We implemented AvatarHunter in a real-world VR application, and the experimental outcomes demonstrate that AvatarHunter can successfully deduce the identity behind the avatar in both closed-world and open-world settings. The efficacy of AvatarHunter is shown to be resilient against various factors, including the length of video clips, combinations of camera angles, and the extent of prior knowledge about the target victim. Additionally, we propose an obfuscation-based defense mechanism and validate its effectiveness through simulation, offering a means to safeguard against AvatarHunter.

References

[1] Y. Meng, Y. Zhan, J. Li, S. Du, H. Zhu, and X. S. Shen, "De-anonymization attacks on metaverse," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[2] D. Liu, C. Huang, L. Xue, W. Zhuang, X. Shen, and B. Ying, "Collaborative and verifiable VNF management for metaverse with efficient modular designs," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 3, pp. 616–628, Mar. 2024.

[3] G. V. Research, "Virtual reality (VR) market size, share & trends analysis report, 2023–2030," 2023. [Online]. Available: https://www.researchandmarkets.com/reports/4312515/virtual-reality-vr-market-size-share-and-trends

[4] Y. Wang et al., "A survey on metaverse: Fundamentals, security, and privacy," 2022, *arXiv:2203.02662*.

[5] V. Inc., "Vrchat," 2021. [Online]. Available: https://hello.vrchat.com/

[6] P. Casey, I. Baggili, and A. Yarramreddy, "Immersive virtual reality attacks and the human joystick," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 550–562, Mar./Apr. 2021.

[7] W.-J. Tseng et al., "The dark side of perceptual manipulations in virtual reality," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2022, pp. 612:1–612:15.

[8] S. Luo, A. Nguyen, C. Song, F. Lin, W. Xu, and Z. Yan, "Oculock: Exploring human visual system for authentication in virtual reality head-mounted display," in *Proc. 27th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2020.

[9] R. Miller, N. K. Banerjee, and S. Banerjee, "Using siamese neural networks to perform cross-system behavioral authentication in virtual reality," in *Proc. IEEE Virtual Reality 3D User Interfaces*, 2021, pp. 140–149.

[10] R. Trimananda, H. Le, H. Cui, J. T. Ho, A. Shuba, and A. Markopoulou, "OVRseen: Auditing network traffic and privacy policies in oculus VR," in *Proc. 31st USENIX Secur. Symp.*, Aug. 2022 pp. 3789–3806, [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/trimananda

[11] C. Shi et al., "Face-Mic: Inferring live speech and speaker identity via subtle facial dynamics captured by AR/VR motion sensors," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, 2021, Art. no. 478490.

[12] D. Edward, "6 best meeting & collaboration tools in VR for oculus quest 2," 2021. [Online]. Available: https://allvirtualreality.com/review/best-meeting-collaboration-tools-vr-quest.html

[13] M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson, "Personal identifiability of user tracking data during observation of 360-degree VR video," *Sci. Rep.*, vol. 10, no. 1, pp. 1–10, 2020.

[14] 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 *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 110:1–110:12.

[15] T. Mustafa, R. Matovu, A. Serwadda, and N. Muirhead, "Unsure how to authenticate on your VR headset? Come on, use your head!," in *Proc. 4th ACM Int. Workshop Secur. Privacy Analytics*, 2018, pp. 23–30.

[16] C. George, M. Khamis, D. Buschek, and H. Hussmann, "Investigating the third dimension for authentication in immersive virtual reality and in the real world," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 277–285.

[17] D. Fu et al., "Unsupervised pre-training for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14750–14759.

[18] T. He, X. Jin, X. Shen, J. Huang, Z. Chen, and X.-S. Hua, "Dense interaction learning for video-based person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1490–1501.

[19] H. Chao, K. Wang, Y. He, J. Zhang, and J. Feng, "GaitSet: Cross-view gait recognition through utilizing gait as a deep set," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3467–3478, Jul. 2022.

[20] B. Sinha, "Unity: The biggest platform for creating VR content," 2017. [Online]. Available: https://digital.hbs.edu/platform-digit/submission/unity-the-biggest-platform-for-creating-vr-content/

[21] VRChat, "Thanks to our community for making 2018 VRChat's best year yet!," 2019. [Online]. Available: https://twitter.com/VRChat/status/1086389685268635648

[22] C. G. Lab, "Motion capture database," 2021. [Online]. Available: http://mocap.cs.cmu.edu

[23] L. Facebook Technologies, "Oculus quest 2," 2022. [Online]. Available: https://www.oculus.com/quest-2/

[24] A. Inc., "Apple vision pro - apple," 2024. [Online]. Available: https://www.apple.com/apple-vision-pro/

[25] H. Corporation, "Vive - VR headsets, games, and metaverse life," 2024. [Online]. Available: https://www.vive.com/us/

[26] P. I. Pte. ltd., "PICO virtual reality| official website | PICO global," 2023. [Online]. Available: https://www.picoxr.com/global

[27] IDC, "IDC - AR & VR headsets market share," 2024. [Online]. Available: https://www.idc.com/promo/arvr

[28] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch, "The effect of avatar realism in immersive social virtual realities," in *Proc. 23rd ACM Symp. Virtual Reality Softw. Technol.*, 2017, pp. 39:1–39:10.

[29] D. Takahashi, "59% of VR developers use unity, but devs make more money with unreal," 2017. [Online]. Available: https://uploadvr.com/vr-developers-unity-unreal/
[30] U. Technologies, "Getting started with VR development in unity," 2022. [Online]. Available: https://docs.unity3d.com/Manual/VROverview.html
[31] C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian, "An approach for user identification for head-mounted displays," in *Proc. ACM Int. Symp. Wearable Comput.*, 2015, pp. 143–146.
[32] Y. Shen et al., "GaitLock: Protect virtual and augmented reality headsets using gait," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 3, pp. 484–497, May/Jun. 2019.
[33] F. Mathis, J. H. Williamson, K. Vaniea, and M. Khamis, "Fast and secure authentication in virtual reality using coordinated 3D manipulation and pointing," *ACM Trans. Comput.- Hum. Interact.*, vol. 28, no. 1, pp. 6:1–6:44, 2021.
[34] S. Eberz, K. B. Rasmussen, V. Lenders, and I. Martinovic, "Evaluating behavioral biometrics for continuous authentication: Challenges and metrics," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 386–399.
[35] A. Kupin, B. Moeller, Y. Jiang, N. K. Banerjee, and S. Banerjee, "Task-driven biometric authentication of users in virtual reality (VR) environments," in *Proc. MultiMedia Model. 25th Int. Conf.*, 2019, pp. 55–67.
[36] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1961–1974, Sep. 2015.
[37] A. Ajit, N. K. Banerjee, and S. Banerjee, "Combining pairwise feature matches from device trajectories for biometric authentication in virtual reality environments," in *Proc. IEEE Int. Conf. Artif. Intell. Virtual Reality*, 2019, pp. 9–16.
[38] J. Liebers et al., "Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 517:1–517:11.
[39] V. Nair et al., "Unique identification of 50,000 virtual reality users from head & hand motion data," in *Proc. 32nd USENIX Secur. Symp.*, Anaheim, CA: USENIX Association, 2023, pp. 895–910.
[40] C. for Biometrics and S. Research, "Gait databases," 2022. [Online]. Available: http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp
[41] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1116–1124.
[42] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 17–35.
[43] B. Falk, Y. Meng, Y. Zhan, and H. Zhu, "POSTER: ReAvatar: Virtual reality de-anonymization attack through correlating movement signatures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 2405–2407.
[44] A. A. Arafat, Z. Guo, and A. Awad, "VR-Spy: A side-channel attack on virtual key-logging in VR headsets," in *Proc. IEEE Virtual Reality 3D User Interfaces*, 2021, pp. 564–572.
[45] M. Kotaru and S. Katti, "Position tracking for virtual reality using commodity WiFi," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2671–2681.
[46] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.
[47] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Real-time high-resolution background matting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8762–8771.
[48] L. van der Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[49] M. Community, "Makehuman," 2022. [Online]. Available: https://github.com/makehumancommunity/makehuman
[50] BLENDER., "Blender," 2022, [Online]. Available: https://www.blender.org/
[51] S. Mihcin et al., "Investigation of wearable motion capture system towards biomechanical modelling," in *Proc. IEEE Int. Symp. Med. Meas. Appl.*, 2019, pp. 1–5.
[52] H. Tanalol, H. Hashim, P. Turumogan, N. A. M. Noor, A. Baharum, and F. D. Deris, "Identifying gamified teaching elements in computer science course," in *Proc. Int. Conf. Platform Technol. Serv. (PlatCon)*, 2023, pp. 18–23.
[53] C. Fan, J. Liang, C. Shen, S. Hou, Y. Huang, and S. Yu, "Opengait: Revisiting gait recognition towards better practicality," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9707–9716.
[54] C. Fan et al., "GaitPart: Temporal part-based model for gait recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14225–14233.
[55] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7091–7100.
[56] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, and H. Hu, "Human pose as compositional tokens," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 660–671.

Yan Meng (Member, IEEE) received the BS degree in electronic and information engineering from the Huazhong University of Science and Technology, in 2016 and the PhD degree in computer science and technology from Shanghai Jiao Tong University, in 2021. He is a research assistant professor in Shanghai Jiao Tong University, China. His research interests include wireless network security and IoT security. He published more than 30 papers, including IEEE TDSC, IEEE TMC, IEEE WCM, IEEE INFOCOM, ACM CCS, and USENIX Security. He received the 2022 ACM China Doctoral Dissertation Award.

Yuxia Zhan (Student Member, IEEE) received the BS degree in computer science and technology from Shanghai Jiao Tong University, in 2021. She is currently working toward the master's degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include virtual reality security and privacy enhancement technology.

Jiachun Li (Student Member, IEEE) received the BS degree in communication engineering from the Huazhong University of Science and Technology, in 2020. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include smart home security and smart healthcare security.

Suguo Du received the PhD degree from the School of Mathematical and Information Sciences, Coventry University, U.K., in 2002. She is currently as professor with the Department of Management Science, Shanghai Jiao Tong University, China. Her current research interests include risk and reliability assessment, vehicular networks security and privacy protection, and social networks security management. Her research has been supported by the National Science Foundation of China.

Haojin Zhu (Fellow, IEEE) received the BSc degree from Wuhan University, China, in 2002, the MSc degree from Shanghai Jiao Tong University, China, in 2005, both in computer science and the PhD degree in electrical and computer engineering from the University of Waterloo (Canada), in 2009. He is currently a professor in the computer science department, 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, and ACM MOBICOM. He received a number of awards including 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, in 2014, Top 100 Most Cited Chinese Papers Published in International Journals, in 2014, Distinguished Member of the IEEE INFOCOM Technical Program Committee, in 2015, 2020, best paper awards of IEEE ICC, in 2007 and Chinacom, in 2008, WASA Best Paper Runner-up Award, in 2017. He is serving as the editorial board for *IEEE Transactions on Wireless Communications* and program committees for top conferences such as USENIX Security, ACM CCS, NDSS, and IEEE INFOCOM.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a university professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks. He is a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign member, and a Distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award, in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario), in 2019, James Evans Avant Garde Award, in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award, in 2015 and education award, in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee, in 2019 and AHSN Technical Committee, in 2013. He has also received the Excellent Graduate Supervision Award, in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA), in 2003 from the Province of Ontario, Canada. He served as the technical program committee chair/co-chair for IEEE GLOBECOM'16, IEEE INFOCOM'14, IEEE VTC'10 Fall, IEEE GLOBECOM'07, and the chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the past president of the IEEE ComSoc. He was the vice president for Technical & Educational Activities, vice president for Publications, member-at-large on the Board of Governors, chair of the Distinguished Lecturer Selection Committee, and member of IEEE Fellow Selection Committee of the ComSoc. He served as the editor-in-chief of the IEEE IoT JOURNAL, IEEE Network, and IET Communications.