CS8395 Security & Privacy in Pervasive Environments

VANDERBILT UNIVERSITY

Privacy Leakage via Unrestricted Motion-Position Sensors in VR: Snooping Typed Input on Virtual Keyboards

by Yi Wu et al. (IEEE S&P 2023)

Haoli Yin

Roadmap

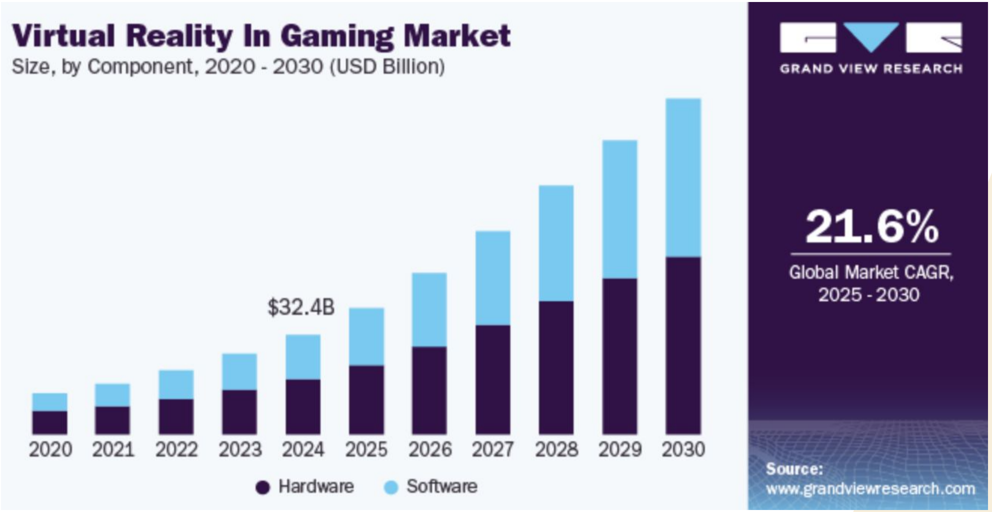
- Introduction
- Key Ideas
- Methods
- Experimentation
- Discussion
- Limitations
- Countermeasures

Introduction

- Background
- Motivation
- Related Work
- Approach Overview

Background

- **VR is Growing** [1]
- VR Devices have:
  - motion, position, orientation sensors
  - Camera, mic, headset, controllers
- Sensitive Inputs in VR
  - Typing passwords
- Privacy Risk
  - Sensors can encode private info



[1] https://www.grandviewresearch.com/industry-analysis/virtual-reality-in-gaming-market

Motivation - Privacy Vulnerabilities in VR

- **Unrestricted Sensors**
  - Most VR sensor data requires no user permission on current platforms (OpenVR, Oculus, WebXR)
- **Smartphones Case Study**
  - Similar “zero-permission” sensor attacks were known on phones (e.g. accelerometer used to infer keystrokes) [2]
- **Attack Scenario**
  - Imagine you log into a VR app and type a password on a virtual keyboard
  - How would you feel if a malicious process was reading your every movement?
  - Could it decipher what you typed?

[2] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Ying Zhang. Accessory: password inference using accelerometers on smartphones. In HotMobile ’12, 2012

Video Demo

https://youtu.be/xaXDmjhTTTc?si=mn9d2BjNpmjr7ikz

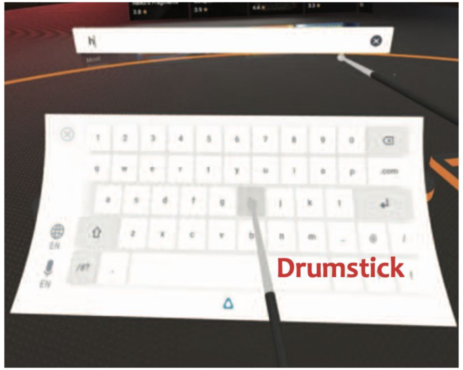
Related Works (and their drawbacks)

- User Authentication [3] → • Narrow focus, leaves sensor data insecure
- Smartphone VR keyboard detection (Samsung Gear) [4] → • Strong assumption of fixed controller rotation between keys

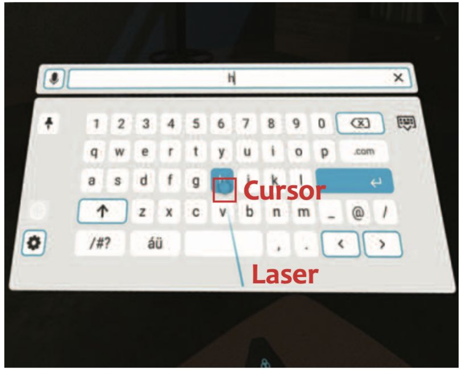
[3] Markus Funk, Karola Marky, Iori Mizutani, Mareike Kritzler, Simon Mayer, and Florian Michahelles. LookUnlock: Using Spatial-Targets for User-Authentication on HMDs. In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI EA'19, pages 1–6. Association for Computing Machinery, 2019.
[4] Zhen Ling, Zupei Li, Chen Chen, Junzhou Luo, Wei Yu, and Xinwen Fu. I know what you enter on gear vr. In 2019 IEEE Conference on Communications and Network Security (CNS), pages 241–249. IEEE, 2019

Approach

- Assumptions:
  - Keyboard Layout (QWERTY)
  - Controller Typing Mechanism
- Deciphers keyboard inputs from sensor data



(a) Drum-based typing



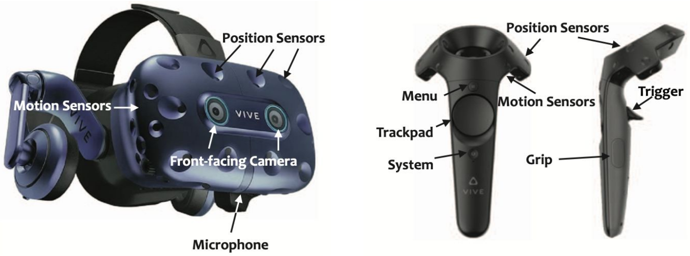
(b) Laser-based typing

Key Ideas

- Threat Model
- Attack Overview

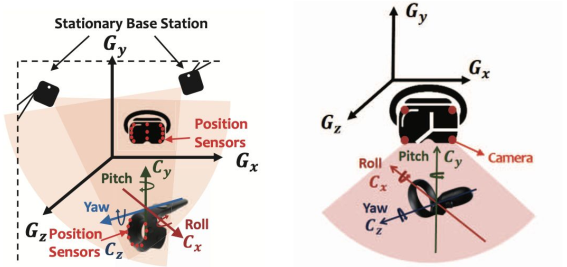
Threat Model

- **Attacker's Method**: Deploy malware or malicious webpage to continuously capture sensor streams (motion, orientation, controller button states)
- **Limited Knowledge Assumption**: The attacker does **NOT** know the user's VR setup or environment (e.g. unknown keyboard app, unknown room layout)
- **Goal**: Infer the sensitive text the user types (passwords or messages) purely from sensor data
- **Realism**: Tested on two popular VR systems (HTC Vive Pro and Oculus Quest)



(a) Sensors on the headset (b) Sensors on the controller

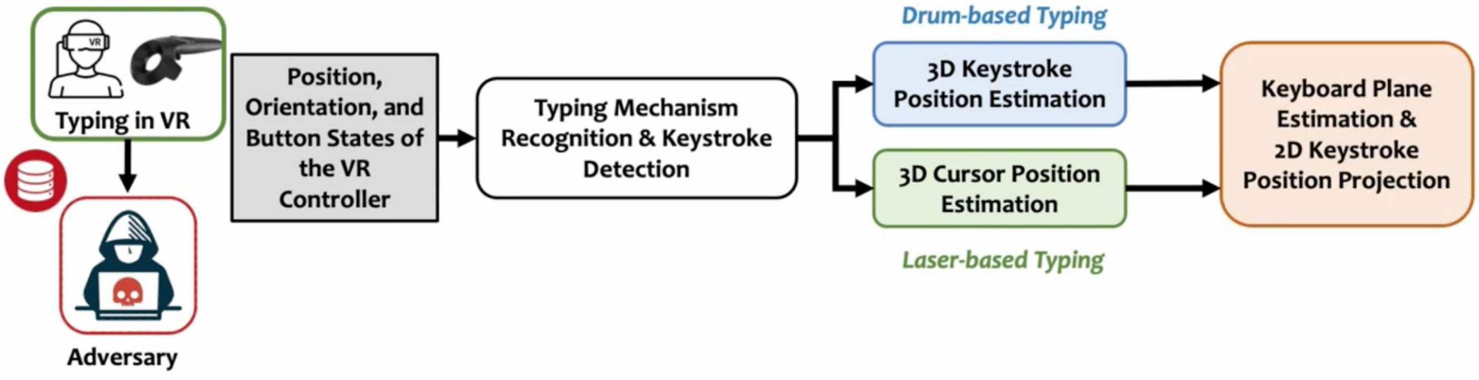
Fig. 1. Sensors in a VR system (i.e., HTC Vive Pro).



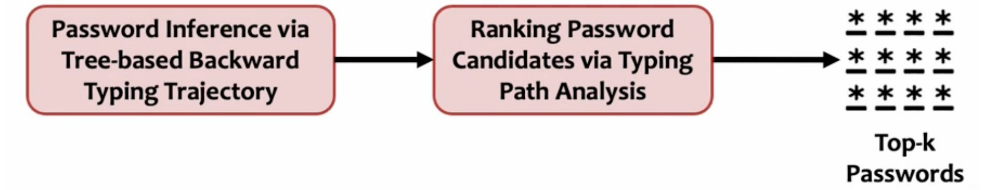
(a) Outside-in tracking (HTC Vive Pro) (b) Inside-out tracking (Oculus Quest)

Fig. 4. Position tracking & coordinate systems in VR.

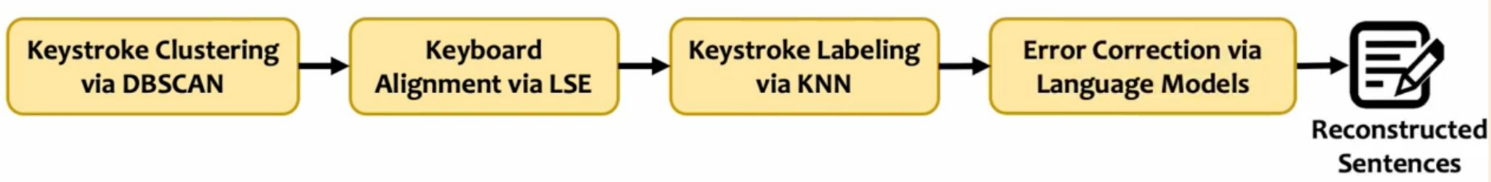
Attack Overview



Inferring Passwords



Inferring Sentences

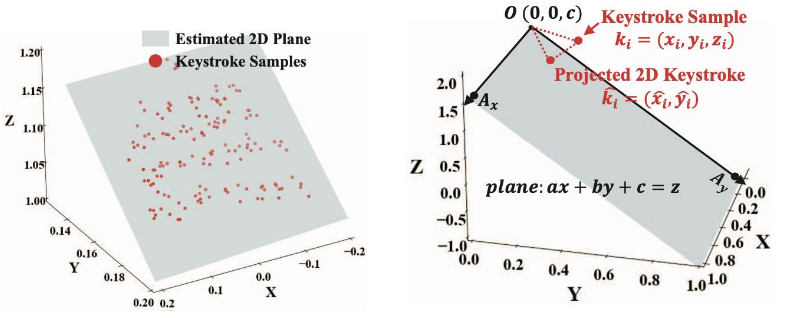


Methods

- Keystroke Position Estimation
- Inferring Passwords
- Inferring Sentences

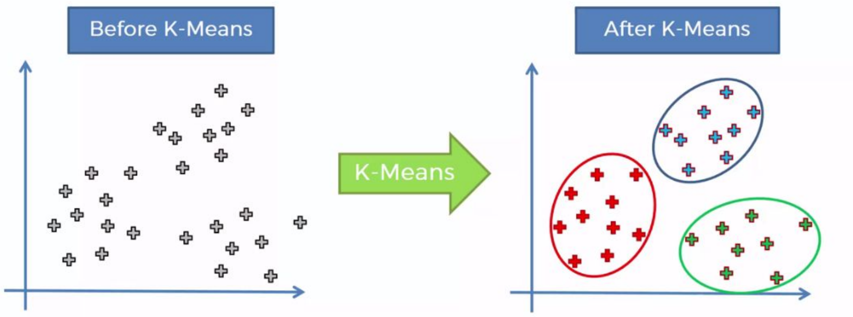
Keystroke Position Estimation

- **Goal**: Determine *which key* was hit by figuring out where the controller was at each keystroke
- **3D to 2D Projection**:
  - Each detected keystroke has a 3D position (controller coordinates)
  - We know what a QWERTY keyboard layout → virtual keyboard
  - Using a least-squares plane fit, project points onto that virtual keyboard plane
  - ✅ 2D coordinates relative to the keyboard surface.
- K-means cluster → use centroids as key mappings for calibrated virtual keyboard
- We know sequence of key presses based off timestamps



(a) Keyboard plane estimation (b) 2D keystroke position projection

Fig. 8. Illustration of Keyboard Plane Estimation & 3D-to-2D Projection.



Inferring Passwords

- **Goal**: Decipher password Input (random characters). Passwords lack context (no dictionary words), so approach is brute-force guided by geometry.
- **Tree-Based Backwards Typing Trajectory → Predict multiple password options**
  - “Enter” key will always be at the end of sequence (and serves as root of tree)
  - Recursively calculate which key could precede it based on the distance to other keys
  - Yields a set of *likely password candidates* best matching hand motions
- **Ranking Password Candidate**
  - Only using distance can be ambiguous since there are multiple likely candidates
  - Leverage directional info in trajectory analysis of similarity to other candidates to rank
  - The top-ranked candidates (e.g., top 3) are used as guesses

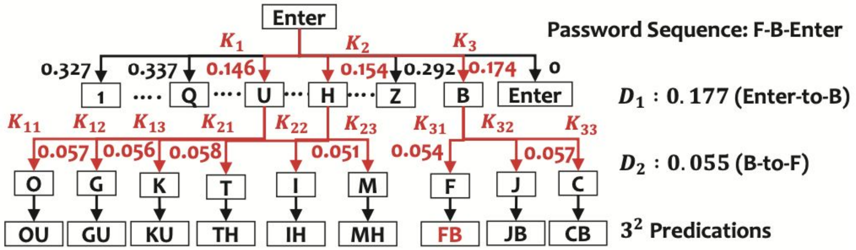


Fig. 9. Tree-based Backward Typing Trajectory Estimation for Password Recovery.

Inferring Sentences

- **Goal**: Reconstruct and label natural language keystrokes
- **Cluster Keystrokes:**
  - Use DBSCAN (density-based clustering) to group 2D keystroke points based on spatial proximity. Use minimum 2 instances/cluster & distance threshold 0.03 units
- **Align Keyboards:**
  - Apply Least Squares Estimation (LSE) to align victim's keyboard with virtual one
  - Randomly select n keys (matching DBSCAN clusters) and solve transformation matrix
- **Label Keystrokes and Correct Errors:**
  - Use K-Nearest Neighbors (K=1) to classify each keystroke by mapping to reconstructed keyboard
    - Example: if a victim's keystroke is closest to the "T" centroid on the attacker's layout, that keystroke is labeled as "T."
  - Refine natural language output grammar correct (e.g. Google Docs Spell Check)

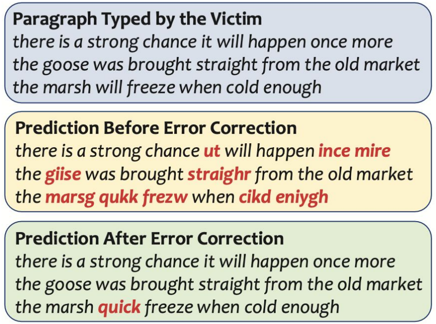


Fig. 16. Examples of Recovered Paragraph.

Experimentation

- Experiment Setup
- Metrics
- Results

Experiment Setup

Study settings:

- 2 systems: HTC Vive Pro (outside-in system) and Oculus Quest (inside-out system)
- 7 participants for each system (14 total)
- 38 keys (26 alpha, 10 num, space key, enter key)
- Used both drum-based and laser-based systems

Simulation Setup:

- Randomly generate passwords of {4, 6, 8} characters
- Randomly selected 10 sentences from Harvard sentences dataset [5]

[5] EH Rothauser. Ieee recommended practice for speech quality measurements. IEEE Trans. on Audio and Electroacoustics, 17:225–246, 1969.

Metrics

- For single keystroke/**character classification**: Accuracy, Precision, Recall
- For Password Inference Metrics:
  - **Top-k Success Rate**: fraction of trials that victim’s password was successfully recovered among the top-k candidate predictions
- For Paragraph Inference Metrics:
  - **Word Recognition Rate** (WRR) = correct words / total words

Results (with brevity)

- **Keystroke Recognition**:
  - The attack can recognize over 89.7% of keystrokes correctly overall
- **Password Recovery**:
  - For random passwords (length 4–8 characters), the attacker's top-3 guesses contain the correct password about 84.9% of the time
  - Even with just a single guess (top-1), success rates were significant (around 50–75% depending on length)
- **Sentence Recovery**:
  - For natural language input, the attacker achieves an average 87.1% Word Recognition Rate
  - Most words in a sentence are reconstructed correctly, often with minor spelling errors. After language-model corrections, many sentences are almost fully readable.
- **Drum vs. Laser**:
  - Drum-based typing had slightly higher accuracy than laser-based in experiments
  - Per-key recognition averaged ~91.7% on drum vs ~81.1% on laser in one test
  - Drum keystrokes produce bigger motion signals, making them a bit easier to classify, but both methods were vulnerable.

Discussion

- Limitations
- Privacy Implications
- Countermeasures

Limitations

1. Have to assume QWERTY layout
2. Environmental variability (controller noise in real environments)
3. Not many participants (k=14), conclusive power is low

Privacy Implications

- **New Side-Channel Threat**:
  - immersive VR isn't purely visual – it's leaking data
  - An attacker doesn't need to "see" your VR screen; motion sensors suffice to know what you type.
- **User Trust & Awareness**:
  - VR users today likely assume typing in a virtual environment is secure (no one looking over your shoulder).
  - But a background app or website could be "shoulder surfing" via sensors. This is an unseen risk – literally invisible to the user.
- **Urgency for Solutions**:
  - These findings put pressure on VR platform providers (like Meta/Oculus, HTC, Valve) to re-evaluate sensor policies.

Countermeasures - Protecting VR Users

- **Restrict Sensor Access:**
  - Implement permissions for motion/position sensors similar to camera or mic
- **UI Indicators**:
  - Hardware or software indicators (lights or on-screen icons) on VR devices to notify the user when sensors are being recorded
  - Similar to webcam LED on computers
- **Anomaly Detection**:
  - VR anti-malware tools could try to detect suspicious sensor logging
  - If an app that shouldn’t need your motion data is constantly polling it, that might be flagged

Questions?