International Journal of Safety and Security Engineering
Vol. 16, No. 3, March, 2026, pp. 535-545
Journal homepage: http://iieta.org/journals/ijsse

Secure Virtual Reality Streaming Under Attack: Fine-Grained Modeling of the Latency Security Trade-Off

Raghad Hazim Saeed[1], Qutaiba I. Ali[2], Farhad E. Mahmood[3*]

[1] Department of Environmental Technologies, College of Environmental Science, University of Mosul, Mosul 42001, Iraq
[2] Department of Computer Engineering, College of Engineering, University of Mosul, Mosul 42001, Iraq
[3] Department of Communications and Intelligent Digital Systems, College of Engineering, University of Mosul, Mosul 42001, Iraq

Corresponding Author Email: farhad.m@uomosul.edu.iq

https://doi.org/10.18280/ijsse.160307

Received: 21 February 2026
Revised: 20 March 2026
Accepted: 26 March 2026
Available online: 31 March 2026

Keywords:
virtual reality, transport layer security, TLS 1.3/DTLS, DDoS mitigation, security model, content delivery, latency model

ABSTRACT

Virtual Reality (VR) streaming systems require ultra-low latency and high throughput to maintain immersive user experiences, yet these requirements make them highly vulnerable to cyberattacks such as Distributed Denial-of-Service (DDoS) and Man-in-the-Middle (MitM). This creates a fundamental challenge in balancing strong security mechanisms with strict real-time performance constraints. This paper presents a latency-aware analytical framework and secured VR streaming architecture that quantitatively models the trade-off between cybersecurity and system performance under adversarial conditions. The proposed approach integrates transport layer security (TLS) 1.3/ Datagram TLS (DTLS), AES-256 encryption, and HMAC-based authentication within a VR streaming pipeline, and introduces a unified model that captures the effects of attack amplification and defense efficiency on end-to-end latency and application goodput. To validate the model, a controlled simulation of a 5-minute VR session over a Wi-Fi 6 environment is conducted, including a DDoS-style traffic flooding scenario. The evaluation compares four configurations: baseline, under attack, secured, and optimized secured. Results show that while attacks significantly increase latency and degrade throughput, the secured architecture maintains stable performance with only moderate overhead (≈30–32 ms latency), and the optimized configuration further reduces latency to approximately 25 ms while preserving security guarantees. Unlike existing VR security approaches that focus on isolated attack detection or privacy protection, the proposed framework provides a quantitative and system-level perspective on the security–performance trade-off. The model can support the design and optimization of real-world VR streaming systems, enabling secure and responsive operation in emerging immersive applications.

1. INTRODUCTION

Virtual Reality (VR) rises, technology creates revolutionary changes across gaming sectors, together with education and healthcare, as well as remote collaboration. The core performance requirements for VR platforms involve maintaining ultra-low latency better than 20 ms while ensuring high-throughput streaming of 25–50 Mbps for delivering high-resolution stereoscopic content. The requirements of VR systems make them prone to network disruptions caused by minor delays as well as packet loss because they function as naturally sensitive systems. Research shows that any transmission delay longer than 20 milliseconds results in motion sickness and weakens user satisfaction [1]. The combination of being open and maintaining high bandwidth makes VR streaming systems prone to numerous cyber threats. The integrity and confidentiality, together with the availability level of VR sessions becomes hacked due to Distributed Denial-of-Service (DDoS) and Man-in-the-Middle (MitM) attacks, plus TCP Reset and Replay attacks, as shown in Figure 1. The lack of built-in protocol-level protections, along with real-time sensitivity in VR environments, makes these threats particularly dangerous when UDP is used for fast streaming. Research findings show that AR/VR platforms allow hacker applications to detect secret user inputs like hand motions and spoken instructions without requiring specific permission access [2, 3]. Integration of contemporary security protocols such as transport layer security (TLS) 1.3, AES-256, and HMAC into VR systems and Datagram TLS (DTLS) adds computation overhead, which threatens to breach performance thresholds necessary for immersive experiences. Current platforms need to choose between security for speed, or they apply generic protections that do not match VR-related real-time requirements [4-9].

Researchers have identified particular areas lacking knowledge regarding VR content delivery security and privacy: i) The research needs better methods to establish end-to-end encryption, which must secure VR streaming sessions

effectively without compromising streaming quality. ii) The absence of timely attack identification methods exists, which specifically detect and counteract both network-based and content-level attacks during VR streaming operations. iii) The shortage of authentication techniques exists that operate securely with minimal resources required for VR equipment. iv) The verification of VR content integrity in real time represents a straightforward problem with limited research addressing this topic to prevent unauthorized modifications or attack injections.

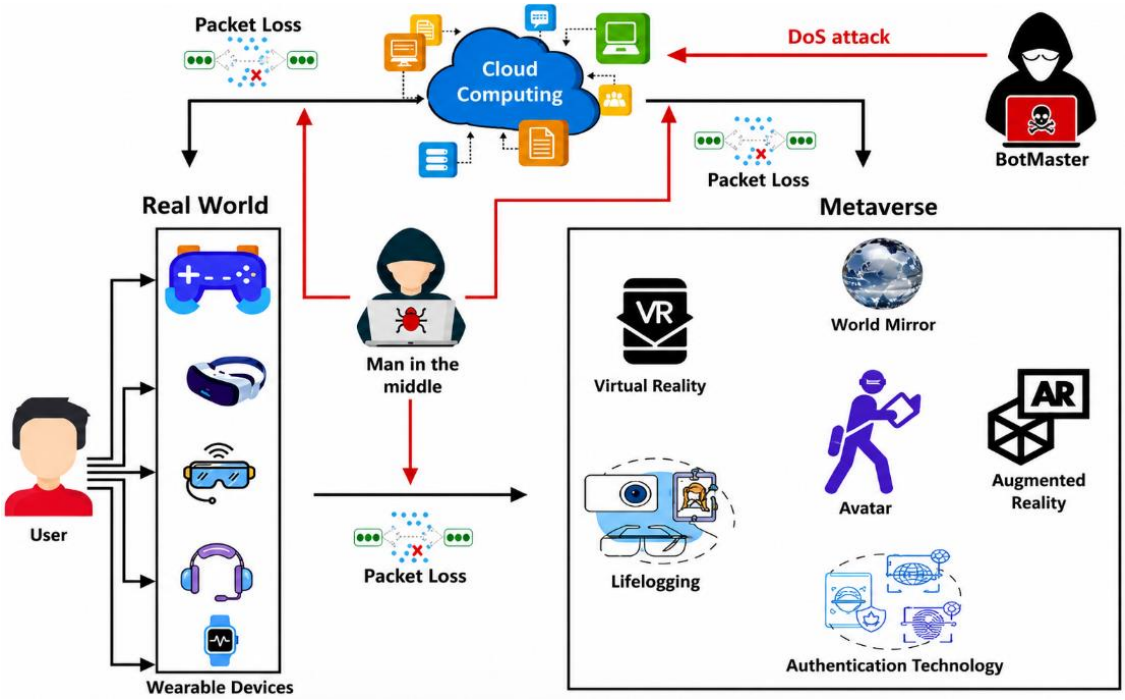

Figure 1. Illustrative threat surface for Virtual Reality (VR) streaming in a metaverse setting

Recent research on VR security has addressed multiple aspects, including attack identification, privacy risks, and protection mechanisms. A number of studies have focused on emerging attack vectors in immersive environments. For example, previous studies have demonstrated various security and privacy vulnerabilities in AR/VR environments. Virtual keystrokes can be inferred using Wi-Fi channel state information through the VR-Spy attack [5]. Application-level exploitation risks in VR environments have also been reported through the "Man-in-the-Room" attack [9]. In addition, software-based side-channel vulnerabilities have been identified in AR/VR systems [2]. Research has further shown that VR motion data can be exploited to infer more than forty personal attributes, raising significant privacy concerns [10]. To mitigate such risks, differential privacy techniques have been proposed [11]. Furthermore, comprehensive analyses have summarized authentication challenges and broader privacy issues in VR environments [12]. Despite these contributions, most existing works focus on either identifying attacks or proposing isolated privacy and security solutions, without considering their combined impact on system performance. In particular, there is a lack of integrated approaches that quantitatively evaluate how adversarial conditions and security mechanisms jointly influence latency and throughput in real-time VR streaming. This limitation motivates the need for a unified analytical framework that captures the trade-off between security and performance, as proposed in this work.

This paper presents a latency-aware framework for secure VR streaming that explicitly models the trade-off between cybersecurity mechanisms and real-time performance under adversarial conditions. The core contribution is a unified analytical model that captures the impact of attacks and defenses on latency and goodput through the introduction of attack amplification and residual impact factors. In addition, the work integrates standard security mechanisms within a VR streaming architecture and validates the model through a controlled simulation of a 5-minute session, providing practical insights into optimizing security while preserving immersive performance. Unlike conventional networking studies that primarily focus on throughput and latency optimization, this work adopts a security and safety engineering perspective for VR streaming systems operating under adversarial conditions. In immersive environments, latency is not merely a performance metric but a safety-critical parameter, as delays beyond acceptable thresholds can lead to motion sickness, disorientation, and degraded user experience. Therefore, cyberattacks such as DDoS, MitM, and replay attacks are interpreted not only as threats to confidentiality and integrity, but also as indirect safety risks due to their impact on real-time responsiveness. In this context, the proposed analytical model provides a unified framework in which performance degradation reflects risk severity, while security mechanisms represent risk mitigation effectiveness, thereby aligning the study with the principles of safety and security engineering.

This paper is structured as follows. Section 1 presents the related work and background and identifies the key research gaps in the VR domain. Section 2 describes the proposed system model and overall architecture, and details the security design and threat considerations. Section 3 reports the experimental results and provides a security/performance analysis. Section 4 summarizes the five-minute performance snapshot, and Section 5 compares the proposed approach with related studies. Finally, the paper concludes with a summary of findings and directions for future work.

2. SECURITY MODEL

Figure 2 illustrates the main phases of a VR session lifecycle. The process begins with the discovery phase, where the client locates the server via UDP broadcast. This is followed by the handshake phase, during which session parameters and cryptographic capabilities are negotiated. In the initialization phase, configuration parameters and connectivity tests are exchanged to ensure readiness for streaming. Finally, the streaming phase handles the continuous transmission of video frames from server to client, along with control and pose feedback in the reverse direction. Each of these stages represents a potential attack surface, particularly during handshake and streaming.

VR systems threats are discussed, the attack tends to extract user-typed secrets from multi-user VR applications; By extracting those secrets, an attacker may get access to the following types of sensitive data:

a. The use of payment systems in VR applications that store credit card information makes these applications susceptible to unauthorized financial data access.
b. The passwords and authentication data used in VR applications work as protection for private accounts, yet this makes them an attractive target for attackers who try to gain unauthorized access.
c. Private Conversations within multi-user VR expose business and personal sensitive information to attackers who may intercept this confidential information [13-15].

The threat modeling procedure consists of several stages; starting with identifying and understanding the possible threats to a specific system, following that countermeasures must be defined to mitigate the threats. Because the system is vulnerable to attacks, this approach helps evaluate security risks and responses. Its difficult task to expect possible threats and analyze every single part of the system for different security aspects by following the Confidentiality, Integrity, Availability (CIA) model, which provides the basis for researchers to identify certain attacks. The primary objective is to secure communication exchanges [16].

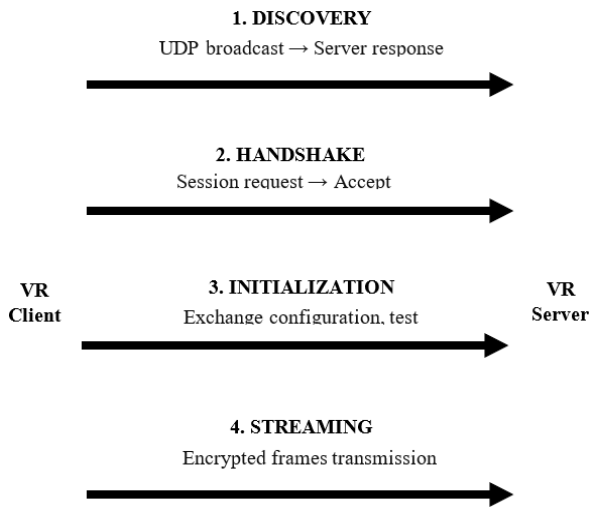

Figure 2. Session lifecycle assumed by the proposed model: discovery, handshake, initialization, and streaming

2.1 The threat model

In this part, we are focusing on attacks perpetrated against VR content delivery. Threats against VR can take many forms and originate from different sources. The results of our survey on the possible attacks against drone swarm [17-25] and the results are summarized in Table 1.

Table 1. Threat model

Attack Name	Description
Man-in-the-Middle (MitM)	An attack on communication pathways can occur when attackers steal information or modify exchanged content between two parties after they fail to detect the breach, which undermines privacy standards and disrupts data integrity.
Distributed Denial of Service (DDoS)	The massive traffic from floods causes VR content servers to slow down or stop their real-time operations.
Replay Attack	An attacker gains unauthorized access to session requests previously recorded as stream start requests for fraudulent activity.
TCP Reset Attack	Attackers can break ongoing connections between two parties by using saved forged TCP RST (reset) packets.
TLS Downgrade Attack (SSL Stripping)	The attacker manipulates connections to make them operate on less secure protocols, thus exposing affected data to interception risks.

Among all security concerns resulting from SSL/TLS misconfiguration, the MitM attack ranks as one of the most common threats. Malicious third parties intercept both client and server communication, which they may modify as well. A hacker intercepting sensitive API data containing usernames, passwords, and other private information will create devastating results, which prove most dangerous for industries such as finance and healthcare due to their strict privacy standards [17, 18].

The DDOS attack mechanism aims to disable VR content delivery platforms by sending continuous waves of traffic from multiple points, thus creating server system downtime, authentication failure, and network delays during service outages. VR demands fast response and large bandwidth for instant deliverables, which makes DDoS attacks significantly reduce performance quality until users experience nausea and dissatisfaction. Cyber attackers take advantage of excessive bandwidth to disable servers and APIs that affect multiplayer server connections and cloud delivery of content. Available solutions for defense against such threats consist of Content Delivery Networks (CDNs) together with rate-limiting technology and AI-driven anomaly detection systems, as well as redundant server infrastructure. Different security measures are implemented to distribute network traffic and protect against failures while safeguarding system availability throughout attacks [18, 19].

The authentication and encryption deficiencies of the SECS/GEM protocol allow attackers to perform replay attacks, which makes them retransmit captured legitimate host-manufacturing equipment messages to trick both parties at any point in time. The attack presents a critical risk to Industry 4 operations because it involves the cybercriminal gaining host privileges and sending fake commands. Request packets to terminate valid sessions and seize control of the network. Time-based validation technologies should be

deployed immediately because they protect against undetected replay attacks [20, 21].

Spoofed ACK packets containing invalid sequence numbers sent to both points of an established TCP connection can be used to perform the TCP RESET attack because of the vulnerability in TCP protocols. The exchange of mismatched ACK packets leads to an unending storm of useless traffic between the connected parties. Such continuous traffic loops cause virtual network systems instability and disconnect to perform Denial-of-Service (DoS). The improper management of unforeseen ACK numbers represents the main vulnerability that lets a low-resource attacker create substantial amplification damage [22].

TLS downgrades occur because attackers abuse the unsecured nature of ClientHello messages exchanged during TLS protocol handshaking. The initial ClientHello message constitutes a vector for attack in SDNs utilizing optional TLS, since attackers can modify it to force both controller and switch to an insecure TLS version and weak cipher suites. Encryption is vulnerable to exposure of critical command data, such as routing rules, before being applied because of the weakened security. An attacker modifies protocol parameters while communication is in progress by lowering the TLS version from 1.2 to 1.0, which drives sessions to adopt insecure encryption principles. The approach creates vulnerabilities despite an organization's deployment of TLS 1.3 because fallback compatibility is typically permitted in systems [23-25].

Figure 3 highlights the critical points where attacks may be injected during the VR session. During the initiation phase, attackers may perform MitM or replay attacks by intercepting or reusing handshake messages. In the streaming phase, threats such as delay injection, packet manipulation, or session hijacking can disrupt real-time communication. These insertion points motivate the need for continuous protection mechanisms across all phases of the session.

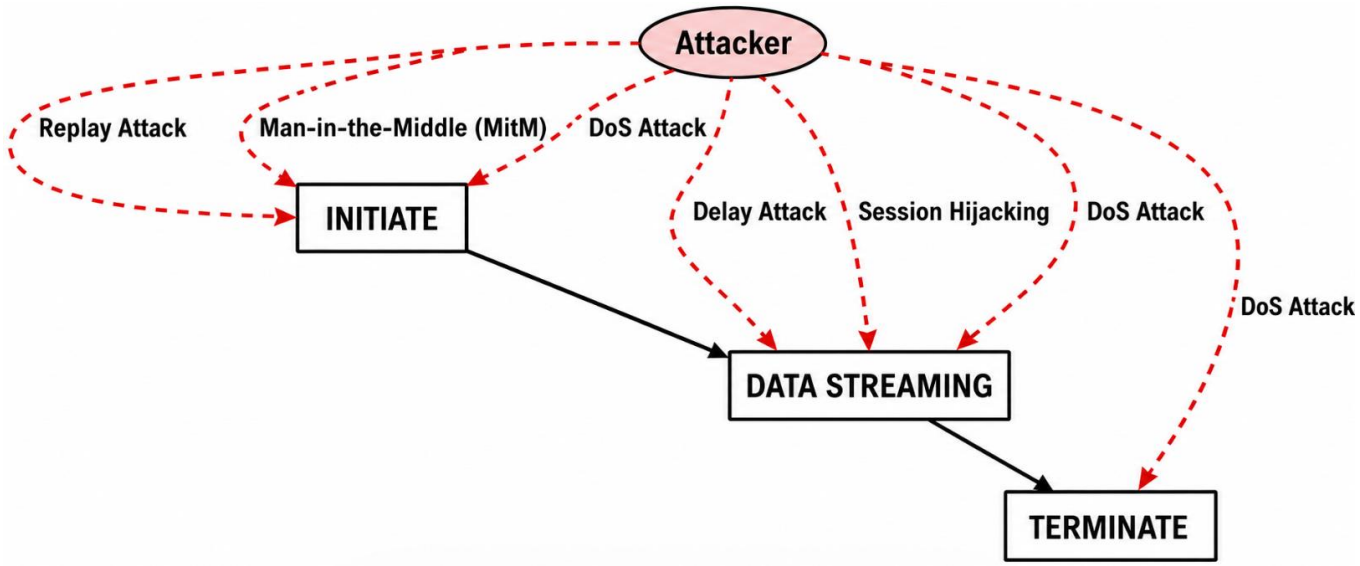

Figure 3. Attack points along Virtual Reality (VR) session phases (initiate, streaming, terminate)

2.2 Security analysis and countermeasures

The proposed network security model must accommodate multiple objectives to protect data validity, along with secrecy and confidentiality, as well as data inalterability. Evaluation of recommended security solutions demonstrates their effectiveness in addressing the predicted threats. No other machinery is at as much risk as VR monitors since their operation depends on fast connections combined with extensive bandwidth, thus making them susceptible to operational failings and system information breaches. The model implements transport-layer security enhancements to provide disruption-free, secure, efficient VR content transmission that protects user anonymity and prevents service breakdowns from unauthorized actors through these security solutions:

- Secure Connection Establishment
- Protection Against Network Flooding
- Low-Latency Optimization
- Authentication & Integrity Verification

After mentioning the possible attacks that could attack our proposed system, along with the network security model that explains in detail the adopted security protocols and features, we can now examine and evaluate the suggested countermeasures against potential threats. Table 2 displays the possible attacks and adopted countermeasures against these attacks [26-29].

Table 2. Suggested security model and countermeasures

Attack Name	Against	Security Countermeasures
MitM Attack	VR Clients & Servers	TLS 1.3, AES-256, mutual auth, and DNSSEC
DDOS	VR Content Servers / Streaming APIs	rate limiting, and cloud DDoS protection
Replay Attack	Session Establishment, Stream Initialization, Retransmission Requests	Use nonces, timestamps, DTLS, and HMAC
TCP Reset Attack	VR Multiplayer Sessions	Use TLS and sequence randomization
TLS Downgrade Attack	VR Authentication and Encryption	Enforce TLS 1.3, HSTS, and disable old TLS

Note: MitM= Man-in-the-Middle; DDOS = Distributed Denial of Service; TLS = transport layer security; VR = Virtual Reality

2.3 The network security model procedures

The delivery of VR content needs high data throughput alongside extremely short latency times for users to perceive a fully immersive experience. A genuine VR experience requires end-to-end latency to remain below 20 ms because higher latencies lead to motion sickness in users. The experience of users is adversely impacted by Cyberattacks, including MitM, DDoS, Replay, and TLS Downgrade attacks, which create network delays and packet loss, together with unauthorized content modification. The two main variables of mathematical security modeling represent attack amplification through λ_residual and defense efficiency through μ [30].

2.3.1 Baseline (unsecured) operation

The basic VR streaming setup applies no security protocols to maintain direct start requests between client and server. A direct start request from the client triggers a server response, which creates a UDP data stream via Wi-Fi with low latency, as shown in Figure 4.

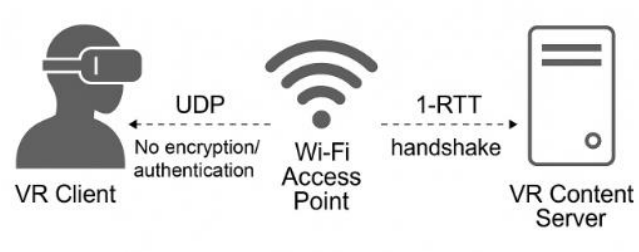

Figure 4. Baseline (unsecured) Virtual Reality (VR) streaming architecture over Wi-Fi: no end-to-end protection for application data

Consumer VR systems tend to choose fast performance over security measures in their standard design. The measured latency stays within 18–21 milliseconds, which satisfies the immersive VR standards (<20 ms) involving only a 1-RTT startup handshake (~30 ms). Dangerous threats, including packet sniffing, session hijacking, and injection attacks, affect the system when it operates this way due to its high vulnerability. The baseline phase does not provide essential security features needed for confidential content transfer despite demonstrating optimal performance conditions, and the equations model latency and throughput primarily as functions of network load, packet size, and transmission delay, as in Eqs. (1) and (2) [14].

$$T_{base} = T_{data} + T_{ack} \quad (1)$$

where, T_{base} is the Baseline (unsecured) one-way/round-trip total latency in seconds. T_{data} is the one way forward for latency. And T_{ack} is the round-trip acknowledgment latency. Baseline throughput (bits/s).

$$R_{base} = \frac{(8\ L_{frame})}{T_{base}} \quad (2)$$

where, L_{frame} is the length of the frame in bits.

2.3.2 Secured under attack

Under the Secured Under Attack scenario, the VR system implements encryption solutions that combine the strength of TLS 1.3 and AES-256 encryption together with HMAC-based authentication, but maintains defense against attacks. Despite their active state, these defense systems can never stop attack traffic from entering the system, where artificial delays are injected. The mathematical model requires the evaluation of two primary factors to express this condition, which are:

Λ_{attack} that describes attack amplification as the factor that measures how much an attack increases latency or network congestion. Systems experience performance degradations when delay or congestion rises above a value of 1 for Λ_{attack}.

The defense efficiency factor $\mu_{defense}$, describes how well security measures decrease attack impact during operations. Partial defense occurs when the efficiency factor ranges from 0 to less than 1. Malicious traffic and encryption overhead cause performance degradation in the system after defenses perform their attack severity reduction role. Figure 5 shows that acceptable VR timings spanning 33 to 40 milliseconds get surpassed while throughput levels drop down to subpar immersive rates between 12 and 15 Mbps. This scenario shows a dementia system that actively defends itself despite being partially exposed to opponent attacks. The system requires strong cryptographical backup together with adaptive security measures, which change protective strategies in response to current attack patterns in real-time [31-33].

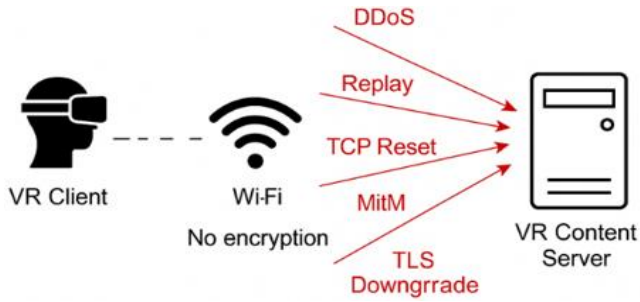

Figure 5. Example of attacks against an unsecured Virtual Reality (VR) streaming section (flooding/DoS, replay, MitM, reset/disrupion)
Note: DoS = Denial-of-Service; MitM= Man-in-the-Middle

Table 3. Attack's negative impact on Virtual Reality (VR) system [34-40]

Attack Name	Attack Mechanism / Execution	Resulting Damage
MitM Attack	Intercepts the UDP handshake, injects fake keys, or modifies handshake parameters.	Data tampering, unauthorized access, and privacy breaches
DDoS Attack	Sends a large volume of UDP packets flooding the server	Latency increase, connection drops
Replay Attack	Captures old "stream start" packets, resends them to the fake session	Session hijacking, duplication
TCP Reset Attack	Sends spoofed TCP-RST packets mimicking legitimate sources	Abrupt session termination, DoS
TLS Downgrade	Alters ClientHello to suggest old TLS versions, weak ciphers	Use of insecure algorithms, potential decryption

An outline of standard attack methods beyond encryption presents data about their negative impact on VR system functionality within Table 3 [16-28].

The described operation demonstrates protective measures that defend the data, yet remain prone to vulnerability because it requires adaptive protection elements to sustain performance quality. The VR system should maintain functionality and security through real-time monitoring as well as traffic shaping and dynamic countermeasures, which can modify its response to current threats in the evolving threat environment.

2.3.3 Secured and defended architecture

Figure 6 presents the secured and defended VR streaming architecture. The system consists of three main functional layers: (i) secure session establishment, where authenticated key exchange is performed; (ii) data protection layer, which applies authenticated encryption (AEAD) and integrity verification to streaming data; and (iii) edge defense mechanisms, including rate limiting and traffic filtering to mitigate flooding attacks. Monitoring components provide real-time feedback to adapt defense strategies. This layered design ensures that confidentiality, integrity, and availability are maintained without significantly compromising latency. The design is compatible with DTLS 1.3 (for datagram-based streaming) or TLS 1.3/QUIC (for encrypted transport with congestion control). The analysis incorporates strong defense mechanisms that minimize the attack effects to a tiny extent. $\lambda_{residual}$ residual scale. Strong cryptographic protocols result in two minor implementation costs, which include processing time delays caused by AES and HMAC overheads, and minimal bandwidth usage reduction. The model includes improved features with $\mu_{defense}$ enhanced as well as methods for security implementation.

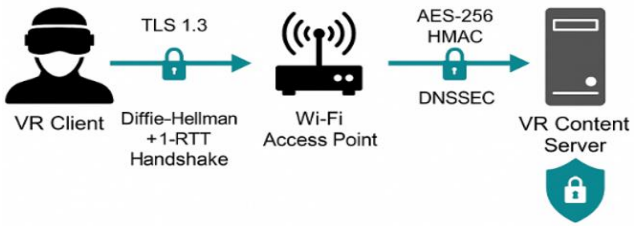

Figure 6. Secured and defended Virtual Reality (VR) streaming: authenticated handshake, applies authenticated encryption (AEAD) encryption, integrity checks, and edge filtering

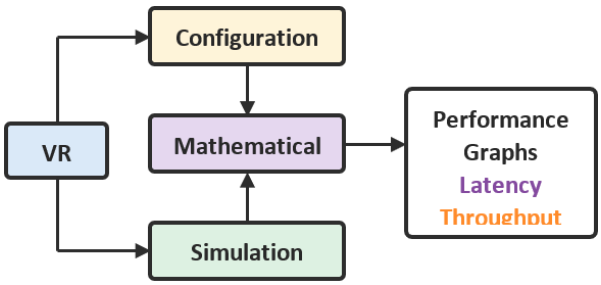

Figure 7. Evaluation workflow: configuration parameters drive the analytical model and the numerical simulation to produce performance curves

The defense components considered in the model are:

(1) Handshake and key establishment: a 1-RTT exchange to derive session keys.
(2) Payload protection: AEAD encryption and integrity verification; anti-replay via sequence numbers and sliding windows.
(3) Edge availability controls: rate limiting and filtering at the AP/server to reduce the effect of flooding and malformed traffic.
(4) Monitoring hooks: per-flow counters and anomaly triggers that can adapt the filtering aggressiveness [41].

Figure 7 illustrates the evaluation workflow that connects the analytical model to the simulation environment. Configuration parameters are first defined and fed into the analytical model to generate expected performance trends. These parameters are then used in the simulation to produce numerical results, enabling validation of the model through comparison with simulated behavior.

3. ANALYTICAL PERFORMANCE MODEL

The analytical model constitutes the core contribution of this work, while the system architecture and simulation are used to contextualize and validate the proposed formulation. This section derives a compact model for frame latency and application goodput under the four operating conditions. All variables are defined with consistent units; time is in milliseconds unless otherwise stated.

3.1 Baseline latency and goodput

Let T_{base} denote the baseline end-to-end frame latency in the unsecured case. We model it as the sum of a data delivery component and a (potential) control/feedback component. For pure UDP streaming, the feedback term may be negligible; for reliable transports or control exchanges, it captures acknowledgement and scheduling effects:

$$T_{base} = T_{data} + T_{ctrl} \tag{3}$$

where T_{data} includes transmission, MAC contention, propagation, and baseline queueing, and T_{ctrl} captures control-plane messages relevant to the session (e.g., acknowledgement of a configuration update).

Baseline application goodput R_{base} is approximated by the delivered payload bits per unit time:

$$R_{base} = \frac{(8\,L_f)}{\frac{T_{base}}{1000}} \tag{4}$$

where L_f is in bytes, T_{base} is in ms, and R_{base} is in bps.

3.2 Attack and security effects

To justify the introduction of the attack amplification factor, we consider the impact of network congestion under adversarial conditions. In VR streaming systems, latency is primarily composed of transmission delay, processing delay, and queueing delay. Under normal conditions, queueing delay remains limited due to controlled traffic load. However, during network attacks such as DDoS flooding or packet injection, the effective traffic load increases significantly, leading to buffer buildup, medium contention (especially in Wi-Fi environments), and retransmission effects. According to classical queueing theory and network congestion models, delay grows non-linearly with load and can be approximated

as a scaled version of baseline latency under moderate-to-high utilization conditions. To capture this effect in a tractable form, we model the impact of attacks using a multiplicative attack amplification factor ($\lambda_{attack} \geq 1$), which scales the latency-critical components of the system. This abstraction is commonly used in network performance modeling to represent congestion-induced delay inflation without requiring full queueing system specification. After applying defense mechanisms such as filtering and rate limiting, the remaining impact of the attack is represented by a reduced residual amplification factor ($\lambda_{res} \geq 1$).

Under flooding or congestion-inducing attacks, the baseline queueing term increases. We represent the extra congestion pressure by an amplification factor $\Lambda_{attack} \geq 1$ applied to the latency-critical portion of the path. In addition, enabling security adds a processing term D_{sec} that accounts for cryptographic operations and defense processing [30, 31]:

$$T_{attack} = (T_{base} + D_{sec}) \cdot \Lambda_{attack} \quad (5)$$

This formulation assumes that the dominant impact of attacks is on queueing and contention delay, while propagation and processing delays remain relatively stable. The multiplicative Λ_{attack} captures the fact that under heavy load, both transmission and queueing delay can scale up due to contention and buffer buildup.

We further decompose the security overhead into cryptographic processing D_{crypto} and edge defense delay D_{def}:

$$D_{\text{sec}} = D_{crypto} + D_{def} \quad (6)$$

$$D_{crypto} = T_{handshake} + T_{enc} + T_{auth} \quad (7)$$

where, $T_{handshake}$ is the amortized cost of session setup (e.g., 1-RTT key exchange distributed over frames), T_{enc} is encryption/decryption time, and T_{auth} is integrity verification (including anti-replay checks).

3.3 Secured-and-defended operation and optimization

In the secured-and-defended case, filtering and rate limiting reduce the residual impact of the attack. We model this by a residual amplification factor $\lambda_{res} \geq 1$ (ideally close to 1) and a defense efficiency factor $\mu \in (0, 1]$ applied to goodput:

$$T_{\text{def}} = (T_{base} + D_{crypto} + D_{def}) \cdot \lambda_{res} \quad (8)$$

$$R_{\text{def}} = R_{base} \cdot \mu \cdot \frac{\left(\frac{T_{base}}{(T_{base} + D_{crypto} + D_{def})}\right)}{\lambda_{res}} \quad (9)$$

The optimized secured configuration reduces D_crypto (e.g., by hardware acceleration, session resumption, or lighter-weight integrity paths) and/or reduces λ_{res} by improving filtering decisions.

3.4 Model parameters and units

Table 4 lists the key parameters used to generate the numerical results. Values represent a Wi-Fi 6 VR streaming profile and are treated as illustrative; the model supports re-parameterization for other deployments.

Table 4. The key parameters used to generate the numerical results

Symbol	Meaning	Unit	Typical Value
T_{base}	Baseline end-to-end latency	ms	10–22
T_{data}	Data delivery latency component	ms	—
T_{ctrl}	Control/feedback latency	ms	—
R_{base}	Baseline application goodput	bps	—
S	Frame size	bytes	150–250 kB
f	Frame rate	frames/s	72–90
D_{sec}	Total security overhead delay	ms	3–10
D_{crypto}	Cryptographic processing delay (encryption + authentication)	ms	2–6
D_{def}	Defense processing delay (filtering, rate limiting)	ms	0–4
$T_{handshake}$	Amortized handshake delay per frame	ms	0–3
T_{enc}	Encryption/decryption delay	ms	1–4
T_{auth}	Authentication and integrity verification delay	ms	1–4
λ_{attack}	Attack amplification factor (latency inflation due to attack)	—	1.1–2.0
λ_{res}	Residual attack amplification after defense	—	1.0–1.2
μ	Goodput efficiency factor under defense	—	0.8–1.0

4. NUMERICAL EVALUATION

To validate the analytical model under realistic conditions, a controlled simulation environment was implemented using Python to emulate a 5-minute VR streaming session. The simulation reflects a Wi-Fi 6-based VR deployment with parameters aligned to typical standalone head-mounted displays. The VR traffic model assumes a continuous stream of compressed video frames with an average frame size of 150–250 kB and a frame rate of 72–90 frames per second, corresponding to a target throughput of approximately 25–50 Mbps. Each frame is segmented into UDP packets with a typical payload size of 1200–1400 bytes to reflect practical MTU constraints. Baseline network conditions assume low packet loss (<1%) and stable channel conditions. Attack scenarios are modeled by injecting additional traffic load to emulate DDoS-style congestion and packet interference. The attack intensity is represented through the attack amplification factor (λ_{attack}), varied in the range of 1.1 to 2.0, reflecting moderate to severe congestion conditions. After applying defense mechanisms, the residual impact is modeled using λ_{res}, typically ranging between 1.0 and 1.2. Security overhead is incorporated based on estimated processing delays for TLS/DTLS operations, including handshake amortization (0–3 ms), encryption/decryption (2–6 ms), and integrity verification (1–4 ms). Defense-related delays, such as filtering and rate limiting, are modeled as an additional 0–4 ms processing cost. The simulation computes per-frame latency and application goodput over time, allowing direct comparison with the analytical model across four configurations: unsecured, under attack, secured and defended, and optimized

secured. These parameters are consistent with the analytical model and enable reproducible evaluation of the latency–security trade-off in VR streaming systems.

We generate latency and goodput curves by evaluating the analytical model across a range of load indices that reflect increasing contention (e.g., more competing flows, larger frames, or higher background traffic). Four configurations are compared:

Unsecured baseline: no end-to-end cryptographic protection.

Under attack: increased congestion pressure ($\Lambda_{attack} > 1$) without effective filtering.

Secured + defended (original): encryption, integrity, anti-replay, and filtering with non-negligible cryptographic cost.

Secured + defended (optimized): reduced cryptographic cost and improved filtering (lower λ_{res}).

Figure 8 reports the modeled end-to-end frame latency. The observed latency trends can be explained by decomposing the total delay into baseline transmission, congestion-induced queueing, and security-related processing overhead. In the under-attack scenario, the increase in latency is primarily driven by the amplification of queueing delay due to increased traffic load, as captured by λ_{attack}. This results in buffer buildup and medium contention, particularly in Wi-Fi environments. In the secured-and-defended configuration, latency increases further due to the addition of cryptographic processing D_{crypto} and defense mechanisms D_{def} However, this increase is controlled and remains within an acceptable range for immersive VR. The optimized secured configuration reduces latency by minimizing D_{crypto} (e.g., through efficient encryption or session reuse) and lowering the residual attack impact λ_{res}. This demonstrates that careful optimization of security mechanisms can significantly improve responsiveness without compromising protection.

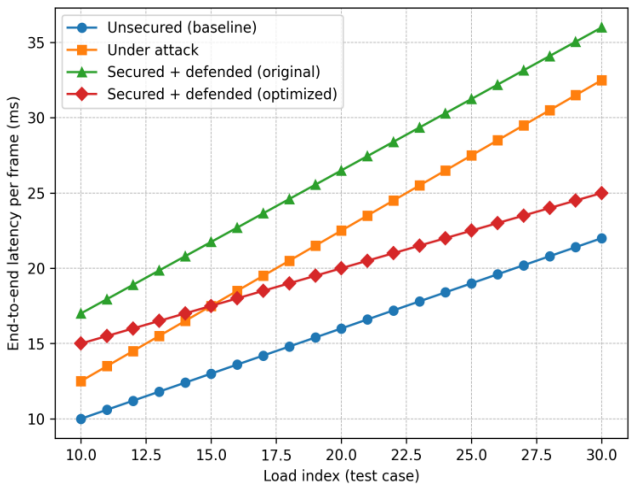

Figure 8. Modeled frame latency under normal conditions and Distributed Denial-of-Service (DDoS) attack scenarios

Figure 9 shows the corresponding application goodput. The goodput behavior reflects the combined effects of bandwidth consumption, congestion, and protocol overhead. Under attack conditions, goodput degradation is mainly caused by excessive competing traffic, which reduces the effective capacity available for VR data transmission. In the secured-and-defended scenario, goodput is further reduced due to additional headers, authentication tags, and processing overhead introduced by security mechanisms. However, the system maintains stability due to effective filtering and rate limiting. The optimized secured configuration improves goodput by reducing unnecessary processing overhead and enhancing traffic filtering efficiency, which increases the effective utilization of available bandwidth. This highlights the importance of balancing security strength with implementation efficiency.

To relate the load-index curves to a short session trace, Figures 10 and 11 provide an illustrative 5-minute snapshot of latency and goodput under time-varying conditions. The same ordering is preserved: attacks create the largest degradation, while the secured design remains stable, and the optimized secured configuration improves both metrics.

The time-based results over the 5-minute session further confirm the stability of the proposed model. While attack scenarios introduce fluctuations due to dynamic congestion, the secured configurations maintain consistent performance. The optimized secured model exhibits lower variance, indicating improved robustness and adaptability under varying network conditions.

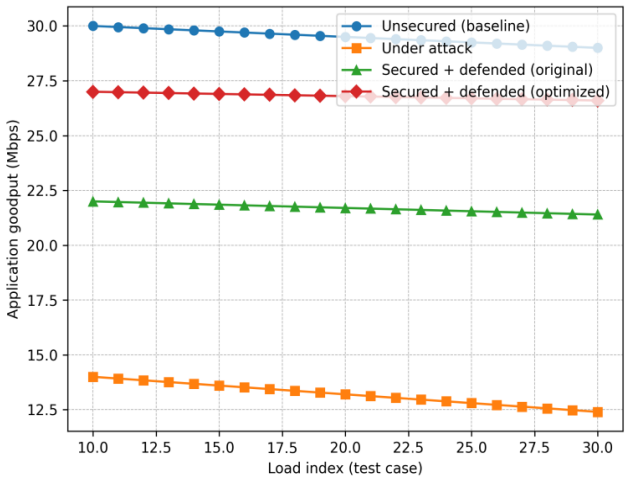

Figure 9. Modeled application goodput versus load index for the same four configurations (generated from Eq. (9))

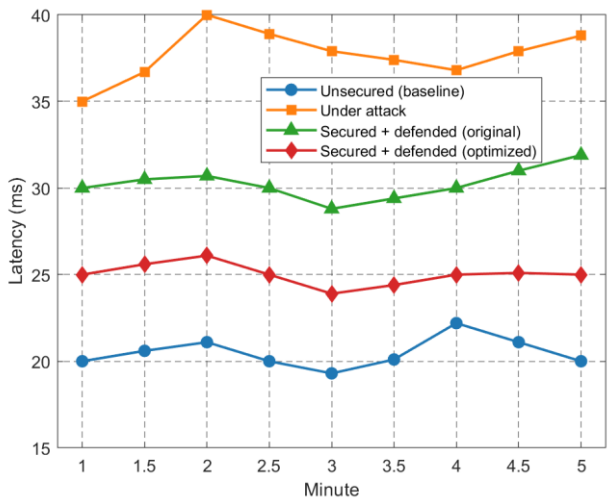

Figure 10. Illustrative minute-level latency evolution over a 5-minute Virtual Reality (VR) session for the four configurations

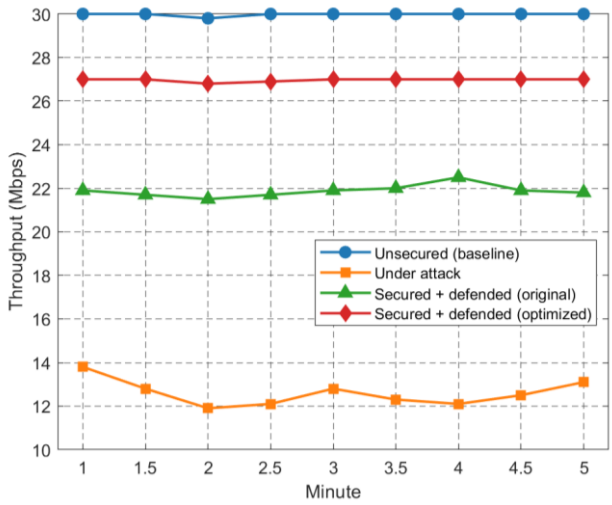

Figure 11. Illustrative minute-level goodput evolution over a 5-minute Virtual Reality (VR) session for the four configurations

To further validate the proposed model under a concrete adversarial condition, we introduce a simulated DDoS attack scenario within the VR streaming environment. In this scenario, the attacker injects a high volume of UDP traffic targeting the VR server, emulating a flooding-based DDoS attack. The injected traffic increases network congestion, leading to buffer buildup, medium contention, and increased packet delays, particularly in the Wi-Fi access network. This behavior is consistent with real-world volumetric DDoS attacks affecting real-time streaming systems. The attack intensity is modeled using the attack amplification factor λ_{attack}, which is increased dynamically from normal conditions ($\lambda_{attack} \approx 1.0$) to severe congestion levels (λ_{attack} up to 2.0). This reflects the transition from normal operation to moderate and high-intensity attack conditions. Figure 8 illustrates the impact of the DDoS attack on end-to-end latency. Under attack conditions, latency increases significantly due to amplified queueing delay and channel contention. The unsecured system shows the highest degradation, exceeding acceptable VR latency thresholds.

When security mechanisms are enabled, including TLS/DTLS encryption, integrity verification, and rate limiting, the system demonstrates improved resilience. Although latency increases due to cryptographic overhead, the secured-and-defended configuration effectively limits attack-induced congestion, resulting in more stable performance. The optimized secured configuration further reduces latency by minimizing cryptographic processing overhead and improving filtering efficiency, which lowers the residual attack amplification factor λ_{res}. As a result, the system maintains latency closer to immersive VR requirements even under attack conditions.

Similarly, Figure 9 shows the impact of the DDoS attack on application goodput. The attack reduces available bandwidth due to excessive competing traffic, while the secured configurations maintain higher goodput by filtering malicious traffic and preserving legitimate data flow. These results confirm that the proposed analytical model accurately captures the behavior of VR streaming systems under DDoS attacks and demonstrates the effectiveness of the integrated security mechanisms.

As shown in Table 5, existing works primarily address specific attack vectors or privacy concerns without integrating them into a performance-aware system model. None of the reviewed approaches provide a quantitative framework that captures how security mechanisms affect latency and throughput under adversarial conditions which was covered in this work.

Table 5. Comparison with the related works

Work	Focus	Method	Considers Latency	Considers Attacks	Quantitative Model	Security–Performance Trade-Off
Al Arafat et al. [5]	Side-channel attack (VR-Spy)	CSI-based detection	NO	YES	NO	NO
Vondráček et al. [23]	VR malware (MitR)	Experimental	NO	YES	NO	NO
Zhang et al. [2]	Side-channel leakage	Experimental	NO	YES	NO	NO
Garrido et al. [11]	Privacy leakage	ML inference	NO	Indirect	NO	NO
Sun et al. [10]	Differential privacy	Statistical	NO	NO	NO	NO
Giaretta [12]	Survey	Literature review	NO	Indirect	NO	NO
This Work	Secure VR streaming under attack	Analytical + Simulation	YES	YES	YES	YES

5. COMPARISION WITH PREVIOUSE WORKS

While prior research has extensively explored security and privacy challenges in VR systems, most existing approaches focus on isolated aspects, such as attack detection, privacy leakage mitigation, or application-level vulnerabilities. These works typically do not consider the joint impact of security mechanisms and adversarial conditions on real-time system performance, which is critical for immersive VR applications. In contrast, the proposed work introduces a unified analytical framework that explicitly models the relationship between security enforcement and performance degradation, particularly in terms of latency and application goodput. This enables a quantitative understanding of how cyberattacks and defense mechanisms interact in real-time VR streaming environments. To highlight this distinction, Table 5 provides a structured comparison between the proposed approach and representative state-of-the-art studies.

6. CONCLUSION

This paper presented a secured and defended VR streaming architecture along with a unified analytical model for

evaluating frame latency and application goodput under adversarial conditions. By decomposing system delay into baseline transmission, cryptographic overhead, and residual attack impact, the proposed model provides a clear framework for understanding the trade-off between security and real-time performance in immersive environments. The results demonstrate that while cyberattacks such as DDoS and MitM significantly degrade system performance, the integration of appropriate security mechanisms—including TLS/DTLS, authenticated encryption, and traffic filtering—can effectively mitigate these effects while maintaining acceptable latency for VR applications. Furthermore, the optimized configuration shows that careful reduction of cryptographic and defense overhead can improve both latency and goodput without compromising security. From a practical perspective, the proposed framework can be used as a design and evaluation tool for real-world VR streaming systems, particularly in Wi-Fi 6 and edge-based deployments. System designers can leverage the model to estimate the impact of different security configurations, select appropriate protection mechanisms, and ensure that immersive performance requirements are met even under adversarial conditions. In addition, the model can support the development of adaptive security strategies, where protection levels are dynamically adjusted based on network conditions and detected threats. The findings of this work contribute toward enabling secure and responsive VR applications in emerging domains such as remote collaboration, healthcare, and metaverse platforms. Future work will focus on extending the model to incorporate more complex network environments, integrating machine learning-based attack detection, and validating the framework through real-world experimental testbeds.

REFERENCES

[1] Van Damme, S., Sameri, J., Schwarzmann, S., Wei, Q., Trivisonno, R., De Turck, F., Torres Vega, M. (2024). Impact of latency on QoE, performance, and collaboration in interactive multi-user virtual reality. Applied Sciences, 14(6): 2290. https://doi.org/10.3390/app14062290

[2] Zhang, Y., Slocum, C., Chen, J., Abu-Ghazaleh, N. (2023). It's all in your head (set): Side-channel attacks on AR/VR systems. In 32nd USENIX Security Symposium (USENIX Security 23), pp. 3979-3996. https://www.usenix.org/conference/usenixsecurity23/presentation/zhang-yicheng.

[3] Pooyandeh, M., Han, K.J., Sohn, I. (2022). Cybersecurity in the AI-based metaverse: A survey. Applied Sciences, 12(24): 12993. https://doi.org/10.3390/app122412993

[4] Mohamad, U.H., Ahmad, M.N., Benferdia, Y., Shapi'i, A., Bajuri, M.Y. (2021). An overview of ontologies in virtual reality-based training for healthcare domain. Frontiers in Medicine, 8: 698855. https://doi.org/10.3389/fmed.2021.698855

[5] Al Arafat, A., Guo, Z., Awad, A. (2021). Vr-spy: A side-channel attack on virtual key-logging in vr headsets. In 2021 IEEE Virtual Reality and 3D User Interfaces (VR), Lisboa, Portugal, pp. 564-572. https://doi.org/10.1109/VR50410.2021.00081

[6] Yang, Z., Li, C.Y., Bhalla, A., Zhao, B.Y., Zheng, H. (2024). Inception attacks: Immersive hijacking in virtual reality systems. arXiv preprint arXiv:2403.05721. https://doi.org/10.48550/arXiv.2403.05721

[7] Lohr, D., Aziz, S., Friedman, L., Komogortsev, O.V. (2023). GazeBaseVR, a large-scale, longitudinal, binocular eye-tracking dataset collected in virtual reality. Scientific Data, 10(1): 177. https://doi.org/10.1038/s41597-023-02075-5

[8] Lee, J., Kim, H., Lee, K. (2023). VRKeyLogger: Virtual keystroke inference attack via eavesdropping controller usage pattern in WebVR. Computers & Security, 134: 103461. https://doi.org/10.1016/j.cose.2023.103461

[9] Miller, M.R., Herrera, F., Jun, H., Landay, J.A., Bailenson, J.N. (2020). Personal identifiability of user tracking data during observation of 360-degree VR video. Scientific Reports, 10(1): 17404. https://doi.org/10.1038/s41598-020-74486-y

[10] Garrido, G.M., Nair, V., Song, D. (2023). Sok: Data privacy in virtual reality. arXiv preprint arXiv:2301.05940. https://doi.org/10.48550/arXiv.2301.05940

[11] Sun, R., Wang, H., Chen, H.T., Xue, M. (2024). Privacy in motion: Implementing differential privacy for user motion in VR. In Proceedings of the 36th Australasian Conference on Human-Computer Interaction, pp. 223-231. https://doi.org/10.1145/3726986.3727017

[12] Giaretta, A. (2024). Security and privacy in virtual reality: A literature survey. Virtual Reality, 29(1): 10. https://doi.org/10.1007/s10055-024-01079-9

[13] AlShekh, R.H., Ali, Q. I. (2025). Forecasting global network traffic trends: The role of virtual reality. arXiv preprint arXiv:2502.00785. https://doi.org/10.48550/arXiv.2502.00785

[14] Wu, Y., Shi, C., Zhang, T., Walker, P., Liu, J., Saxena, N., Chen, Y. (2023). Privacy leakage via unrestricted motion-position sensors in the age of virtual reality: A study of snooping typed input on virtual keyboards. In 2023 IEEE Symposium on Security and Privacy (SP), Francisco, CA, USA, pp. 3382-3398. https://doi.org/10.1109/SP46215.2023.10179301

[15] Jahan, F., Sun, W., Niyaz, Q., Alam, M. (2019). Security modeling of autonomous systems: A survey. ACM Computing Surveys (CSUR), 52(5): 1-34. https://doi.org/10.1145/3337791

[16] Casey, P., Baggili, I., Yarramreddy, A. (2019). Immersive virtual reality attacks and the human joystick. IEEE Transactions on Dependable and Secure Computing, 18(2): 550-562. https://doi.org/10.1109/TDSC.2019.2907942

[17] Alwazzeh, M., Karaman, S., Shamma, M.N. (2020). Man in the middle attacks against SSL/TLS: Mitigation and defeat. Journal of Cyber Security and Mobility, 9(3): 449-468. https://doi.org/10.13052/jcsm2245-1439.933

[18] Khan, K. (2023). Securing immersive realms: A review of security measures in 360-degree virtual reality video streaming. International Journal of Science and Research, 12(12): 727-735. https://doi.org/10.21275/SR231208045505

[19] Ling, Z., Li, Z., Chen, C., Luo, J., Yu, W., Fu, X. (2019). I know what you enter on gear VR. In 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, pp. 241-249. https://doi.org/10.1109/CNS.2019.8802674

[20] Al-Shareeda, M.A., Manickam, S., Laghari, S.A., Jaisan, A. (2022). Replay-attack detection and prevention mechanism in industry 4.0 landscape for secure

SECS/GEM communications. Sustainability, 14(23): 15900. https://doi.org/10.3390/su142315900
[21] Nguyen, S.D., Mimura, M., Tanaka, H. (2017). Leveraging man-in-the-middle DoS attack with internal TCP retransmissions in virtual network. In International Conference on Information Systems Security, pp. 367-386. https://doi.org/10.1007/978-3-319-72598-7_2
[22] Kim, J., Seo, M., Lee, S., Nam, J., et al. (2024). Enhancing security in SDN: Systematizing attacks and defenses from a penetration perspective. Computer Networks, 241: 110203. https://doi.org/10.1016/j.comnet.2024.110203
[23] Vondráček, M., Baggili, I., Casey, P., Mekni, M. (2023). Rise of the metaverse's immersive virtual reality malware and the man-in-the-room attack & defenses. Computers & Security, 127: 102923. https://doi.org/10.1016/j.cose.2022.102923
[24] El-Hajj, M. (2024). Cybersecurity and privacy challenges in extended reality: Threats, solutions, and risk mitigation strategies. Virtual Worlds, 4(1): 1. https://doi.org/10.3390/virtualworlds4010001
[25] Van Maris, A., Zook, N., Caleb-Solly, P., Studley, M., Winfield, A., Dogramadzi, S. (2020). Designing ethical social robots—A longitudinal field study with older adults. Frontiers in Robotics and AI, 7: 1. https://doi.org/10.3389/frobt.2020.00001
[26] Nookala, G. (2024). The role of SSL/TLS in securing API communications: Strategies for effective implementation. Journal of Computing and Information Technology, 4(1): 1-8.
[27] Sheffer, Y., Holz, R., Saint-Andre, P. (2015). Summarizing known attacks on transport layer security (TLS) and datagram TLS (DTLS) (No. RFC7457). IETF Datatracker.
[28] McKay, K., Cooper, D. (2017). Guidelines for the selection, configuration, and use of transport layer security (TLS) implementations (No. NIST Special Publication (SP) 800-52 Rev. 2 (Draft)). National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/draft.
[29] Geris, A., Cukurbasi, B., Kilinc, M., Teke, O. (2024). Balancing performance and comfort in virtual reality: A study of FPS, latency, and batch values. Software: Practice and Experience, 54(12): 2336-2348. https://doi.org/10.1002/spe.3356
[30] Zhou, J., Fu, W., Hu, W., Sun, Z., He, T., Zhang, Z. (2024). Challenges and advances in analyzing TLS 1.3-encrypted traffic: A comprehensive survey. Electronics, 13(20): 4000. https://doi.org/10.3390/electronics13204000
[31] Hazarika, A., Rahmati, M. (2023). Towards an evolved immersive experience: Exploring 5G-and beyond-enabled ultra-low-latency communications for augmented and virtual reality. Sensors, 23(7): 3682. https://doi.org/10.3390/s23073682
[32] Harba, E.S.I. (2017). Secure data encryption through a combination of AES, RSA and HMAC. Engineering, Technology & Applied Science Research, 7(4): 1781-1785. https://doi.org/10.48084/etasr.1272
[33] Barker, E., Chen, L., Keller, S., Roginsky, A., Vassilev, A., Davis, R. (2017). Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography (No. NIST Special Publication (SP) 800-56A Rev. 3 (Draft)). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-56Ar3
[34] Turner, J.M. (2008). The keyed-hash message authentication code (HMAC). Federal Information Processing Standards Publication, 198(1): 1-13. https://csrc.nist.gov/files/pubs/fips/198-1/final/docs/fips-198-1_final.pdf.
[35] Raaen, K., Kjellmo, I. (2015). Measuring latency in virtual reality systems. In International conference on entertainment computing, pp. 457-462. https://doi.org/10.1007/978-3-319-24589-8_40
[36] Abdulkareem, O.A., Kontham, R.K., Mahmood, F.E. (2024). Securing smart grids: Machine learning-driven ensemble intrusion detection for IoT RPL networks. International Journal of Safety & Security Engineering, 14(5):1517-1525. https://doi.org/10.18280/ijsse.140519
[37] Alsharbaty, F.S., Ali, Q.I. (2024). Smart electrical substation cybersecurity model based on WPA3 and cooperative hybrid intrusion detection system (CHIDS). Smart Grids and Sustainable Energy, 9(1): 11. https://doi.org/10.1007/s40866-024-00192-7
[38] Qaddoori, S.L., Ali, Q.I. (2023). An embedded intrusion detection and prevention system for home area networks in advanced metering infrastructure. IET Information Security, 17(3): 315-334. https://doi.org/10.1049/ise2.12097
[39] Merza, M.E., Hussein, S.H., Ali, Q.I. (2023). Identification scheme of false data injection attack based on deep learning algorithms for smart grids. Indonesian Journal of Electrical Engineering and Computer Science, 30(1): 219-228. https://doi.org/10.11591/ijeecs.v30.i1.pp219-228
[40] Ali, Q.I., Lazim, S. (2012). Design and implementation of an embedded intrusion detection system for wireless applications. IET Information Security, 6(3): 171-182. https://doi.org/10.1049/iet-ifs.2011.0152
[41] Abdulkareem, O.A., Kontham, R.K., Mahmood, F.E. (2024). Collaborative Intrusion detection system to identify joint attacks in routing protocol for low-power and lossy networks routing protocol on the internet of everything. Mesopotamian Journal of CyberSecurity, 4(3): 251-277. https://doi.org/10.58496/MJCS/2024/026