Modeling and Defense of Social Virtual Reality Attacks Inducing Cybersickness

Samaikya Valluripally, Aniket Gulhane, Khaza Anuarul Hoque, *Senior Member, IEEE*, and Prasad Calyam, *Senior Member, IEEE*

Abstract—Social Virtual Reality Learning Environments (VRLE) offer a new medium for flexible and immersive learning environments with geo-distributed users. Ensuring user safety in VRLE application domains such as education, flight simulations, military training is of utmost importance. Specifically, there is a need to study the impact of "immersion attacks" (e.g., chaperone attack, occlusion) and other types of attacks/faults (e.g., unauthorized access, network congestion) that may cause user safety issues (i.e., inducing of *cybersickness*). In this article, we present a novel framework to quantify the security, privacy issues triggered via immersion attacks and other types of attacks/faults. By using a real-world social VRLE viz., vSocial and creating a novel attack-fault tree model, we show that such attacks can induce undesirable levels of cybersickness. Next, we convert these attack-fault trees into stochastic timed automata (STA) representations to perform statistical model checking for a given attacker profile. Using this model checking approach, we determine the most vulnerable threat scenarios that can trigger high occurrence cases of cybersickness for VRLE users. Lastly, we show the effectiveness of our attack-fault tree modeling by incorporating suitable design principles such as *hardening*, *diversity*, *redundancy* and *principle of least privilege* to ensure user safety in a VRLE session.

Index Terms—Security and privacy, user safety, cybersickness, virtual reality learning environments, attack-fault trees, statistical model checking, risk assessment, design principles

1 Introduction

Social Virtual Reality (VR) applications transfer real-world contexts into simulations as part of interactive learning in a multi-user scenario [1]. Novel application areas adopt virtual reality learning environments (VRLEs) for special education, surgical training, and flight simulations purposes. The advancement in VR technology enhances user-system interactions and user-experience by processing and visualizing of the collected activity data from multiple wearable devices (i.e., VR headsets) and geographically distributed users. Social VR benefits from the convergence of VR technology, smart devices and cloud platforms, in order to integrate real-world smart objects with virtual world objects (user avatars) within virtual environments. However, this flexibility although enables seamless VRLE interactions, it can cause serious security, privacy issues [2] that disrupt user safety by inducing cybersickness [3].

Recent works [4], [5], [6] highlight the importance of security and privacy (SP) issues in VR applications. However, they lack in evaluation of critical SP, system/network fault issues that impact VRLE applications' functionality and user experience. For instance, an intruder can gain unauthorized access (e.g., using fake credentials) to tamper the VRLE content which constitutes as a security breach. The unauthorized access can also lead to a privacy breach through eavesdropping during a VRLE session. Similarly, an intentional network fault can be triggered by an adversary by launching a Denial of Service (DoS) attack. As a result of such an intentional cyber-attack which we also define as a fault-attack, the VRLE content can be rendered unavailable to VRLE users. In addition, a faulty VRLE component (e.g., infrared sensor, gyroscope) can be used for distorting the view of a VRLE user during a session. The impact of such security and privacy breaches can induce *cybersickness*, which is a set of unpleasant symptoms such as eyestrain, headache, nausea or even vomiting, thus compromising user safety in a VR session [3], [7], [8].

To motivate the impact of security and privacy issues on user safety, we consider a VRLE application viz., vSocial [1] shown in Fig. 1 that is designed for youth with autism spectrum disorder (ASD). This multi-modal VRLE system is hosted on High Fidelity [9], a social VR platform that is deployed over high-speed networks. High Fidelity can be used to render 3D visualizations based on the dynamic human computer interactions with an edge cloud node i.e., vSocial Cloud Server. Owing to the inherent inter-connectivity of the network-edge and the core cloud in the VRLE setup, the VR application is vulnerable to novel attacks known as *immersion attacks*. To elucidate, using SP issues as the vulnerabilities, an attacker can: (i) cause defacement of VRLE content with offensive images known as *overlay attack* [5], (ii) obstruct the user view or trigger noise attenuation during VRLE sessions known as *occlusion attack* [5], and (iii) create application issues i.e., reduction of graphical content or delays between both user and avatar movement.

- *The authors are with the Electrical Engineering and Computer Science Department, University of Missouri-Columbia, Columbia, MO 65201 USA. E-mail: {svbqb, arggm8}@mail.missouri.edu, {hoquek, calyamp}@missouri.edu.*

Manuscript received 7 Dec. 2020; revised 6 Oct. 2021; accepted 13 Oct. 2021. Date of publication 19 Oct. 2021; date of current version 11 Nov. 2022.
This work was supported by the National Science Foundation under Grants CNS-1647213, CNS-1659134, and CNS-2114035.
(Corresponding author: Prasad Calyam.)
Digital Object Identifier no. 10.1109/TDSC.2021.3121216

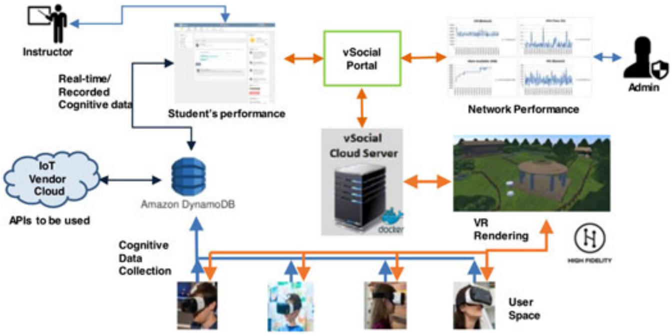


Fig. 1. vSocial system components used for real-time student learning session management.

Failure to address such security, privacy and resulting safety (SPS) issues in VRLE results in alteration of instructional content, compromise of learning outcomes, abuse of access privileges leading to confidential student information disclosure and/or poor student engagement due to cybersickness. Thereby, we formulate our problem focus to be on delivering VRLE content in a high-performance and safe manner in a user VRLE session, even in the cases where SPS issues arise. More specifically, our goal is to model the inter-relationship between security, privacy and user safety for identification of cybersickness events and for creation of relevant defense mechanisms to increase the resilience of social VRLE systems.

In this paper, we present a novel framework to quantify the security, privacy issues triggered via immersion attacks and other types of attacks/faults which affect user experience in terms of inducing cybersickness during use of heads-up displays in social VRLEs. We model the security, privacy issues using attack-fault trees (AFTs) by building upon our prior results in [2] where we utilized attack tree formalism in the context of security and privacy issues. In contrast, this work goes beyond the concept of simple cyber-attacks in VRLEs and models immersion attacks and fault-attacks to explore how they can be used to cause security/privacy breaches. To the best of our knowledge, ours is the first work that explores the impact of immersion attacks on user safety in VRLE environments. We use the AFTs [10] to model the temporal dependencies for a given attacker profile. The temporal dependencies relate to the cyber-attacks/fault-attacks on VRLE components and their impact on cybersickness factors obtained via experimental behavioral analysis. For quantitative evaluation of the developed AFTs, we convert them to stochastic timed automata (STA) [11] formalism, and then use formal verification techniques, to be specific statistical model checking (SMC) [12]. The use of AFTs enables us to derive graphical models that provide a systematic representation of various cyber-attack/fault-attack scenarios and their respective consequences towards a common system disruption goal. The utilization of SMC allows us to reason about the dynamic user-system interactions in VRLEs [13].

Our main contributions summary is as follows:

- We quantitatively measure cybersickness with respect to security and privacy issues using our proposed framework as a consequence of cyber attacks, faults and their combinations.
- We perform a trade-off analysis by evaluating the severity of different types of security/privacy attacks and their impact on the cybersickness in the vSocial application [1]. From the related results, we find that the causes of Denial of Service attack and data leakage to be the most dominant candidates to induce high levels of cybersickness in a VRLE session.
- We then use the trade-off analysis and perform a risk assessment of the identified threat vectors' impact on cybersickness levels in a social VRLE. In addition, we evaluate the cybersickness in a networked VRLE setup by identifying the most critical system components in cyber attack/fault scenarios.
- We demonstrate the effectiveness of using design principles (also known as security principles) i.e., *hardening, diversity, redundancy and principle of least privilege* for enhancing the security and privacy of VRLE sessions in the event of severe threats.

Paper Organization. The remainder of the paper is organized as follows: Section 2 discusses related works. Section 3 introduces the necessary background and terminology. Section 4 discusses the proposed security and privacy framework in detail. Section 5 presents the quantitative results using our proposed framework on the VRLE case study. Section 6 discusses the effectiveness of recommended design principles on the security and privacy threat scenarios impacting cybersickness. Section 7 presents details on the benefits and limitations of using our proposed work in the design of VRLEs. Section 8 concludes the paper.

2 Related Works

2.1 Security and Privacy Risks in VR Environments

2.1.1 Network-Based Attacks

There have been several prior studies that highlight the importance of security and privacy threats on Augmented Reality (AR) devices, and edge computing. A recent study [14] on challenges in AR and VR discusses the threat vectors for educational initiatives, however this study does not characterize related attack impacts. Survey articles such as [4], [6], [15], [16], [17] are significant for understanding the concepts of threat taxonomy and attack surface area of sensors and fog computing applications. They highlight the need to go beyond specific components such as network, hardware or user interface, and propose end-to-end solutions that consider system and data vulnerabilities [5]. An observation from the above state-of-art is that - there is a dearth of scholarly works on the quantitative evaluation for security and privacy threats in the context of VR applications.

2.1.2 Immersion Attacks

Existing works [5] discuss about the immersion attacks that can cause physical harm and disrupt the user experience in a virtual environment [5]. For instance, authors in [5] present a proof-of-concept for a *disorientation attack* by changing the translation and yaw. They also show creation of a *physical collision attack* by tampering the SteamVR [18] chaperone file (also referred to as a chaperone-attack) [5], and demonstrate how an *overlay attack* can display unintended images during a session. The work in these studies [5], [19] are exemplar sources of attack possibilities in order to study unique cyber attack patterns relevant to VRLEs that can potentially induce cybersickness for users. In addition, the works in [14], [20], [21], [22] discuss the security, privacy challenges, ethical issues in VR applications. However, they do not study the impact of such security, privacy challenges on the VRLE user safety. Furthermore, the authors in the work [23] discuss about how the participants can get disoriented due to the exposure in a virtual environment even for a 20 minutes session. We build our work based on these prior works [5], and explore potential novel attack surfaces related to security, privacy, safety issues in VRLE applications and their associated impact on inducing user cybersickness.

2.2 Cybersickness in VR Environments

2.2.1 *Potential Factors Inducing Cybersickness*

Several works [24], [25] define cybersickness as a form of motion/simulation sickness due to several physiological factors of the user in correlation with immersion and presence in a VRLE. For instance, the work in [7], [8] detail that the sensory conflict between the vestibular and visual senses causes motion sickness in an unfamiliar virtual environment. Another work [26], attributes cybersickness as a form of disorientation due to the user's view point. In contrast, the work in [27] notes the difference between motion sickness and cybersickness in terms of the individual causes and their associated impacts on the users using motion sickness susceptibility questionnaire (MSSQ) via experimental simulation. In addition, the work in [28] found that that graphical quality in virtual environments significantly affects immersion and induces cybersickness. Moreover, VRLE application issues such as jitter, position tracking error are also considered as cybersickness factors in prior works [24], [25]. Our work builds on these existing works, but uniquely identifies potential factors that induce cybersickness and thus impact user safety within social VRLE application sessions.

2.2.2 *Measuring Cybersickness in VR Environments*

Several works [27], [29] devise questionnaires related to cybersickness in relation to motion sickness factors. The work in [29] adapts the simulator sickness questionnaire (SSQ) to quantify the physiological effects. Their survey methodology considers factors related to general discomfort, fatigue, eyestrain, difficulty focusing, headache, fullness of head, blurred vision, dizziness with eyes closed, and vertigo. Effects such as cybersickness, simulator sickness, and motion sickness have been found to be correlated in [27]. Similarly, the work in [30] explores the subjective aspects of the virtual environments i.e., presence, engagement, immersion and their relation to cybersickness symptoms (i.e., consequences on user experience). In addition, there have been studies that examined the effect on user immersion by implementing such questionnaires within a virtual environment [31]. The authors in [31] found that users in the virtual environments prefer such deployment of virtual questionnaires over traditional paper survey formats. However, we remark that this finding is dependent on the integration of the questionnaires such that the immersion is preserved during the virtual sessions for the users. We adapt these works in devising our questionnaires that are seamlessly integrated in the VRLE sessions to help us determine the potential factors of cybersickness.

3 Background and Terminology

3.1 Social VRLE Application Case Study: vSocial

Our Social VRLE (i.e., vSocial) consists of virtual objects (a.k.a. avatars) and creates shared VR environments using High Fidelity to enable synchronous interaction amongst participants who are geographically distributed at remote locations. Our vSocial application is used for delivering Social Competence Intervention (SCI) curriculum as shown in Fig. 1 that consists of modules such as: VR content rendering, web applications and classroom portal with instructional content hosted as web pages. The VRLE setup consists of consumer wearables, and base stations for accurate localization and tracking of controllers [1]. The consumer wearables in vSocial include: VR headsets (e.g., Oculus Rift [32] and the HTC Vive [33]) equipped with VR hand-held controllers, EEG headsets (e.g., Muse) on the client-side. The cloud server in vSocial architecture shown in Fig. 1 delivers the VRLE content to the users in these virtual classrooms and stores the user engagement information and activities in real-time. Our social VR platform benefits the rendering capability of the relevant learning content into a VRLE session on a 'web entity' using web technologies. The web entity allows instructors to create, modify, present and share slides via web pages hosted on the Slides.com web service. It also enables student course enrollment and progress management via social network packages such as HumHub [34]. These VRLE sessions are orchestrated on a cloud server using web applications that control the virtual scenes content (e.g., curriculum delivered as game activities) in a multi-user scenario. Remote participants in our system use multiple head-mounted display devices such as HTC Vive and Oculus Rift, however their actions are co-ordinated within e.g., games or learning tasks across the multiple user locations. The vSocial server provides functionalities such as user access control, session content management, and student progress tracking. Given that the vSocial server is a critical system component, it is an attractive target for an attacker.

3.2 Attacks in Social VRLE Application Use Case

To understand the potential cyber-attacks/fault-attacks and their impact on cybersickness, we perform an experimental behavior analysis on a vSocial instance. For this, we simulate exemplar security, privacy and safety attacks based on our prior work [2] through our experimental validations. We validate the SP attacks of vSocial application using simulation tools (such as Clumsy 0.2 [35] and Wireshark [36]). For the purposes of this paper, we define: (a) *security* – as a condition that ensures a VR system to perform critical functions with the establishment of confidentiality, integrity, and availability [37], (b) *privacy* – as a property that regulates the IoT data collection, protection, and secrecy in interactive systems [37], (c) *safety* – as the disruption in the system that compromises the user's overall well-being [37].

3.2.1 *Security Concerns in Social VRLEs*

Social VRLEs use distributed head-mounted displays and wearable devices when connecting to virtual classrooms.

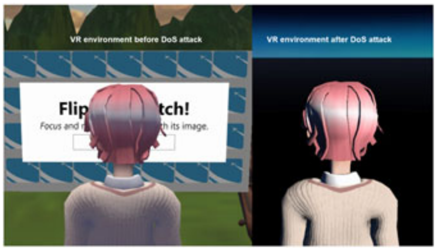
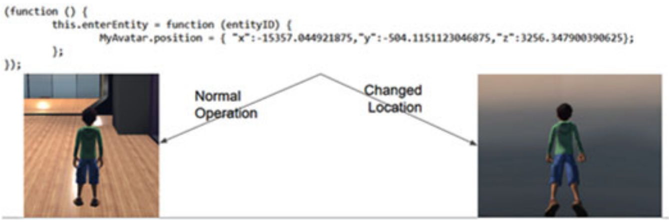

VR environment before DoS attack
VR environment after DoS attack
Flip tch!
Focus and h its image.

Fig. 2. A before and after scenario showcasing the effect of DoS attack on vSocial causing a server crash.



```
(function () {
        this.enterEntity = function (entityID) {
                MyAvatar.position = { "x":-15357.044921875,"y":-504.1151123046875,"z":3256.347900390625};
        };
});
```

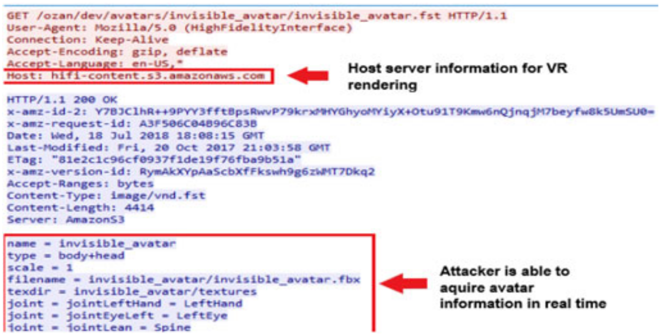
Fig. 3. vSocial system content affected due to insertion of malicious scripts in the student learning environment.

Consequently, user experience in social VRLE applications is highly sensitive to Distributed Denial of Service (DDoS) attacks. To elucidate, a malicious user simulates a DoS attack scenario on the vSocial environment, via *packet tampering*, *packet duplication*, and *packet drop* which results in a VRLE server crash as shown in Fig. 2. Based on our validation experiments, a packet drop of 80% disrupts the communication between the user and VRLE server thereby impacting the learning experience [2]. Similarly, a tamper rate of 20% is sufficient to crash the VRLE server for all the connected VRLE users as shown in Fig. 2. In case of packet tampering, a man-in-the-middle attack scenario can reveal confidential information as discussed in [2].

Another security attack scenario can involve an attacker gaining unauthorized access by impersonating as a valid user. *Gaining unauthorized access* to the instructor account can trigger a privacy attack with threats such as: *disclosure of confidential user information*, and *tampering* of the learning content, edge computing and network devices. For instance, an intruder in social VRLE inserts malicious scripts to initiate data tampering of the VRLE application content as shown in Fig. 3. With this, the attacker can modify the boundaries of the users' virtual view that can lead to a user running into a wall and getting physically hurt. *Failure to address such security attack scenarios can lead to an immersive attack i.e., overlay attack [38]) that can impact the VRLE user experience.*

3.2.2 Privacy Concerns in Social VRLEs

A user privacy breach can involve an intruder who enters into a VRLE world by gaining unauthorized access (e.g., using fake credentials) and tries to eavesdrop the virtual classroom conversations. This scenario can trigger a novel immersion attack known as *man-in-the-room* attack specific to VRLEs. Moreover, the attacker can gain access to the virtual location of the user and can disrupt the orientation of the user's virtual object (avatar). Privacy attacks can also involve *packet tampering* that was demonstrated in [2], where an attacker performs an illegal packet capture to extract sensitive information (*packet sniffing attack* as shown in Fig. 4).



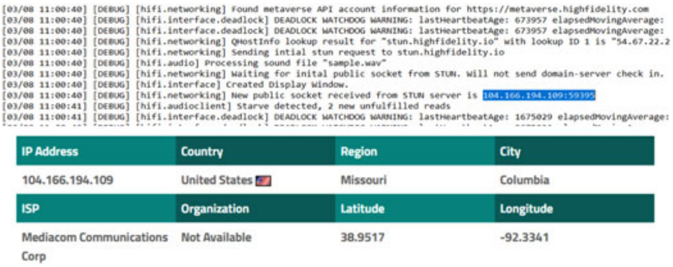
```
GET /ozan/dev/avatars/invisible_avatar/invisible_avatar.fst HTTP/1.1
User-Agent: Mozilla/5.0 (HighFidelityInterface)
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: hifi-content.s3.amazonaws.com

HTTP/1.1 200 OK
x-amz-id-2: Y7BJClhR++9PYY3fftBpsRwvP79krxM0YGhyoMYiyX+Otu91T9Kmw6nQjnqjM7beyfw8k5UmSU0=
x-amz-request-id: A3F506C04B96C83B
Date: Wed, 18 Jul 2018 18:08:15 GMT
Last-Modified: Fri, 20 Oct 2017 21:03:58 GMT
ETag: "81e2c1c96cf0937f1de19f76fba9b51a"
x-amz-version-id: RymAkXYpAaScbXfFkswh9g6zWMT7Dkq2
Accept-Ranges: bytes
Content-Type: image/vnd.fst
Content-Length: 4414
Server: AmazonS3

name = invisible_avatar
type = body+head
scale = 1
filename = invisible_avatar/invisible_avatar.fbx
texdir = invisible_avatar/textures
joint = jointLeftHand = LeftHand
joint = jointEyeLeft = LeftEye
joint = jointLean = Spine
```

Host server information for VR rendering
Attacker is able to aquire avatar information in real time

Fig. 4. Packet Sniffing attack to disclose avatar and host server information.



```
[03/08 11:00:40] [DEBUG] [hifi.networking] Found metaverse API account information for https://metaverse.highfidelity.com
[03/08 11:00:40] [DEBUG] [hifi.interface.deadlock] DEADLOCK WATCHDOG WARNING: lastHeartbeatAge: 673957 elapsedMovingAverage:
[03/08 11:00:40] [DEBUG] [hifi.interface.deadlock] DEADLOCK WATCHDOG WARNING: lastHeartbeatAge: 673957 elapsedMovingAverage:
[03/08 11:00:40] [DEBUG] [hifi.networking] QHostInfo lookup result for "stun.highfidelity.io" with lookup ID 1 is "54.67.22.2
[03/08 11:00:40] [DEBUG] [hifi.networking] Sending intial stun request to stun.highfidelity.io
[03/08 11:00:40] [DEBUG] [hifi.audio] Processing sound file "sample.wav"
[03/08 11:00:40] [DEBUG] [hifi.networking] Waiting for inital public socket from STUN. Will not send domain-server check in.
[03/08 11:00:40] [DEBUG] [hifi.interface] Created Display Window.
[03/08 11:00:40] [DEBUG] [hifi.networking] New public socket received from STUN server is 104.166.194.109:59395
[03/08 11:00:41] [DEBUG] [hifi.audioclient] Starve detected, 2 new unfulfilled reads
[03/08 11:00:41] [DEBUG] [hifi.interface.deadlock] DEADLOCK WATCHDOG WARNING: lastHeartbeatAge: 1675029 elapsedMovingAverage:
```

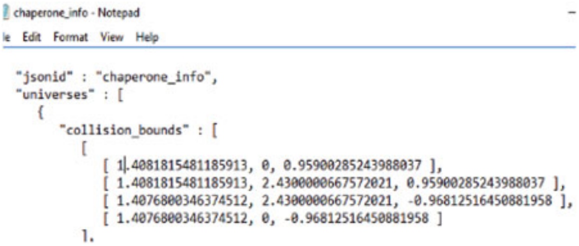
IP Address	Country	Region	City
104.166.194.109	United States 🇺🇸	Missouri	Columbia
ISP	**Organization**	**Latitude**	**Longitude**
Mediacom Communications Corp	Not Available	38.9517	-92.3341

Fig. 5. A malicious user exposes the IP address of a valid VRLE user (disclosing user physical location) by gaining access to VRLE activity logs.

Another form of privacy breach can occur when the attacker *discloses confidential user information* via packet sniffing attack to gain access to users' physical location information and credentials as shown in Fig. 5. Such privacy attacks can create: (a) loss of confidentiality (LoC) when sensitive information is disclosed, and (b) loss of integrity (LoI) when the attacker tampers with the VRLE content. *Failure to address such privacy attack scenarios can disrupt the User Immersive Experience (UIX) in an ongoing VRLE session by obstructing the view of the users in their learning sessions i.e., occlusion attack, or by creating a noise attenuation issue or by causing disorientation of the content.*

3.2.3 Safety Concerns in Social VRLEs

Based on the works in [5] and [19], we perform safety attacks that can potentially disrupt the UIX in a vSocial instance. For instance, we assume that a *XSS browser attack* can occur, where the web entities in vSocial are targeted to hook the browser for hijacking the VRLE content. Immersion attacks e.g., *overlay attack* overlays and replaces visuals with offensive content on a user's local machine which can partially compromise the VRLE [5]. Performing an overlay attack along with an SQL injection can create more impact on user privacy as the confidential information can be captured via overlay entities. By modifying the boundaries in a SteamVR chaperone file as shown in Fig. 6 via TFTP [18], an attacker can disorient the user avatar or can lead a user to run into walls or physical objects. With this immersion attack, a VRLE user can experience light-headedness, a potential cybersickness factor. In addition, *a network fault-attack* that is triggered by low bandwidth events (e.g., *packet loss scenarios*) can disrupt the VRLE content, thereby inducing



```
"jsonid" : "chaperone_info",
"universes" : [
   {
      "collision_bounds" : [
         [
            [ 1.4081815481185913, 0, 0.95900285243988037 ],
            [ 1.4081815481185913, 2.4300000667572021, 0.95900285243988037 ],
            [ 1.4076800346374512, 2.4300000667572021, -0.96812516450881958 ],
            [ 1.4076800346374512, 0, -0.96812516450881958 ]
         ],
```

Fig. 6. Immersion attack to tamper the chaperone file.

cybersickness for a user who is remotely participating in a VR session.

Using our experimental validations, we next describe a preliminary study on the outlined security, privacy threat scenarios and their respective impact on the cybersickness. Based on the above experiments, our problem focus is to deliver learning content seamlessly and safely in a user VRLE session, even during the event of a security or a privacy breach. Towards this aim, we perform a systematic study to identify the potential cybersickness factors that rely on both the simulator sickness as well as user experience factors adapted from relevant prior works [3], [7]. DSN-2

3.2.4 *Impact Analysis of SP Factors on Cybersickness*

In this section, we perform an experimental behavior analysis of the above simulated SP attack scenarios and their impact on potential cybersickness factors in VRLEs. For this, we adopt several cybersickness factors based on the existing works [28], [29], [30] that include e.g., the Virtual Sickness Questionnaire (VSQ). To measure the impact on each of these cybersickness factors during the VRLE users session, we organize the cybersickness factors under 6 questions as shown in Table 1. These questions are integrated as virtual questionnaires (VQs) and post-session survey for feedback collection such that they do not cause any disruption to the VRLE users [31]. In addition, based on our prior work in [2]: (i) we simulate a *security attack, privacy attack and combination of security, privacy attacks* across three different user activities as shown in Fig. 7 and, (ii) subsequently quantify cybersickness using a set of VQs. In this context, we set up a vSocial [1] instance with activities that rely on fine movement, visual clarity, and clear audio, all of which are disrupted by SP attacks as shown in Fig. 7.

First, we collect baseline data (i.e., benign behavior) using VQs at the end of activity1. Next, during activity 2, we simulate malicious packet drops to disrupt VRLE content rendering as a security attack and then collect user feedback. Similarly, in activity 3 we simulate a privacy attack to trace the user's virtual location and then play a distracting noise to disrupt the user experience. This disruption is stopped approximately halfway through activity 3, where the user response is recorded. For the rest of activity 3, a combination of security and privacy attacks (i.e., packet tampering + disclosure of user location), are simulated after which the user exits the vSocial environment to give feedback via a post-session paper survey.

TABLE 1
Survey Questions Asked During a Cybersickness Experiment

Label	Cybersickness Questions	Related Works
Nausea	How often do you feel nauseaous ?	[28], [29]
Discomfort	What level do you feel discomfort or disoriented during the sessions ?	[29]
Vection	How often do you have the feeling of self-movement ?	[28], [29]
Eyestrain	What level do you feel eye strain or light-headed ?	[29]
Task Completion	How likely are you able to finish the tasks in the session ?	[28], [29]
Mental Engagement	How mentally engaged are you in the environment ?	[29], [30]

To maintain uniformity across the collected feedback, we used the same VQ as shown in Fig. 8. For this survey, 15 VRLE user participants provided their feedback for each of the questions in our cybersickness survey on a scale of 1 to 5, where 1 (very poor), 2 (poor), 3 (Moderate), 4 (good), 5 (excellent). This collected data allows us to quantify the aggregated impact value of cybersickness score due to SP attacks in the VRLE. We stop the attack simulations until the checkpoint (attack duration is 180 seconds), to avoid further discomfort in relation to the time spent by the user in vSocial.

Results of the Impact Analysis of SP Attacks on Cybersickness Factors. Using the collected data for the SP versus cybersickness factors analysis, we outline the data points for each cybersickness factor versus the average rating given by the users as shown in Fig. 8. The cybersickness factors listed in Table 1 are represented as N,D,V,E,T,M on x-axis and the average of the user responses on y-axis as shown in Fig. 8. Based on the results, we observe that security, privacy, and combination of SP attacks have significant impact the cybersickness factors. To elucidate, the combination of security, privacy attacks causes significant dizziness and nausea, the primary factors that induce cybersickness. From these results, we understand that SP attacks do certainly induce cybersickness by disrupting users and also the functionality of the VRLE. With this analysis, we further explore the potential security and privacy, fault-attack in detail (single and combination of threats) that could induce cybersickness

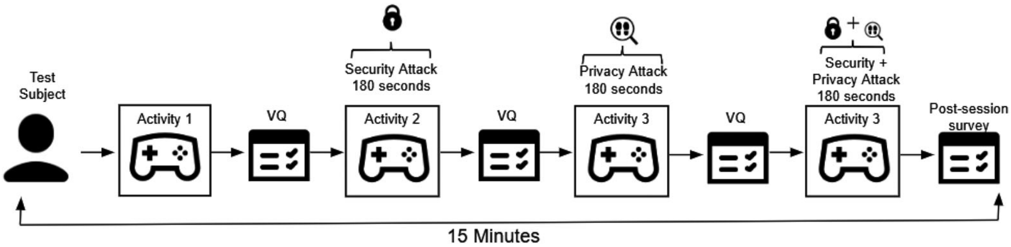

Test Subject
Activity 1
VQ
Security Attack 180 seconds
Activity 2
VQ
Privacy Attack 180 seconds
Activity 3
VQ
Security + Privacy Attack 180 seconds
Activity 3
Post-session survey
15 Minutes

Fig. 7. Activities and their associated virtual questionnaires used for the immersion survey experiment.

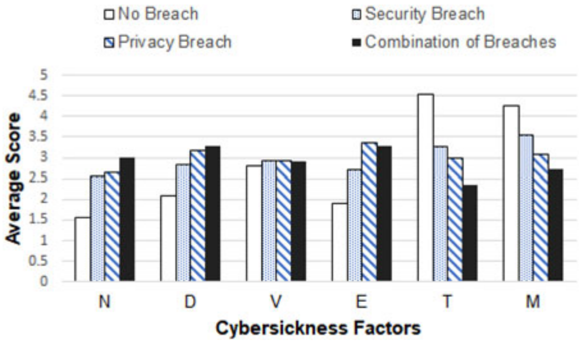


Fig. 8. Results of cybersickness-survey experiments.

in a VRLE using attack-fault trees (AFTs). Moreover, we build the AFTs using: (a) the experimental validations in Section 3.1, and (b) the above qualitative impact results relating inducing cybersickness shown in Fig. 8. Each of these collected impact data on cybersickness factor results are mapped to a numerical scale of 1(Low)-to-5(High) and are further utilized in the quantitative analysis of AFTs detailed in Section 5.

3.3 Statistical Model Checking

Statistical model checking (SMC) is a variation of the well-known classical model checking [39] approach for a system that exhibits stochastic behavior. The SMC approach to solve the model checking problem involves simulating (Monte Carlo simulation) the system for finitely many runs, and using hypothesis testing to infer whether the samples provide a statistical evidence for the satisfaction or violation of the specification [40].

Stochastic Timed Automata. Stochastic timed automata (STA) is an extended version of timed automata (TA) with stochastic semantics. A STA associates logical locations with continuous, generally distributed sojourn times [41]. In STA, constraints on edges and invariants on locations, such as clocks are used to enable transition from one state to another [10].

Definition 1 (Stochastic timed automata). *Given a timed automata which is equipped with assignment of invariants $\mathcal{I}$ to locations $\mathcal{L}$, we formulate an STA as a tuple $T = \langle\, \mathcal{L}, l_{init}, \Sigma, \mathcal{X}, \mathcal{E}, \mathcal{I}, \mu \,\rangle$, where $\mathcal{L}$ is a finite set of locations, $l_{init} \in \mathcal{L}$ is the initial location, Σ is a finite set of actions, $\mathcal{X}$ is the finite set of clocks, $\mathcal{E} \subseteq \mathcal{L} \times \mathcal{L}_{clk} \times \Sigma \times 2^{\mathcal{X}}$ is a finite set of edges, with $\mathcal{L}_{clk}$ representing the set of clock constraints, $\mathcal{I}$: $\mathcal{L} \longrightarrow \lambda$ is the invariant where λ is the rate of exponential assigned to the locations $\mathcal{L}$, μ is the probability density function (μ_l) at a location $l \in \mathcal{L}$.*

An exemplar STA is shown in Fig. 9 that consists of the locations {Initial, Wait, Fail}. Herein, the *Initial* location represents the start of execution of an STA and a *clock* x is used to keep track of the global time. The communication in an STA exists between its components using message broadcast signals in a bottom-up approach. The STA is activated by broadcasting *initiate!* signal, which transitions to wait location and waits for the *fail* signal. In an STA, time delays are governed as probability distributions (used as invariants) over the locations. The Network of Stochastic Timed Automata (NSTA) is defined by composing all component automaton to obtain a complete stochastic system satisfying the general compositionality criterion of TA transition rules [12], [41].

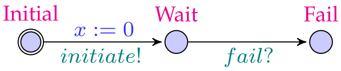


Fig. 9. An exemplar STA.

UPPAAL SMC. UPPAAL SMC is an integrated tool for modeling, validation, and verification of real-time systems modeled as a network of stochastic timed automata (NSTA) extended with integer variable, invariant, and channel synchronizations [43]. The UPPAAL SMC model checker tool is embedded with a simulator to mainly check the behavior of the NSTA and a query engine to visualize the probability distributions and number of runs with time bounds, and also compute expected values. In SMC, the probability estimate is derived using an estimation algorithm as well as by using statistical parameters, such as $1 - \alpha$ (required confidence interval) and ϵ (error bound) [44]. For instance, if we indicate goal state in the STA of *Top_event* as *Fail*, then the probability of a successful occurrence within time t can be written as: $Pr[x <= t](<> Top\_event.Fail)$ where, $<>$ represents the existential operator () and x is a clock in the STA to track the global time.

3.4 Design Principles

To build a trustworthy VRLE system architecture which ensures security and privacy, integration of design principles in the life cycle of edge computing interconnected and distributed wearable device based systems is essential [42]. We adapt the following three design principles from NIST SP800-160 [37], [42] such as: (i) *Hardening* – defined as reinforcement of individual or types of components to ensure that they are harder to compromise or impair, (ii) *Diversity* – defined as the implementation of a feature with diverse types of components to restrict the threat impact from proliferating further into the system, (iii) *Principle of least privilege* – defined as limiting the privileges of any entity, that is just enough to perform its functions and prevents the effect of threat from propagating beyond the affected component and (iv) *Redundancy* – defined as deployment of redundant components, such that the normal functionality of a system is retained when a system component is compromised or impaired.

4 Security and Privacy Framework

In this section, we present our proposed framework to identify the most vulnerable attacks/fault components inducing cybersickness by performing security, privacy and safety analysis of social VRLE applications. An overview of the approach followed in our framework is shown in Fig. 10. First, we model the SP attack vectors inducing cybersickness by using an attack-fault tree (AFT) formalism based on our preliminary results from Section 3.2.4. Second, each AFT is translated into an equivalent STA to form an NSTA, as input into the UPPAAL SMC tool. Third, we use the quantitative assessment from the tool to determine if the probability of disruption for cybersickness occurrence is higher than a set threshold determined as part of VRLE design requirements. Lastly, we perform risk assessment of the identified

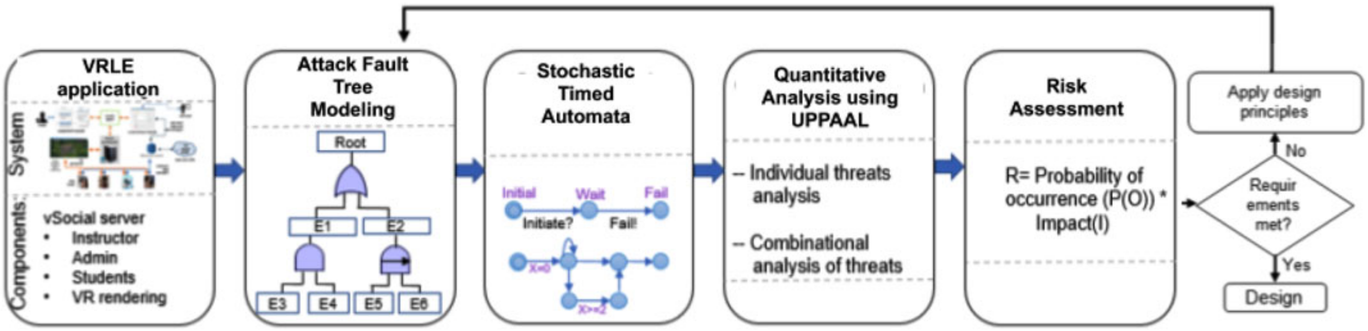


Fig. 10. Proposed framework for security and privacy analysis of a social VRLE.

attack vectors to determine the potential impact on VR application functionality and impact in terms of inducing cybersickness. Based on this determination, we subsequently prescribe the design principles such as: *hardening*, *diversity*, *redundancy* and *principle of least privilege* that can be adopted in VRLE deployments. Moreover, each of these design principles serve as mitigation strategies that rely on NIST guidelines and further measure the impact on cybersickness occurrence levels for the best performing mitigation strategy. Thereby, our problem scope in this case reduces to identifying the cybersickness occurrence in the event of anomaly and to provide seamless multi-user interaction within the VRLE sessions. Overall, our framework steps help in the investigation of potential cyber attacks and fault-attacks. Further, they help in recommendations of VRLE application design alternatives based on design principles.

4.1 Formalization of Security and Privacy Attack-Fault Trees

Attack-fault trees (AFTs) are hierarchical models that show how an attacker goal (root node) can be refined into smaller sub-goals (child/intermediate nodes) via gates until no further refinement is possible such that the basic attack steps (BAS) are reached. BAS represents the *leaf nodes* of an AFT [43]. To explore dependencies on VRLE attack surfaces, AFTs enable sharing of subtrees. Hence, AFTs are often considered as directed acyclic graphs, rather than trees [10].

Using AFTs, our proposed framework can capture the relationship and the impact of both attacks and faults [10]. Traditional threat models such as Common Vulnerability Score System (CVSS) [44] or STRIDE [45] can identify the vulnerabilities of a system in a systematic manner. However, these traditional threat models lack the ability to perform a quantitative analysis on multi-modal inter-dependencies of identified threats. Existing works [46], [47], and [48] provide the steps that include identifying risks and analysis of both safety and security threats. However, these prior works focus on application-specific models to identify risk levels.

As an alternative, there have been efforts to use attack trees (ATs) [49] as well as fault trees (FTs) [50] which are similar in terms of how basic attack steps (BAS) and basic component failure (BCF), respectively. They are modeled using the leaf nodes of the tree, and show propagation of attack events via gates through the system [10]. However, both ATs and FTs vary based on the type of goal (security attack for ATs and fault occurrence for FTs) along with their associated analysis [51], [52]. Moreover, such approaches are not feasible due to their: (a) lack of capability in modeling the propagation of failure rates (accidental or malicious), (b) lack of support of shared sub-trees, and (c) failure to consider the quantification of impact of disruption for different attacker profiles. These ATs and FTs when considered separately cannot show a cause and effect analysis between security and safety issues in VRLE. For instance, in our previous works [2], [53], we utilized ATs for modeling attack scenarios in VRLEs, and observed similar limitations i.e., ATs do not support modeling the inter-dependencies between fault and attacks.

In order to capture the multi-stage and the dynamic temporal inter-dependencies of the causal security, privacy and safety, our approach is to model both the security and safety aspects of VRLE into a combined formalism known as attack-fault tree (i.e., AFTs) modeling [10]. AFTs encompass both ATs and FTs to support all the above syntactic constructs and the leaf behavior from attack trees [49], [52] and dynamic fault trees [50], [51]. Fault-attacks in VRLEs include two categories: the first category includes faults that can directly impact the user safety (*cybersickness*); similarly, the second category includes faults that cause security and/or privacy issues that can compromise user safety. Each of the fault tree gates adopted in these AFTs provide a classical formalism for reliability engineering that is heavily used in numerous industry domains [50], [51].

Definition 2 (Attack-Fault Trees (AFTs)). *An attack-fault tree A is defined as a tuple* $\{N, Child, Top\_event, l\} \cup \{AFT$ *elements*$\}$ *where, N is a finite set of nodes in the attack-fault tree; Child:* $N \rightarrow N^*$ *maps each set of nodes to its child nodes;* Top_event *is an unique goal node of the attacker where* $Top\_event \in N$; *l: is a set of labels for each node* $n \in N$; *and AFT elements: is a set of elements in an attack-fault tree A.*

Definition 3 (AFT elements). *AFT elements aid in generating the attack-fault tree and are defined as a set of* $\{G \cup BE \cup IE\}$ *where,* $G = \{OR, AND, SAND, SOR, PAND\}$ *represents the set of* gates *used for modeling the multi-threat and fault-attack scenarios in AFTs, and BE is the set of basic events* $\{BAS \cup BCF\}$ *and IE is the set of intermediate events.*

Attack-Fault Tree Elements. Attack-fault tree elements aid in generating an attack-fault tree and are defined as a set of $\{G \cup L\}$ where, G represents gates; L represents leaf nodes. Following are the descriptions of each of the AFT elements.

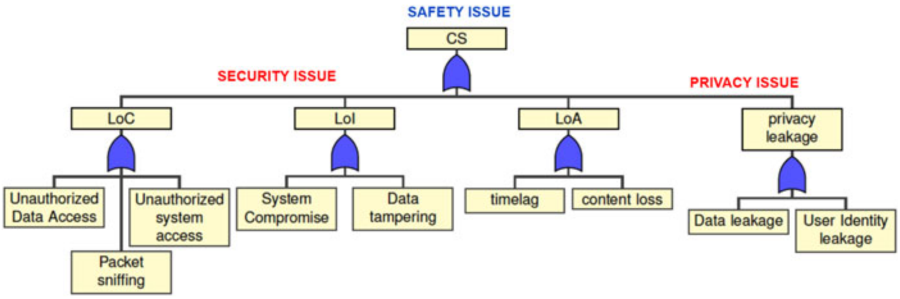
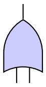
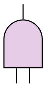
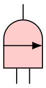
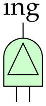


Fig. 11. Formalized safety attack-fault tree (AFT) with threat scenarios inducing cybersickness (CS).

Attack-Fault Tree Gates. Given an attack-fault tree A, we formally define the attack-fault tree gates $G = \{OR, AND, SAND, PAND\}$.

-  **OR:-** An OR gate is disrupted if either of its child nodes are disrupted.
-  **AND:-** An AND gate is disrupted when all its child nodes are disrupted.
-  **SAND:-** A Sequential AND (SAND) gate is disrupted in the order of left to right only when its leftmost child node is disrupted. To elucidate, the gate is disrupted using the condition: the success of previous step determines the success of the upcoming child node.
-  **PAND:-** A Parallel AND (PAND) gate model the order dependent disruption (i.e., left to right) but activates its child nodes all at once.

We limit our attack-fault tree modeling to these gates, however attack-fault trees [10] can adopt any other gates from the static/dynamic fault trees. The output nodes of the gates G in an attack-fault tree A are defined as *Intermediate nodes* (I), which will be located at a level that is greater than the leaf nodes.

Basic Attack Step (BAS). A BAS collectively represents all the individual atomic steps within a composite attack-fault scenario. Each of the BAS also represents the *leaf nodes* of an AFT [43].

Attack-Fault Tree Leaves. Given an attack-fault tree A, we formally define the *attack-fault tree leaves* $L_{\text{node}} = \{BAS \cup leaf\ nodes\}$.

In other words, L_{node} is the terminal node with no other child node(s) which is either modeled as BAS or a simple leaf node (modeled with exponential distribution) of the AFT. To elucidate, for an attacker to impersonate as a valid user, the prospective BAS can include: (i) spoofing attack, and (ii) session hijacking to the system depending on the attacker profile. For an attack-fault tree A, we assumed the attack duration to have an exponential rate and model the equation as : $P(t) = 1 - e^{\lambda t}$ where, λ is the rate of exponential distribution [2]. We use the exponential distribution because of its tractability and ease of handling, since they are defined by a single parameter.

4.2 Threat Modelling Using Attack-Fault Trees

Based on the results discussed in Section 3 and experimental evidence from our prior work [2], we model potential VRLE security, privacy and fault-attack scenarios that induce cybersickness in the form of an AFT as shown in Fig. 11. As part of our framework approach, we consider cybersickness as the primary goal for an attacker. We model our main AFT as shown in Fig. 11 using different security, privacy, fault-attacks in the form of sub-trees as shown in Figs. 12, 13, and 14. Exploring the security aspect in CIA triad of {Confidentiality, Integrity, Availability} results in an enormous number of leaf nodes in the AFT. For the purposes of our work, we term the main generated AFT that is covering SPS attack scenarios inducing cybersickness as *Safety-AFT*.

To elucidate, the safety AFT contains root node as cybersickness which branches out to intermediate nodes such as security issues relevant to CIA triad i.e., {Loss of Confidentiality (LoC), Loss of Integrity (LoI), Loss of Availability (LoA)}. This branching results in an enormous number of leaf nodes as shown in Figs. 12 and 13. The intermediate nodes serve the purpose of establishing the relationship of root node to leaf nodes which are basically the basic attack steps to disrupt cybersickness. We continue the branching of intermediate nodes until no further division is possible, i.e., a case that terminates as leaf nodes. Similarly, the other sub-tree whose root node is privacy leakage address the privacy threat scenarios that induce cybersickness for a social VRLE user. Our safety AFT has a goal node (i.e., Security/Privacy) which can be compromised by an attacker via potential attack steps shown as child nodes (i.e., intermediate and leaf nodes).

We remark that our work advances knowledge in linking the cybersickness factors that can be potentially triggered due to an SP attack in a VRLE system. We consider significant technical and SP issues in the cybersickness experiments, and both sets of issues relate to the same technical constraints (e.g., network bandwidth), and application impairments (e.g., flickering). For instance, we model technical issues such as low bandwidth and network fault scenarios in a VRLE as factors that can induce cybersickness as shown in our safety AFT. In our generated safety-AFT, we also explore the temporal dependencies in terms of sharing subtrees where the cause and effect relationship is outlined. To elucidate, an intermediate node in a sub-tree can be used

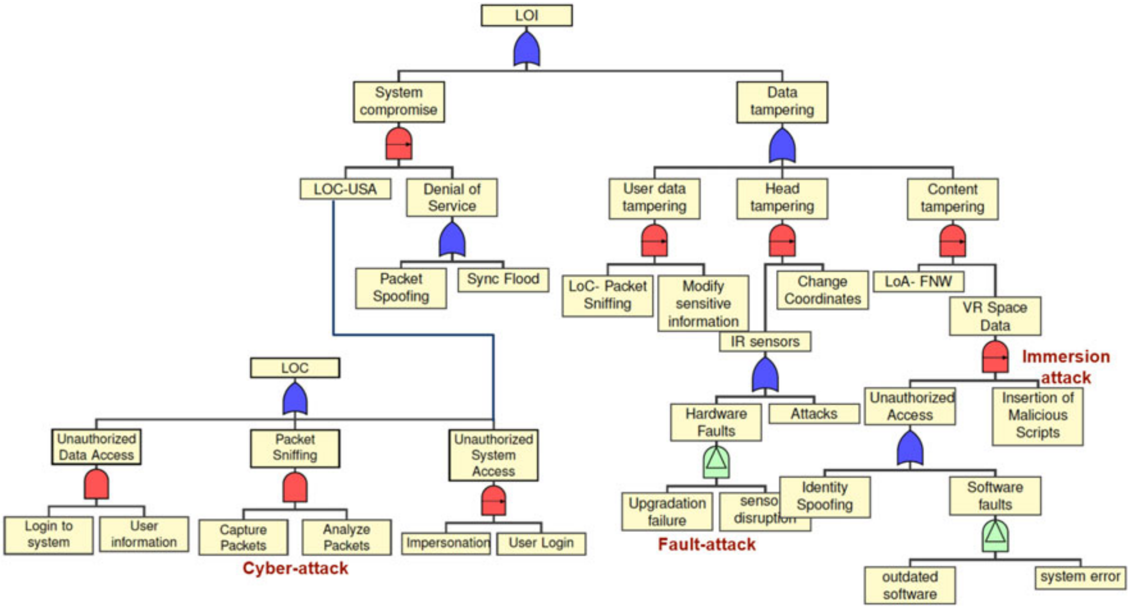


Fig. 12. Formalized attack-fault sub-tree with threat scenarios triggering a Loss of Confidentiality (LoC) and/or Loss of Integrity (LoI) issues.

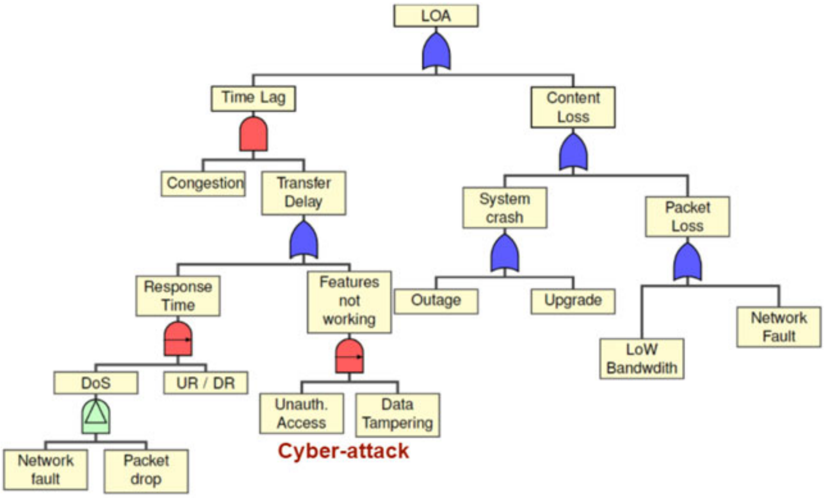


Fig. 13. Formalized attack-fault sub-tree with threat scenarios triggering a Loss of Availability (LoA) issue.

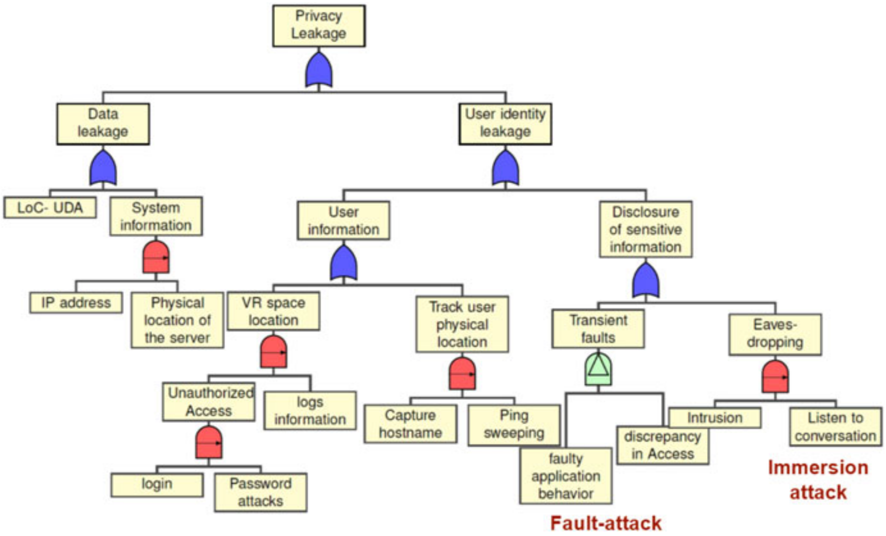


Fig. 14. Formalized attack-fault sub-tree with threat scenarios triggering a privacy leakage issue.

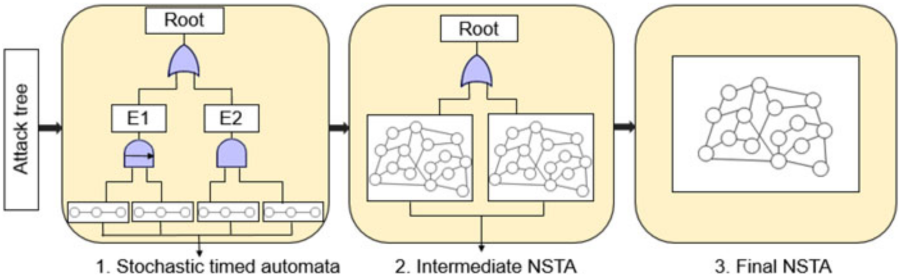


Fig. 15. Framework for translation of AFTs into network of stochastic timed automata (NSTA).

as leaf node in a different sub-tree which can act as a sharing node. For example, the unauthorized system access (USA) which is an intermediate node in the LoC subtree is used as a leaf node represented as LoC-USA in the LoI subtree as shown in the Fig. 12. Similarly, LoA subtree shares its intermediate node "Features not working (FNW)" as a leaf node which is represented as LoA-FNW in the LoI subtree. Moreover, the LoC subtree intermediate node i.e., Unauthorized data access (UDA) acts as the leaf node LoC-UDA in the privacy leakage subtree as shown in Fig. 14. Using the experimental validations detailed in Section 3.2, we show each of the intermediate nodes via potential attack-fault steps triggering cybersickness to build more realistic and practical AFTs. We further use these experimental results to classify each of the SP issues under Loss of Confidentiality (LoC), Loss of Integrity (LoI) and Loss of Availability (LoA). These categories are considered as intermediate nodes and are modeled based on a cause-effect relationship using AFTs. Using this generated safety AFT, we discuss the process of converting an AFT into STA to perform stochastic model checking in the next section.

4.3 Translation of AFTs into Stochastic Timed Automata (STA)

In this section, we generate STA for each of the sub-trees in Figs. 12, 13 and 14 that are part of the safety-AFT in Fig. 11. An overview of our translation approach is shown in Fig. 15 where: (i) each of the leaf nodes in these AFTs are converted into individual STAs. The intermediate events, which are basically the output of the logic gates that are used at different levels are converted imperatively into STA; (ii) the generated STAs are composed in parallel by including the root node; (iii) the obtained NSTA is then used for statistical model checking in order to verify the security, privacy and fault-attack properties formalized as SMC queries. As mentioned earlier, these obtained STAs are used for performing model checking to verify the cybersickness metrics formalized as SMC queries.

To demonstrate the translation of an AFT into an STA, we first consider the safety AFT shown in Fig. 11. As part of the translation, each of the security AFT element (leaf and gates) input signals are connected to the output signal of child nodes. The generated network of STAs (i.e., NSTA) communicates using $\{initiate, fail\}$ signals. *initiate* - indicates activation signal of AFT element. This signal is sent initially from the root node to its children. *fail* - indicates disruption of that attack-fault tree element. This signal is sent to the parent node from its child node to indicate an STA disruption. The scope of the above signals also includes special symbols such as: i)'?' (e.g., initiate?) means that the event will wait for the reception of the intended signal, ii) '!'(e.g., initiate!) implies output signal broadcasts to other STA in the AFT.

Illustrative Example. In the subsequent paragraph, we illustrate our translational approach by converting the safety AFT into an exemplar NSTA. For instance, we show the conversion of root node (i.e., CS) (Top_event) into equivalent STA as shown in Fig. 16. Here, the STA broadcasts $initiate$ signal and waits for the $fail$ signal from the child nodes to disrupt the CS node. In addition, the clock x is a UPPAAL global variable where we declare $x = 0$ to keep track of the time progression as mentioned in Section 3.

An example of OR gate in the safety AFT involves the child node *Transfer Delay* of the LoA sub-tree shown in Fig. 13. The OR gate of the *Transfer Delay* is disrupted, when any of its child nodes *Features not working*, *Response Time* sends a *fail* signal as shown in Fig. 17. This $fail!$ signal is sent to the Top_event which forces a transition to $Disrupt$ state, representing LoA in the system thereby disrupting cybersickness. Similarly, STAs for the AND gate, SAND gates and the leaf nodes are also developed.

For instance, we consider the sub-tree LoA in Fig. 13 whose intermediate node *Timelag* is converted into an STA representation with AND gate as shown in Fig. 18. For this, the STA for intermediate node *Timelag*, waits for the disrupt (*fail*) signal from both *Traffic Congestion*, *Transfer delay* as shown in Fig. 18. The *Timelag* node gets disrupted once the fail signal is received from its child nodes. Similarly, the node *Features not working* with a SAND gate is disrupted only if its child nodes *Unauthorized access*, *Data tampering* send the *fail* signal in a sequential manner (i.e., left to right order) as shown in Fig. 19.

Next, we explain PAND gate which we have used to model the faults in NSTA as shown in Fig. 20. In case of PAND gate for the node DoS in the LoA sub tree, we convert the equivalent STA where its children (i) DoS (ii) UR are initialized at the same time via *initiateA8* and *initiateA9* signals. The disruption of the PAND gate occurs once its

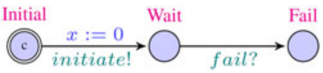


Fig. 16. STA of CS root node in safety AFT.

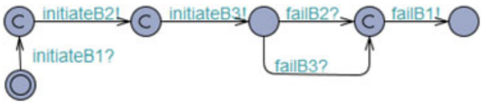


Fig. 17. STA of OR gate and signal transition between the root node and the childnode for the LoA subtree in the safety-AFT.

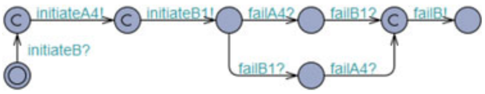


Fig. 18. STA of AND gate for the LOA sub-tree of the safety AFT.

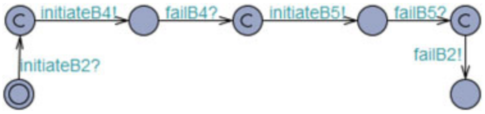


Fig. 19. STA for SAND gate in safety AFT.

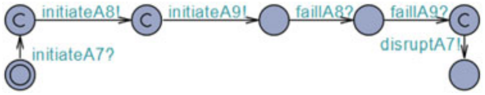


Fig. 20. STA for PAND gate in safety AFT.

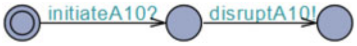


Fig. 21. STA for leaf node data tampering in LOA subtree of the safety AFT.

both children are disrupted in a sequential manner as shown in Fig. 20.

Moreover, the leaf node *Data Tampering* of the LOA subtree is converted to STA as shown in Fig. 21. Here, the STA gets activated after receiving *initiateA10* signal form its parent node *FNW* and disrupts the gate by sending a disrupt signal. Each of these converted STA leaf nodes are instantiated with λ (rate of exponential) values. For the given λ values to the leaf nodes, the probability of occurrence is calculated. This value then propagates upward in the tree to calculate the probability of LoA thereby can be used to determine the probability of the main root node of the safety AFT i.e., CS.

Using this translational approach, we convert all the nodes including the leaf nodes and their associated BAS in our safety AFT into equivalent STAs. These developed STAs are composed using the parallel composition [41] technique to form an NSTA, which is then used for SMC by the UPPAAL tool [12].

4.3.1 Attacker Profiling (AP) for BAS

The level of cybersickness induced due to each of these threat factors is also dependent on the considered attacker profiles (AP). For this, we develop AP based on prior works [43], [54] with attributes such as – (i) *time* taken to execute an attack, (ii) *cost* incurred to perform the attack, (iii) *skill* level of the attacker, and (iv) *resources* required to perform a cyber/immersion attack [55], [56] as shown in Table 2. To explain, we enlist the APs required to perform cyber-attacks/immersion attacks (e.g., DoS, Impersonation, SQL injection, control of users) as shown in the Table 2. With these generated AP, we showcase the logical steps taken by an attacker (i.e., BAS) to disrupt a leaf node. Each of these BAS are equipped with exponential distribution, discrete probabilities [10] in the safety AFT. In addition, we incorporate these APs for quantitative evaluation (i.e., in-depth analysis) to identify the most vulnerable components in the generated safety AFT of social VRLE applications. These APs can be further extended depending on the attributes considered for profiling.

TABLE 2
Attacker Profiles for Modeling Different BAS

Type of Attack	Attacker	Skills	Resources
DoS Attack	Attacker1	High	Low (ping sweeping),
	Attacker2	Low	High (sync flood)
Impersonation	Attacker1	High	Low (spoofing attack),
	Attacker2	Low	High (Session Hijack)
SQL injection	Attacker1	High	High (Fraud. access),
	Attacker2	Low	Medium (Alter Data)
Insert malicious scripts	Attacker1	High	Low (Add URL),
	Attacker2	Low	Medium (spooky vis.)

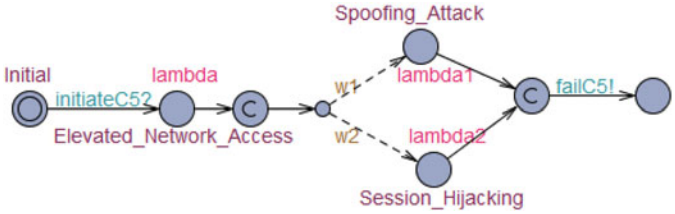


Fig. 22. BAS for impersonation.

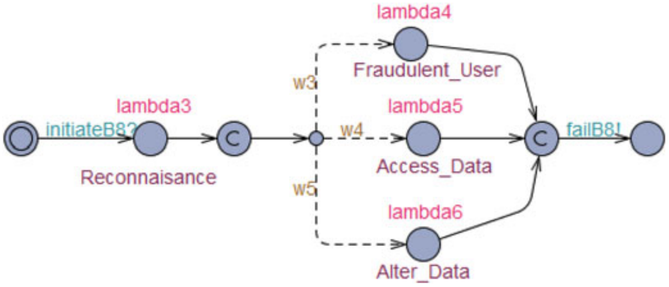


Fig. 23. BAS for SQL injection.

4.3.2 Modeling of Basic Attack Step (BAS)

A BAS is equipped with an exponential distribution representing the attack duration (i.e., λ) over the edges and nodes, as well as discrete probabilities quantifying the attack success irrespective of the execution time. In each of the BAS, the quantitative probabilities are obtained using the weighted probabilities ($w1$, $w2$) [43], [54], [55] as shown in Figs. 22 and 23. We next use these probabilities as input factors to the formulation detailed in [10] in order to determine the λ value for each attack event i.e., rate of exponential over the edges and nodes. Such a derivation of λ values of the attack events in Table 2 are listed in Table 4. For every attack scenario in BAS, the steps taken by the attacker are different depending on the attacker profile.

To explain our approach in modeling a BAS, we consider the leaf nodes – *Impersonation, SQL injection, Insert malicious scripts and Denial of service* in our safety AFT. For instance, the STA of the leaf node *impersonation* gets activated after receiving $initiate$ signal from the parent node as shown in the BAS representation Fig. 22. The BAS for *Impersonation* leaf node is equipped with weighted probabilities as shown in Fig. 22. This BAS allows us to model the likelihood of choosing an attack path based on the weighted probabilities. For example,

TABLE 3
λ Values for Leaf Nodes of Security & Privacy Subtrees in Safety AFT

LoC subtree		LoI subtree		LoA subtree		Privacy leakage subtree	
Threat scenario	λ	**Threat scenario**	λ	**Threat scenario**	λ	**Threat scenario**	λ
Login to system	0.0089	Packet spoofing	0.0068	Network traffic	0.0000113	Ping sweeping	0.002162
User data	0.004162	SYNC flood	0.0068	Upload/down -load rate	0.0001	User physical location	0.0000078
Capture packets	0.00098	Modify sensitive data	0.002642	Unauthorized access	0.006478	Password attacks	0.08687
Analyze packets	0.0048	Change coordinates	0.002642	Low Bandwidth	0.000121	Capture Hostname	0.004162
Impersonation	0.006892	Identity spoofing	8.1219E-06	——-		Intrusion	0.006628
User login	0.0089	Insert malicious scripts	0.008			Listen to conversation	0.08

in case of *Impersonation* node, after getting elevated access to the network, the attacker chooses a path (i.e., by performing a spoofing or session hijacking attack) with probabilities $w1/w1+w2$, $w2/w1+w2$, respectively as shown in Fig. 22. After exploiting one of these attack paths, a *fail* signal is sent to the parent node to denote disruption of the *Impersonation* node in our safety AFT. In addition, the λ values listed in Table 2 are obtained from our prior works [2], [53] and from [57], [58].

Another example of BAS of SQL injection is shown in Fig. 23, where the logical steps taken by the attacker are outlined. To elucidate, the attacker can perform SQL injection in following ways: i) login as a fraudulent user, ii) access data, and iii) alter data, without having authorized access. In addition, the probability of taking different paths is described based on the probability weights i.e., $w3$, $w4$ and $w5$ distributed over the edges. After disruption, *fail* signal is sent to the parent node representing disruption of the respective event. Using such translation of AFTs, we next perform quantitative analysis to show the effectiveness of our proposed framework implementation that is based on the secure-by-design concept.

5 Quantitative Results

In this section, we present the quantitative results obtained from experiments that use our proposed framework. We quantitatively assess individual as well as combinations of leaf nodes in the safety AFT, to study how multiple SPS threats associated to cyber-attacks/fault-attacks affect the cybersickness occurrence. For this, we consider the subtrees pertaining to threat scenarios – *LoI, LoC, LoA* and *privacy leakage* for the outlined safety AFT shown in Fig. 11. In the following analysis, we assume that our design requirement is to keep the probability of cybersickness below the threshold of 0.25. For evaluation purposes, we use λ values based on: (i) our BAS approach detailed in Section 4.3.2, (ii) quantified λ values [59], [60] by performing the attack scenarios i.e., different attack steps as detailed in Section 3 and from our prior works [2], [53], and (iii) assignment of arbitrary values. For the quantification of λ values, we start from the leaf node of AFT – BAS, to get the attack occurrence over a period of time which can then be used as the rate of exponential i.e., the λ values listed in Table 3.

Arbitrary λ values are considered only for a few anomaly events in the safety AFT mainly due to their rate of occurrence being very low. In such cases, we assign a relatively low λ value which is arbitrary mainly for the nodes such as {outage, upgrade, sensor disruption, faulty application behavior, outdated software, login to system, user login, password attacks} of the safety AFT. For example, a power outage scenario has a less likelihood of occurrence in comparison with other fault scenarios such as faulty application behavior or network faults. Thus, we consider a relatively small arbitrary λ value for the power outage fault event (λ = 0.001). We remark that our STA models are parametric, and thus any other user-defined λ values can be easily analyzed without any major changes to our methodology.

Using these λ values as parameters to the leaf nodes, we utilize the SMC queries as explained in Section 3.3 to analyze and find the probability of cybersickness for the generated STAs. For our experimental purposes, we consider CS (i.e., root node of safety AFT) as the goal node. Any other user-specified threshold values for different applications can also be used in our framework. This is due to the fact that the model checking approach takes the user-specified values at the beginning of an experiment. In addition, we also consider an error bound ϵ value of 0.01 and 95% confidence interval for the calculation of probability of disruption. In the following set of experiments, we present the obtained probability of the goal nodes with respect to the time window used by the attacker.

5.1 Vulnerability Analysis in the Safety AFT

With this quantitative analysis, we identify the most vulnerable components that can act as inducing factors for cybersickness in a VRLE session. For this, we assign the values of λ for the leaf nodes shown in Table 3. For the remaining leaf nodes in the safety AFT, we consider a very small positive constant $(K) \approx 0.00001$. This is because, in real world systems, multiple attack scenarios can happen. To identify a vulnerability in the safety AFT, we analyze: (i) individual leaf nodes, and (ii) combinations of leaf nodes, to determine their effect on the probability of cybersickness occurrence.

TABLE 4
Values of λ Given to the Basic Attack Steps of the Safety AFT

Basic attack steps of safety AFT	
Threat scenarios	λ
Impersonation	0.00004925, 0.00005466
SQL injection	0.0000502, 0.0000854, 0.0000492
Insert malicious scripts	0.0005389, 0.0006899
Denial of service	0.0003638, 0.0003084

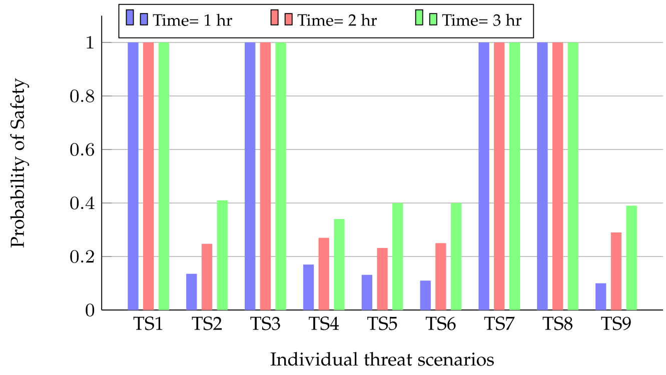


Fig. 24. TS of safety AFT where - $TS3$, $TS7$, $TS8$ are the most vulnerable nodes.

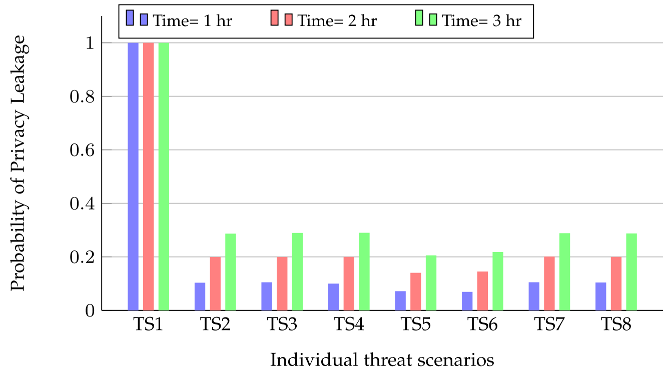


Fig. 25. TS of privacy leakage AFT where - $TS1$ is the most vulnerable node.

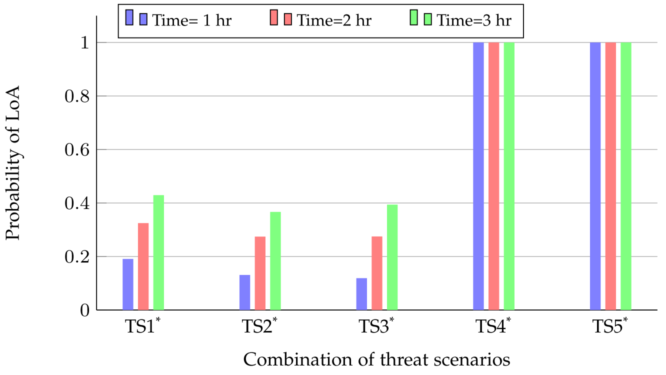


Fig. 26. TS of loss of availability AFT where - $TS4$ an $TS5$ are the most vulnerable nodes.

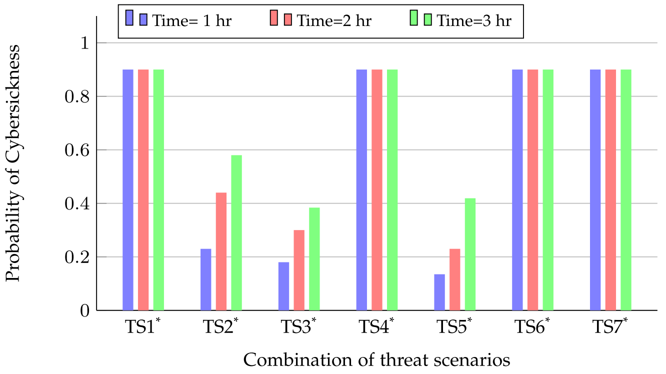


Fig. 27. Combinations of TS of safety AFT where - $TS1$, $TS4$, $TS6$ are the most vulnerable nodes.

5.1.1 Individual Leaf Node Analysis

In Fig. 24, we show the probability of cybersickness over multiple time windows for each leaf node in the safety AFT. We perform a thorough analysis of leaf nodes in the safety AFT for threat scenarios across different time intervals i.e., *t = {1 hr, 2 hr and 3 hr}*. For the individual leaf node analysis, the considered threat scenarios (TS) shown in Fig. 24 are termed as: $TS1$ – Impersonation, $TS2$ – User login, $TS3$ – User physical location, $TS4$ – Capture packets, $TS5$ – Ping sweeping, $TS6$ – Intrusion, $TS7$ – Upgrade, $TS8$ – Low bandwidth, $TS9$ – Network fault. As shown in Fig. 24, the leaf nodes $TS1$ and $TS8$ (for causing a DoS attack), $TS3$ and $TS7$ (for sensitive data leakage) are the most vulnerable in the safety AFT with the probability of 1. In addition, we also determine $TS6$ that is considered as the next vulnerable node in the safety AFT.

With the exhaustive simulation of security/privacy attack scenarios in VRLEs, we consider multiple LoC, LoI, LoA and privacy leakage scenarios for an in-depth quantitative analysis of the system. These multiple scenarios are possible due to the capability of AFTs in capturing the multi-stage and dynamic temporal inter-dependencies as a cause-and-effect relationship. Next, we show a proof of in-depth quantitative analysis of our AFTs (including safety AFT and subtrees), where we determine the probability of occurrence of security/privacy threats changing for different attack goals (e.g., cybersickness, LoA). For example, Figs. 25 and 26 show such multi-stage scenarios for LoA and privacy leakage of the safety AFT. To elucidate, we describe Unauthorized Data Access (LoC-UDA) has the maximum probability of occurrence for privacy leakage as shown in Fig. 25. Similarly, for the LoA subtree, we determine from our in-depth quantitative analysis that the Outage and Upgrade nodes have the highest probability of occurrence for the LOA issue as shown in Fig. 26.

5.1.2 Combination of Leaf Node Analysis

Herein, we consider combinations of leaf nodes to identify their impact on cybersickness. For these experiments, we explore two scenarios: In the first scenario, we consider combinations of leaf nodes that belong to the same sub-tree (e.g., LoI). In the second scenario, we consider leaf nodes from different sub-trees (e.g., LoA and privacy leakage). The considered combination of threat scenarios are enlisted as: $TS1^*$ – {Capture hostname, Ping sweeping}, $TS2^*$ – {Data tampering, user information}, $TS3^*$ – {Data tampering, faulty application behavior}, $TS4^*$ – {Low bandwidth, unauthorized access}, $TS5^*$ – {Sync flood, ping sweeping}, $TS6^*$ – {Intrusion, Listen to conversation}, $TS7^*$ – {Impersonation, Packet Spoofing}. As shown in Fig. 27 $TS1^*$, $TS4^*$, $TS6^*$ and $TS7^*$ are the most vulnerable combination of threat scenarios with a probability of 0.9 for a cybersickness event. As part of further analysis in Section 6, we discuss about the potential candidates for design principles to apply on these leaf nodes such that the VRLE application resilience is enhanced against security threats.

5.2 Evaluation of Attacker Profiles in Terms of Impact on Disruption of Safety AFT

In this section, we analyze how APs impact the disruption of an intermediate node, root node (CS) for the safety AFT

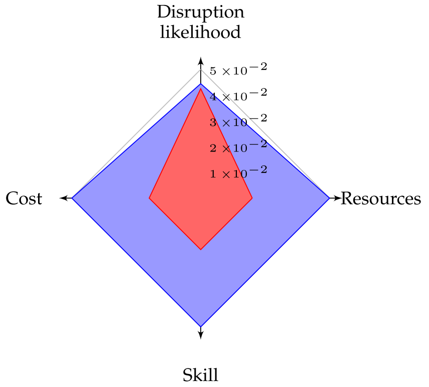


Fig. 28. Probability of disruption of cybersickness for different attacker profiles.

shown in Fig. 11. For instance, the Fig. 28 considers the APs enlisted in Table 2 along with the AP attributes such as cost, resources, skills required and their associated likelihood of the CS disruption in the safety AFT. From the graphical analysis, we determine that for *time* $\approx$ 3 *hours*, the disruption of likelihood of CS for *Attacker 1* is 4.44×10^{-2} and for *Attacker 2* is 4.25×10^{-2}. Similarly, in Fig. 29, we also analyze the impact on disruption of an intermediate node $\{$*Response Time*$\}$ in the LoA subtree of the safety AFT. From our results in Figs. 28 and 29, we determine this change of probability of disruption between the considered APs is due to the change in the profile attributes i.e., skill, cost, time, resources. With this analysis, we consider both the attacker profiles in determining the vulnerable components for a given social VRLE application design.

5.3 Risk Assessment Based on Vulnerability Analysis and Attacker Profiles

In this section, we perform risk assessment on the identified vulnerability components outlined in Section 5.1 for the safety AFT. To determine the severity of these identified threat vectors in a VRLE, we adopt a widely accepted NIST-based risk assessment procedure [37], [61]. Taking the vulnerability results into account, we calculate their associated risk values based on the formulation

$$R = P(O) \times I, \tag{1}$$

where, $P(O)$ is the probability of occurrence, I is the level of impact in the event of attack occurrence and R is the associated risk. With the impact analysis results in Section 3.2, we further map the cybersickness levels to a numerical scale of 1(Low)-to-5(High) in the quantitative analysis of the AFTs. In addition, these results are also considered as part of the input factors for our NIST-based risk assessment termed as “Impact” defined in Equation (1).

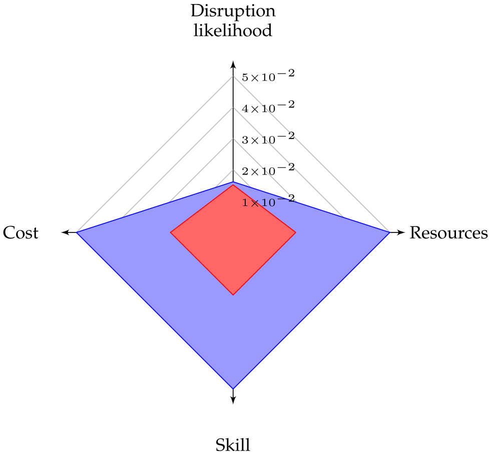


Fig. 29. Probability of disruption of response time in safety attack tree for different attacker profiles.

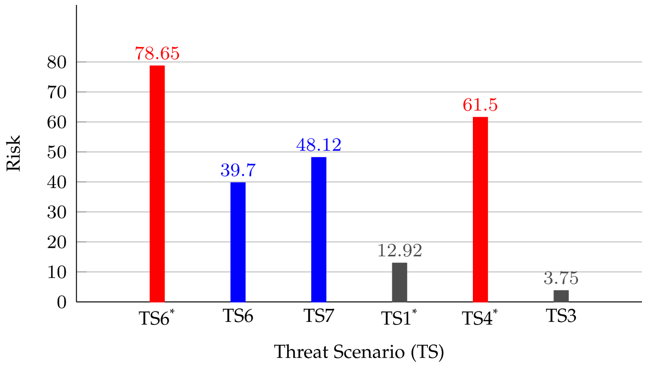


Fig. 30. Risk assessment of threats affecting cybresickness.

In addition, we use the semi-quantitative scale based on the NIST SP 800-53 [37] to categorize the risk levels as *‘High’*, *‘Moderate’*, *‘Low’* based on the I and $P(O)$ values for a given threat scenario. For this, we determine the $P(O)$ from our vulnerability analysis of the STAs related to the safety AFT discussed in Section 5.1. Similarly, we estimate the I using NIST guidelines in terms of impact on cybersickness based on our preliminary experiments results on cybersickness levels as shown in Fig. 8. The rational behind such a calculation is to get the most conservative estimate of the critical components in the VRLE application. To categorize into the appropriate risk level for each of the threat scenarios, we first calculate the risk% using the Equation (1). To elucidate, the calculated risk% is assigned the risk level using a percentage scale from 0 to 100%, where $>= 100\%$ (*very high*), $> 70\%$ (*high*), $> 50\%$ (*moderate*), $> 20\%$ (*low*), $< 20\%$ (*very low*).

The risk assessment shown in Fig. 30 details the risk value of each of the most vulnerable components considered as threat scenarios in Section 5.1. These threat scenarios – $TS3$, $TS6$, $TS7$, $TS1^*$, $TS4^*$, $TS6^*$ are enlisted on the X-axis and the Y-axis represents the associated risk%.

Based on our risk assessment results, we identify that $TS6^*$ has a higher risk level with value 78.65%. Similarly, $TS4^*$ also falls under high risk category when compared to $TS3$, $TS6$, $TS7$ and other combinations. In addition, the risk levels for $TS6$, $TS7$ are higher than the combination $TS1^*$, which is a privacy breach scenario. The low risk level we can observe for $TS1^*$ can be considered as a potential risk factor if the hostname of the user is obtained for an attacker with higher AP. From our risk analysis, we can also determine that - among the considered threat scenarios, a *DoS* and a *man-in-the-room attack* cause higher risk for occurrence of cybersickness. Based on the above, we can see how SP attacks can be used to trigger novel attack scenarios such as an immersion attack, to induce undesired cybersickness levels. Thus from our above quantitative results, we can conclude that along with physiological conditions, the security and privacy related threats can act as major factors for inducing cybersickness in a VRLE session.

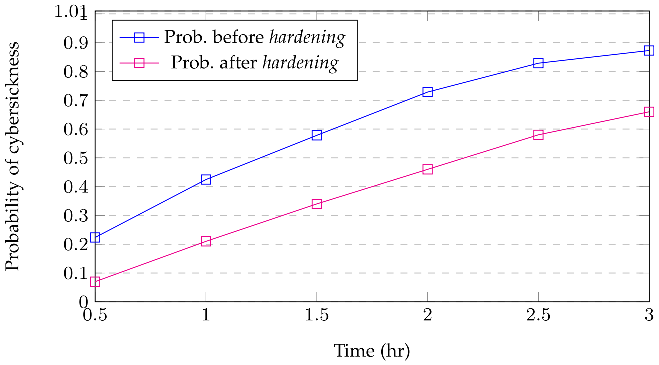


Fig. 31. Probability of cybersickness reduced by 24.14% in safety AFT due to application of hardening design principle.

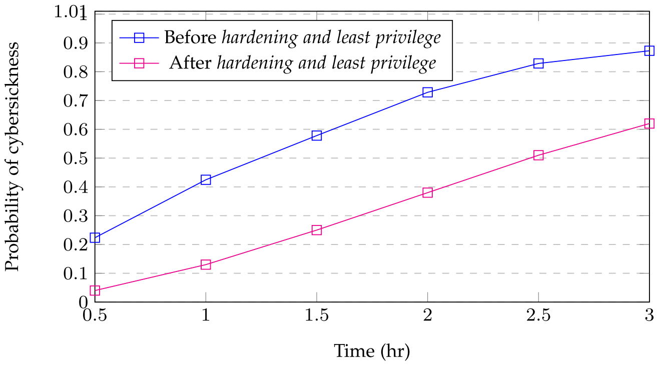


Fig. 33. Probability of cybersickness reduced by 28.96% in safety AFT due to application of hardening and principle of least privilege design principles.

6 Recommended Design Principles

In this section, we examine the effect of applying various design principles to the most vulnerable components identified in the Section 5.1 for our safety AFT. Existing works such as NIST SP800-160 [37], [42] suggest that the services for safeguarding security and privacy are critical for successful operation of current devices and sensors connected to physical networks as part of edge computing systems. As mentioned in Section 3.4, these design principles are essential to construct a trustworthy edge computing based system architecture. The goal is to apply a combination of design principles at different levels of abstraction to help in developing effective mitigation strategies. We adopt a selection of design principles such as *hardening*, *diversity*, *Redundancy* and *principle of least privilege* among the list of principles available in NIST SP800-160. In the following, we demonstrate their effectiveness by showing that there is a reduction in the probability of disruption terms after adopting them in a VRLE design.

Implementation of Design Principles on Safety AFT. To study the effect of design principles on the cybersickness occurrence, we incorporate *hardening*, *redundancy* as shown in Figs. 31 and 32 relating to the safety AFT. Specifically, each of the subtrees in the safety AFTs are updated with new nodes after incorporating the security principles. Next, we re-evaluate each of the AFTs to analyze the impact on cybersickness occurrence in the event of an anomaly. For instance, as part of *hardening* principle, we added two new nodes : {Addition of firewall rules, Security protocols} to the "password attack" node to address Unauthorized access as shown in the privacy leakage subtree in Fig. 14. We then analyze the impact on the most vulnerable nodes triggering a DoS attack in the safety AFT where, we observe that the probability of cybersickness disruption is reduced from 0.87 to 0.66 (24.14%), with the given attacker profile as shown in Fig. 31.

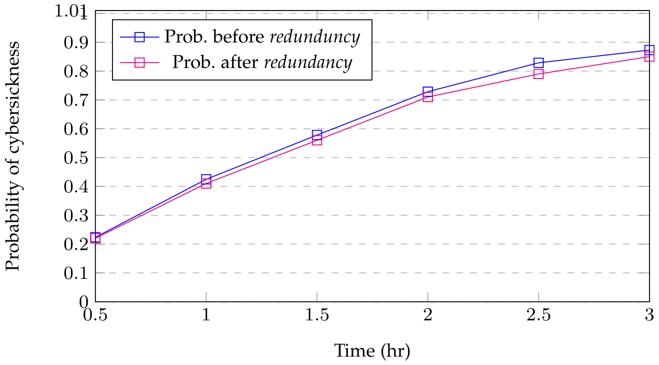


Fig. 32. Probability of cybersickness reduced by 2.6% in safety AFT due to application of redundancy design principle.

The decrease in the disruption of cybersickness is due to the increase in the resource requirement for an attacker to compromise a VRLE application which is incorporating the *hardening* principle. We apply the *redundancy principle*, where we add new nodes with similar functionalities (i.e., multiple authentication mechanisms) to the impersonation node in the LoI subtree shown in Fig. 12 of the safety AFT. This redundancy principle on the safety AFT intentionally reduces the probability of disruption of cybersickness by 2.6% as shown in Fig. 32. Similarly, we add the *principle of least privilege* in the privacy leakage subtree of the safety AFT, where we add a new child node "Access limitation" to the eavesdropping node and observe that the probability of cybersickness in the safety AFT is reduced by 2.61%.

Thus, from the above implementation of individual design principles, we can observe that *hardening* is more effective in reducing the disruption of cybersickness compared to any of the other design principles. In addition, our results demonstrate the benefits in implementing a combination of design principles in safety AFT to overall improve the attack mitigation efforts.

To study the effect on disruption of the cybersickness, we adopt a combination of design principles for the safety AFT such as: (i) {*hardening*, *principle of least privilege*}, (ii) {*hardening*, *redundancy*}, and (ii) {*Redundancy*, *principle of least privilege*}. We observe that there is a significant drop in the probability of disruption of cybersickness from 0.87 to 0.62 (28.96%) due to {*hardening*, *principle of least privilege*} as shown in Fig. 33. Similarly, Fig. 34 shows the effect of combination of {*hardening*, *redundancy*} that results in reducing the disruption of cybersickness with 25.2% in a safety AFT. In addition, we apply the {*redundancy*, *principle of least privilege*} combination, which results in a reduction of cybersickness by 3.05% as shown in Fig. 35.

Finally, we evaluate the effect of all the combined design principles on safety AFT. The Pr of the cybersickness for the safety AFT is reduced by 35.18% as shown in Fig. 36. From the above numerical analysis, we can conclude that incorporating relevant combination of standardized design principles and

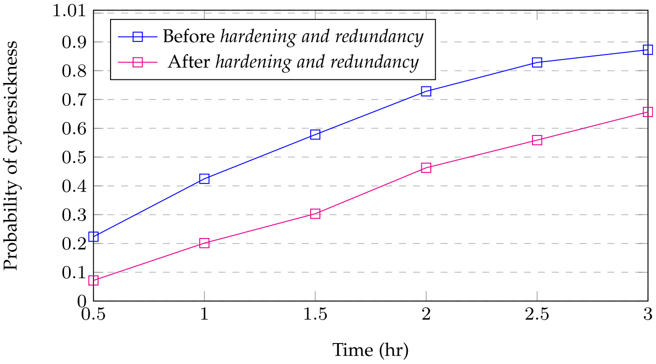


Fig. 34. Probability of cybersickness reduced by 25.52% in safety AFT due to application of hardening and redundancy design principles.

their joint implementation have the potential to better mitigate the impact of sophisticated and well-orchestrated cyber attacks on edge computing assisted VRLE systems with wearable devices. In addition, our above results provide insights on how the adoption of the design principles can provide the necessary evidence to support a trustworthy level of security, privacy and safety for the users in VRLE systems that are used for important societal applications such as: special education, surgical training, and flight simulators.

7 Discussion

We remark that our work is one of the first studies that focuses on analyzing SP issues in a VRLE system and measures their impact on triggering cybersickness. Specifically, there are three major advantages of our proposed framework. First, using our proposed framework, we identify the most vulnerable security and privacy issues inducing cybersickness in VRLEs. Second, we assess the risk level for each of the considered SP issues in a VRLE system. Lastly, we further use the measured risk level and the calculated impact on cybersickness in order to determine the specific node parameters in the safety AFT that require the implementation of security design principles in the VRLE application design.

There are two major limitations of our proposed framework. First, our framework efficacy is dependent on properly choosing the lambda values for the quantitative analysis of cybersickness likelihood. It is indeed possible to determine the lambda values by carefully performing extensive experiments on choosing different attack paths for triggering cybersickness [43], [55]. Second, in a relatively high-scale VRLE setup case, the AFT might grow too large, and pertinent attack/fault-tree reduction techniques will need to be employed. For instance, to reduce the size of the AFT in high-scale VRLE setup cases, popular techniques in the dynamic fault tree (DFT) domain such as graph rewriting [62], graph partitioning [63], or state space reduction through the bisimulation-based [64] technique can be employed.

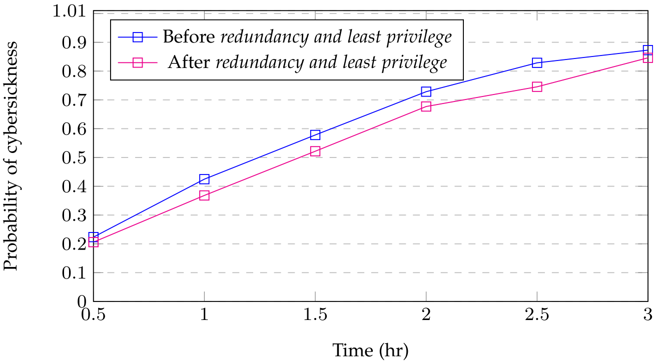


Fig. 35. Probability of cybersickness reduced by 3.05% in safety AFT due to application of redundancy and principle of least privilege design principles.

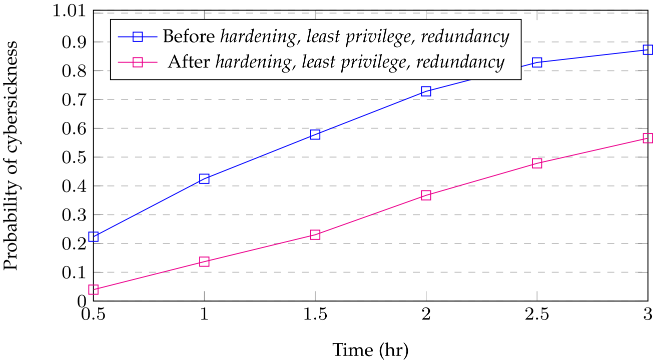


Fig. 36. Probability of cybersickness reduced by 35.18% in safety AFT due to application of hardening, redundancy and principle of least privilege design principles.

8 Conclusion

Social Virtual Reality Learning Environments (VRLE) provide immersive experience by delivering online content to distributed users. Currently, VRLE applications are being adopted in various application domains, however the related security, privacy and user safety (SPS) issues are under-explored. In this paper, we present a novel quantitative framework to analyze potential security, privacy issues that induce *cybersickness* (a safety issue) in a VRLE session. With our preliminary experiments that considered a social VRLE application case study viz., vSocial, we demonstrated the disruptive effects of various cyber-attacks/fault-attacks based SPS issues (i.e., DoS, eavesdropping via man-in-the-room) on cybersickness levels. We utilized attack-fault tree (AFT) formalism to model the security issues (i.e., LoC, LoI and LoA scenarios) and privacy issues (i.e., privacy leakage) inducing cybersicknesscybersickness in a VRLE session. Specifically, we developed relevant AFTs and converted them into stochastic timed automata (STA) and then performed model checking using the UPPAAL SMC tool.

Using our proposed framework, we determined causes of: *DoS attack, data leakage, man-in-the-room attack and unauthorized access* as the most vulnerable components that can induce higher level of cybersickness in VRLE sessions. By using the NIST SP800-16 risk assessment method, we determined the severity of these identified threat scenarios (i.e., critical issues) in terms of impact on cybersickness and degradation of application functionality. Furthermore, we illustrated the effectiveness of our framework by analyzing different design principle candidates. We showed a 'before' and 'after' performance comparison to investigate the effect of applying suitable design principles to reduce the probability of cybersickness occurrence. From our experiments with the vSocial application, we determined that the choice of a suitable design principle pertaining to the most vulnerable threat components is significant for use in an attack mitigation strategy. Specifically, we observed that implementing a combination of design principles can result in a more effective mitigation strategy. Among the design principle candidates, we found: (i) *{hardening, principle of least privilege}*, (ii) *{hardening, redundancy}* were the most suitable combinations for reducing the probability of cybersickness occurrence with an average of 27.24%.

As part of future work, our work on privacy AFT can be extended by including ethical issues outlined in [22]. In addition, evaluation of different attack-defense strategies and potential performance adaptations can be performed. This can help to assess their effectiveness in mitigating the cybersickness occurrence, to ultimately ensure a safe, trustworthy and high-performing VRLE to the application users.

Acknowledgments

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] C. Zizza, A. Starr, D. Hudson, S. S. Nuguri, P. Calyam, and Z. He, "Towards a social virtual reality learning environment in high fidelity," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf.*, 2018, pp. 1–4.

[2] A. Gulhane *et al.*, "Security, privacy and safety risk assessment for virtual reality learning environment applications," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf.*, 2019, pp. 1–9.

[3] L. Rebenitsch and C. Owen, "Review on cybersickness in applications and visual displays," *Virt. Reality*, vol. 20, no. 2, pp. 101–125, Jun. 2016. [Online]. Available: https://doi.org/10.1007/s10055–016-0285-9

[4] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 1606–1616, Apr. 2019.

[5] 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.

[6] K. Fu *et al.*, "Safety, security, and privacy threats posed by accelerating trends in the Internet of Things," *Comput. Community Consortium*, vol. 29, no. 3, pp. 1–9, 2017.

[7] J. J. LaViola, "A discussion of cybersickness in virtual environments," *ACM SIGCHI Bull.*, vol. 32, no. 1, pp. 47–56, Jan. 2000. [Online]. Available: https://doi.org/10.1145/333329.333344

[8] S. Davis, K. Nesbitt, and E. Nalivaiko, "A systematic review of cybersickness," in *Proc. Conf. Interactive Entertainment*, 2014, pp. 1–9.

[9] High fidelity. Accessed: Dec. 03, 2020. [Online]. Available: https://highfidelity.com

[10] R. Kumar and M. Stoelinga, "Quantitative security and safety analysis with attack-fault trees," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng.*, 2017, pp. 25–32.

[11] N. Bertrand *et al.*, "Stochastic timed automata," *Log. Methods Comput. Sci.*, vol. 10, pp. 1–73, 2014.

[12] A. David, K. G. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen, "Uppaal SMC tutorial," *Int. J. Softw. Tools Technol. Transfer*, vol. 17, no. 4, pp. 397–415, 2015.

[13] A. Souri, A. M. Rahmani, N. J. Navimipour, and R. Rezaei, "A symbolic model checking approach in formal verification of distributed systems," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, 2019, Art. no. 4.

[14] Securing your reality: Addressing security and privacy in virtual and augmented reality applications. [Online]. Available: https://tinyurl.com/yxlplqqe

[15] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, 2018.

[16] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. Int. Conf. Wireless Algorithms Syst. Appl.*, 2015, pp. 685–695.

[17] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, 2018.

[18] SteamVR. Accessed: Dec. 03, 2020, 2019. [Online]. Available: http://store.steampowered.com/steamvr

[19] A. Yarramreddy, P. Gromkowski, and I. Baggili, "Forensic analysis of immersive virtual reality social applications: A primary account," in *Proc. IEEE Security Privacy Workshops*, 2018, pp. 186–196.

[20] J. A. de Guzman, K. Thilakarathna, and A. Seneviratne, "Security and privacy approaches in mixed reality: A literature survey," *ACM Comput. Surv. (CSUR)*, vol. 52, no. 6, pp. 1–37, 2019.

[21] K. M. Stanney, R. R. Mourant, and R. S. Kennedy, "Human factors issues in virtual environments: A review of the literature," *Presence*, vol. 7, pp. 327–351, 1998.

[22] The XRSI medical and privacy framework. Accessed: Jul. 13, 2021. [Online]. Available: https://www.xrsi.org

[23] Virtual reality headsets could put childrens health at risk. Accessed: Dec. 03, 2020. [Online]. Available: https://www.theguardian.com/technology/2017/oct/28/virtual-reality-headset-children-cognitive-problems

[24] M. S. Dennison, A. Z. Wisti, and M. D'Zmura, "Use of physiological signals to predict cybersickness," *Displays*, vol. 44, pp. 42–52, 2016.

[25] S. Martirosov and P. Kopecek, "Cyber sickness in virtual reality - literature review," in *Proc. 28th Int. DAAAM Symp.*, 2017, pp. 718–726.

[26] Y. Farmani and R. J. Teather, "Viewpoint snapping to reduce cybersickness in virtual reality," in *Proc. 44th Graph. Interface Conf.*, 2018, pp. 168–175.

[27] A. M. Gavgani, F. R. Walker, D. M. Hodgson, and E. Nalivaiko, "A comparative study of cybersickness during exposure to virtual reality and "classic" motion sickness: Are they different?," *J. Appl. Physiol.*, vol. 125, no. 6, pp. 1670–1680, 2018.

[28] A. Tiiro, "Effect of visual realism on cybersickness in virtual reality," University of Oulu, 2018.

[29] H. K. Kim, J. Park, Y. Choi, and M. Choe, "Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment," *Appl. Ergonom.*, vol. 69, pp. 66–73, 2018. [Online]. Available: https://doi.org/10.1016/j.apergo.2017.12.016

[30] K. Tcha-Tokey, E. Loup-Escande, O. Christmann, and S. Richir, "A questionnaire to measure the user experience in immersive virtual environments," in *Proc. Virt. Reality Int. Conf.*, 2016, pp. 1–5.

[31] G. Regal, R. Schatz, J. Schrammel, and S. Suette, "VRate: A Unity3D asset for integrating subjective assessment questionnaires in virtual environments," in *Proc. 10th Int. Conf. Qual. Multimedia Experience*, 2018, pp. 1–3.

[32] Oculus rift: A virtual reality system to immerse in virtual worlds. Accessed: Dec. 03, 2020. [Online]. Available: https://www.oculus.com/rift

[33] Vive: Discover virtual reality beyond imagination. Accessed: Dec. 03, 2020. [Online]. Available: www.vive.com

[34] HumHub: The flexible open source social network kit. Accessed: Dec. 03, 2020. [Online]. Available: https://www.humhub.org/en

[35] Clumsy0.2. Accessed: Dec. 03, 2020. [Online]. Available: https://jagt.github.io/clumsy/download.html

[36] Wireshark. Accessed: Dec. 03, 2020. [Online]. Available: https://www.wireshark.org/

[37] R. Ross, M. McEvilley, and J. Oren, "Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems," 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800–160.pdf

[38] University news, "University of new haven researchers discover critical vulnerabilities in popular virtual reality application," Accessed: Dec. 03, 2020, 2019. [Online]. Available: https://tinyurl.com/u87l2oq

[39] E. M. Clarke Jr., O. Grumberg, D. Kroening, D. Peled, and H. Veith, *Model Checking*. Cambridge, MA, USA: MIT Press, 2018.

[40] H. L. Younes, M. Kwiatkowska, G. Norman, and D. Parker, "Numerical vs. statistical probabilistic model checking," *Int. J. Softw. Tools Technol. Transfer*, vol. 8, no. 3, pp. 216–228, 2006.

[41] P. Ballarini, N. Bertrand, A. Horváth, M. Paolieri, and E. Vicario, "Transient analysis of networks of stochastic timed automata using stochastic state classes," in *Proc. Int. Conf. Quantitative Eval. Syst.*, 2013, pp. 355–371.

[42] A. Laszka, W. Abbas, Y. Vorobeychik, and X. Koutsoukos, "Synergistic security for the industrial Internet of Things: Integrating redundancy, diversity, and hardening," in *Proc. IEEE Int. Conf. Ind. Internet*, 2018, pp. 153–158.

[43] G. Norman, D. Parker, and J. Sproston, "Model checking for probabilistic timed automata," *Formal Methods Syst. Des.*, vol. 43, no. 2, pp. 164–190, 2013.

[44] Common vulnerability scoring system SIG. Accessed: Dec. 15, 2020. [Online]. Available: https://www.first.org/cvss/

[45] A. Shostack, "Experiences threat modeling at microsoft," in *Proc. MODSEC@ MoDELS*, 2008, pp. 1–11.

[46] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "SAHARA: A security-aware hazard and risk analysis method," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2015, pp. 621–624.

[47] C. Raspotnig, V. Katta, P. Karpati, and A. L. Opdahl, "Enhancing chassis: A method for combining safety and security," in *Proc. Int. Conf. Availability Rel. Secur.*, 2013, pp. 766–773.

[48] C. Schmittner, Z. Ma, and P. Smith, "FMVEA for safety and security analysis of intelligent and cooperative vehicles," in *Proc. Int. Conf. Comput. Saf. Rel. Secur.*, 2014, pp. 282–288.

[49] F. Arnold, D. Guck, R. Kumar, and M. Stoelinga, "Sequential and parallel attack tree modelling," in *Proc. Int. Conf. Comput. Saf. Rel. Secur.*, 2014, pp. 291–299.

[50] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl, "Fault tree handbook," *Nuclear Regulatory Commission Washington DC*, 1981.

[51] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Comput. Sci. Rev.*, vol. 15, pp. 29–62, 2015.

[52] B. Kordy, L. Pi ètre-Cambacédès, and P. Schweitzer, "Dag-based attack and defense modeling: Don't miss the forest for the attack trees," *Comput. Sci. Rev.*, vol. 13, pp. 1–38, 2014.

[53] S. Valluripally, A. Gulhane, R. Mitra, K. A. Hoque, and P. Calyam, "Attack trees for security and privacy in social virtual reality learning environments," in *Proc. 17th IEEE Annu. Consum. Commun. Netw. Conf.*, 2020, pp. 1–9.

[54] M. Rocchetto and N. O. Tippenhauer, "On attacker models and profiles for cyber-physical systems," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2016, pp. 427–449.

[55] Attacker classification to aid targeting critical systems for threat modelling and security review. Accessed: Dec. 03, 2020. [Online]. Available: http://www.rockyh.net/papers/AttackerClassification.pdf

[56] Profiles of cyber-criminals and cyberattackers. Accessed: Dec. 03, 2020. [Online]. Available: https://tinyurl.com/rstn7df

[57] A. Lenin, J. Willemson, and D. P. Sari, "Attacker profiling in quantitative security assessment based on attack trees," in *Proc. Nordic Conf. Secure IT Syst.*, 2014, pp. 199–212.

[58] E. Ruijters, D. Guck, M. van Noort, and M. Stoelinga, "Reliability-centered maintenance of the electrically insulated railway joint via fault tree analysis: A practical experience report," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2016, pp. 662–669.

[59] P. Saripalli and B. Walters, "QUIRC: A quantitative impact and risk assessment framework for cloud security," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, 2010, pp. 280–288.

[60] M. Kiani, A. Clark, and G. Mohay, "Evaluation of anomaly based character distribution models in the detection of SQL injection attacks," in *Proc. 3rd Int. Conf. Availability Rel. Secur.*, 2008, pp. 47–55.

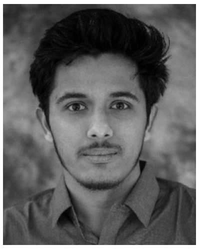
[61] M. Dickinson *et al.*, "Multi-cloud performance and security driven federated workflow management," *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, pp. 240–257, First Quarter 2021.

[62] S. Junges, D. Guck, J.-P. Katoen, A. Rensink, and M. Stoelinga, "Fault trees on a diet," in *Proc. Int. Symp. Dependable Softw. Eng.: Theories Tools Appl.*, 2015, pp. 3–18.

[63] N. Cauchi, K. A. Hoque, A. Abate, and M. Stoelinga, "Efficient probabilistic model checking of smart building maintenance using fault maintenance trees," in *Proc. 4th ACM Int. Conf. Syst. Energy-Efficient Built Environ.*, 2017, pp. 1–10.

[64] A. Abate, "Approximation metrics based on probabilistic bisimulations for general state-space Markov processes: A survey," *Electron. Notes Theor. Comput. Sci.*, vol. 297, pp. 3–25, 2013.

Samaikya Valluripally received the BTech degree in computer science from Jawaharlal Nehru Technological University, India, in 2014, and the MS degree in computer science from the University of Missouri-Columbia, Missouri, in 2016. She is currently working toward the PhD degree in computer science at the University of Missouri-Columbia, Missouri. Her research interests include cloud computing, cloud security for AR/VR and IoT applications, big data analytics.



Aniket Gulhane received the BE degree in computer engineering from Savitribai Phule Pune University, India, in 2016. He is currently working toward the MS degree in computer science at the University of Missouri-Columbia, Missouri. His research interests include cloud security, formal methods and human computer interaction.

Khaza Anuarul Hoque (Senior Member, IEEE) received the MSc and PhD degrees from the Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada, in 2011 and 2016, respectively. He is currently an assistant professor in the Department of Electrical Engineering and Computer Science, University of Missouri-Columbia (MU), Missouri, where he directs the Dependable Cyber-Physical Systems (DCPS) Laboratory. Before joining MU, he was a postdoctoral research fellow at the University of Oxford, U.K. His research interests include formal methods, cyber-physical systems, cybersecurity, and safe AI/ML.

Prasad Calyam (Senior Member, IEEE) received the MS and PhD degrees from the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, Ohio, in 2002 and 2007, respectively. He is currently an associate professor in the Department of Computer Science, University of Missouri-Columbia, Missouri. Previously, he was a research director at the Ohio Supercomputer Center. His research interests include distributed and cloud computing, computer networking, and cyber security.