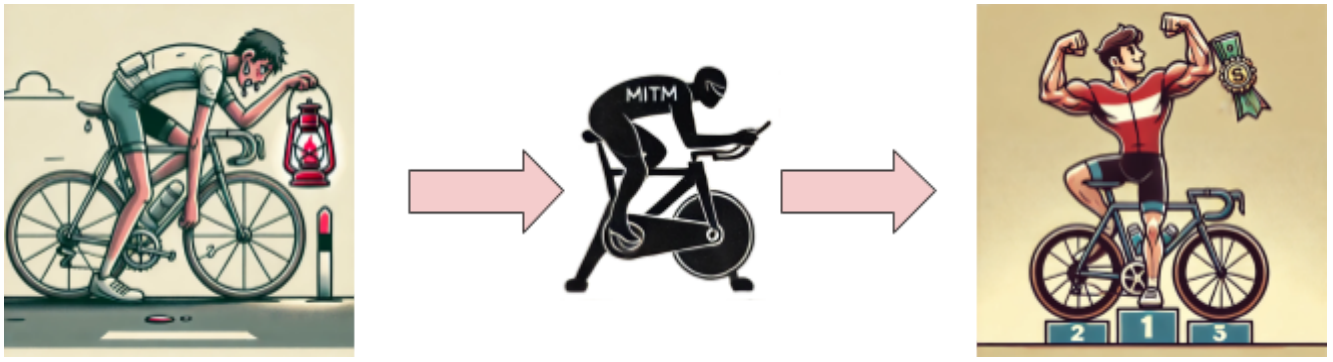
Man-in-the-Middle Attacks on Bluetooth Communication in Virtual Cycling Systems



v1.0

18. Feb. 2025

Bastian Wegener

https://www.f-e-r-a.org

FAIR E-RACING ALLIANCE

1. Introduction

Bluetooth Low Energy (BLE)[1] is a widely used wireless protocol for connecting **cycling trainers** with **virtual riding platforms** or **cycling computers**. BLE enables seamless **bidirectional communication**, allowing cycling software to receive **real-time power metrics** from the trainer while sending **control signals** such as incline adjustments and rider weight settings. This interaction is essential for creating an immersive and realistic indoor cycling experience.

Popular Virtual Cycling Platforms

- **Zwift**[2] – A leading virtual cycling platform offering **structured workouts, competitive races, and an interactive virtual world**.
- **MyWhoosh**[3] – Provides **real-time feedback, structured training plans, and a dynamic virtual environment** for indoor cycling.
- **TrainingPeaks Virtual**[4] **(formerly IndieVelo)** – A dedicated **e-cycling platform** designed for **training, racing, and performance analysis** in a virtual setting.
- **Rouvy**[5] – Combines **virtual racing with real-world route simulations**, allowing cyclists to ride on **lifelike terrains with augmented reality elements**.

Importance of Accurate Power and Control Data

- **Power Output Accuracy** – Competitive virtual cycling platforms rely on **precise power measurements** to ensure fair play. Manipulated power data can **distort race outcomes and compromise competition integrity**.
- **Control Signal Reliability** – Virtual cycling platforms **adjust trainer resistance based on terrain changes**. If these signals are tampered with, riders may **experience incorrect resistance levels**, potentially leading to **ineffective training or safety risks**.

Compromising this data can have significant real-world consequences, ranging from unfair advantages in competitive races to potential safety risks for riders.

Security Risks: BLE Vulnerabilities & MitM Attacks

While BLE offers a convenient and efficient way to connect devices, it is vulnerable to **Man-in-the-Middle (MitM) attacks**. A malicious actor can exploit security weaknesses to **intercept and manipulate data** in transit. Common exploits include:

- **Power Value Manipulation** – Falsifying power output data to gain an unfair advantage in races or training sessions.
- **Control Signal Tampering** – Altering incline, resistance, or weight values to disrupt workouts or influence race dynamics.

Although this paper primarily examines **MitM attacks on cycling trainers**, similar Bluetooth **security vulnerabilities exist in other fitness devices**, such as **heart rate straps**, **cadence sensors**, and **power meters**. These devices also **rely on BLE for real-time data** transmission and could be susceptible to **data interception or spoofing** if security measures are weak.

Scope of the Paper

This paper examines how **MitM attacks** can be used to **spoof power values from the trainer to the cycling software** and **modify control signals such as resistance, incline and weight from the software to the trainer**. We will explore:

1. **BLE communication mechanisms**
2. **Potential attack methods**
3. **Security countermeasures to mitigate these risks**

2. Bluetooth Communication in Cycling Trainers

2.1 Overview of Bluetooth Low Energy (BLE) in Cycling Trainers

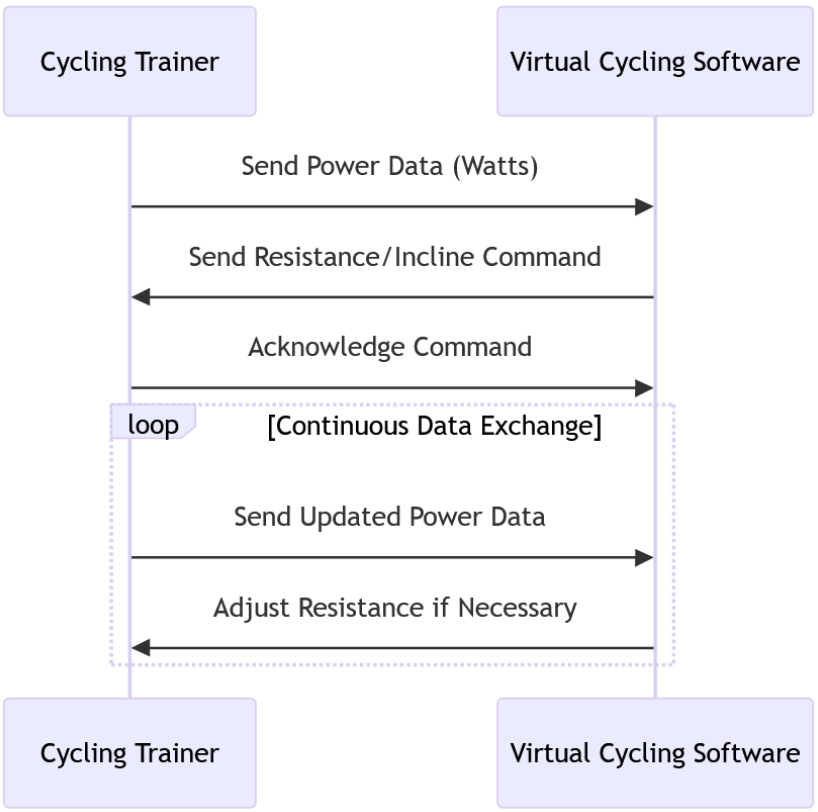
Bluetooth Low Energy (BLE) is the primary protocol used by modern cycling trainers to communicate with virtual riding software. BLE operates on the 2.4 GHz frequency band and is optimized for low power consumption while maintaining reliable data transmission.

BLE follows a client-server model:

- **Cycling Trainer (Peripheral/Server)**: Continuously broadcasts data (e.g., power output, speed, cadence) and listens for control commands.
- **Virtual Cycling Software (Central/Client)**: Connects to the trainer, reads sensor data, and sends commands (e.g., resistance adjustments).

2.2 Data Exchange Mechanism

BLE devices communicate using **GATT (Generic Attribute Profile)**[6], which defines how data is structured and accessed. The trainer provides **services**, which contain **characteristics** that store and transmit cycling metrics.



normal Bluetooth communication

Some key BLE services and characteristics in a cycling trainer:

- **Cycling Power Service (0x1818)**[7]
  - Power Measurement Characteristic (0x2A63) → Transmits real-time power data (watts).
- **Fitness Machine Service (0x1826)**[8]
  - Indoor Bike Data Characteristic (0x2AD2) → Reports speed, cadence, and distance.
  - Fitness Machine Control Point (0x2AD9) → Receives control commands like resistance or incline changes.
- **Device Information Service (0x180A)**[9] → Provides manufacturer details and firmware version.

2.3 Security Features in BLE Communication

BLE implements some security mechanisms[10], but many trainers rely on weak or optional protection:

1. **Pairing & Bonding**: Establishes a secure connection between the trainer and the software, but often without strong authentication.
2. **Encryption**: BLE supports AES-CCM encryption, but most trainers transmit unencrypted data.
3. **Authentication**: BLE 4.0/4.1 lacks robust authentication, and many devices don't implement BLE 4.2+ security features.

2.4 Common Vulnerabilities in BLE Communication for Cycling Trainers

Despite BLE offering security features like encryption and authentication, many cycling trainers and virtual cycling applications implement these protections **inconsistently or inadequately**, making them susceptible to **man-in-the-middle (MitM) attacks**. Below are some of the most common vulnerabilities:

a) Lack of Strong Authentication

- Many trainers **automatically accept connections** from any BLE central device without requiring a PIN or user approval.
- If pairing is required, it is often based on **Just Works** mode (no authentication), making it easy for an attacker to intercept and impersonate the trainer.
- BLE Secure Connections (introduced in BLE 4.2) offers stronger authentication, but many cycling trainers still use BLE 4.0/4.1.

b) Absence of Data Encryption

- BLE devices can encrypt data during transmission, but many cycling trainers do not enable encryption by default.
- Even if encryption is used, some devices allow **downgrade attacks**, where an attacker forces a connection to use an older, weaker encryption method.
- Some implementations rely on **public key exchange during connection setup**, which can be intercepted and replaced by a MitM attacker. This allows the attacker to **inject their own public key**, decrypt encrypted communication, and manipulate data before re-encrypting it for the intended recipient.

c) Static and Predictable Device Addresses

- Many BLE devices use **public** Bluetooth addresses (static MAC addresses), making it easy for an attacker to identify and target a specific trainer.
- Even when random addresses are used, some trainers **do not refresh them periodically**, allowing prolonged tracking and interception.

d) Unprotected GATT Characteristics

- Some trainers allow unrestricted access to their **power measurement characteristics (0x2A63)** and **control characteristics (0x2AD9)**.
- If read/write permissions are too lax, an attacker can:
    - **Read power values** without permission (data leakage).
    - **Modify resistance settings** without authentication (interference with training).

e) Vulnerability to Passive Sniffing

- BLE devices broadcast advertising packets containing key device information.
- Since these packets are often **unencrypted**, an attacker can easily identify:
    - The trainer's **Bluetooth address**
    - The **GATT services and characteristics** available
    - The **device name and manufacturer**, which can help in targeting specific brands.

f) Susceptibility to Replay Attacks

- Some BLE trainers **do not use unique session keys** for communication.
- An attacker can **record valid power data or resistance commands** and replay them later to manipulate the session.

g) Weak Mitigation Against Signal Jamming or Spoofing

- BLE operates on the **2.4 GHz band**, which is shared with Wi-Fi and other devices.
- Attackers can disrupt BLE communication using **signal jamming** or inject fake **advertising packets** to impersonate a real trainer.

2.5 Implications for a MitM Attack

These vulnerabilities make it possible for an attacker to:

- **Intercept and modify power values** (e.g., making it seem like a rider is producing more/less power than they actually are).
- **Send fake control commands** (e.g., setting resistance to unrealistic levels).
- **Impersonate a trainer** to trick the software into connecting to an attacker-controlled device.

3. Man-in-the-Middle (MitM) Attack on Trainer-Software Communication

3.1 Understanding a MitM Attack in BLE

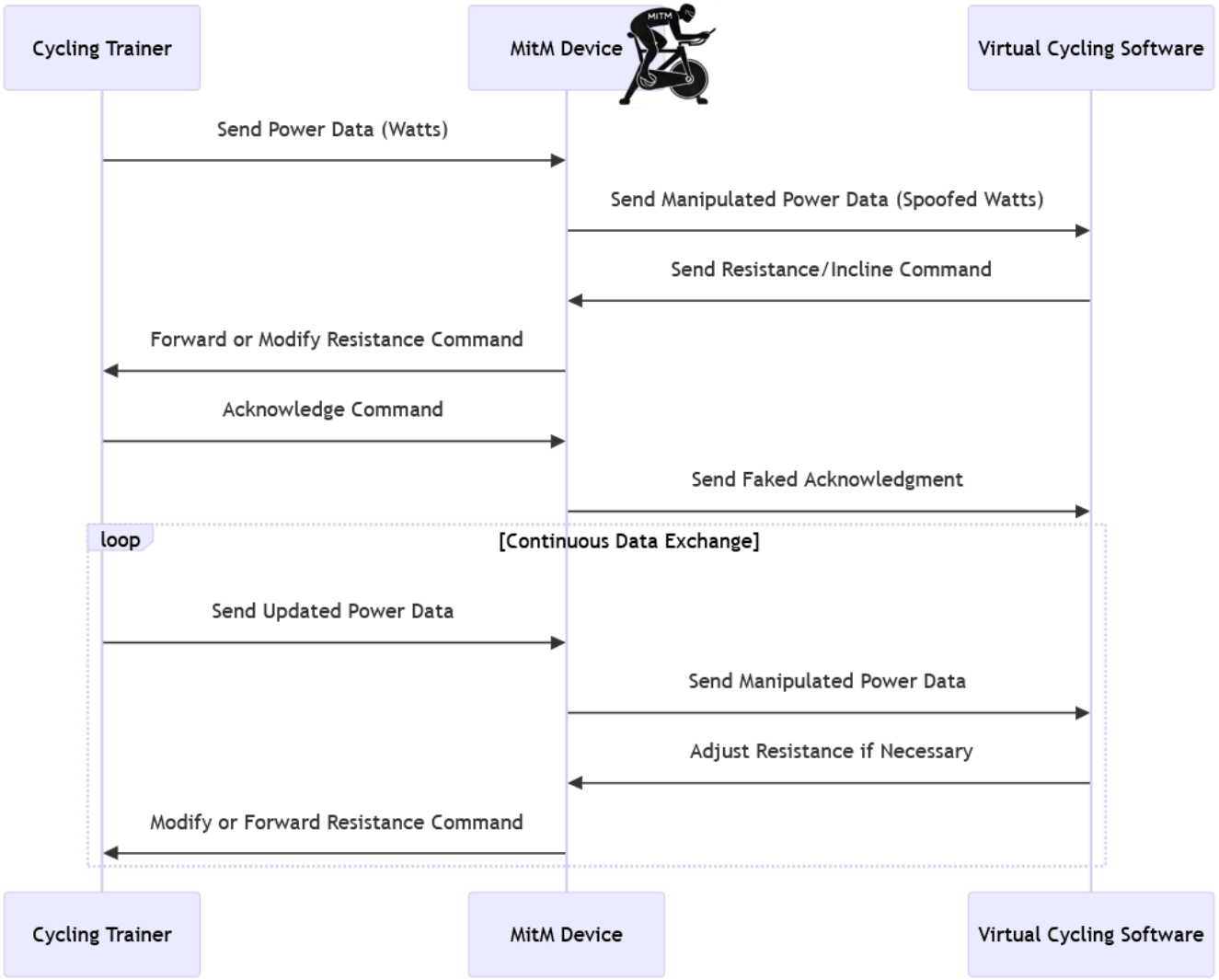
A **man-in-the-middle (MitM) attack** occurs when an attacker intercepts and modifies communication between two devices—in this case, the cycling trainer and the virtual cycling software. Instead of directly communicating, both devices unknowingly exchange data through the attacker's device, which can manipulate messages in real time.

3.2 MitM Attack Steps in a Cycling Trainer Scenario

1. **Scanning and Identifying the Target Trainer**
    - The attacker scans for nearby BLE devices and identifies the cycling trainer based on its **advertising packets** (device name, services, and characteristics).
2. **Intercepting the Connection**
    - The attacker **spoofs the trainer's identity**, tricking the software into connecting to the attacker's device instead of the real trainer.
    - Simultaneously, the attacker **connects to the real trainer**, acting as a relay between the two.
3. **Modifying Data in Transit**
    - The attacker can now intercept and alter:
        - **Power values sent from the trainer to the software** (e.g., increasing power output artificially).
        - **Control commands sent from the software to the trainer** (e.g., modifying resistance settings).
4. **Forwarding the Modified Data**
    - The manipulated data is forwarded to its intended recipient, making it appear as if the values are legitimate.

3.3 Why MitM Attacks Work in This Context

- **Weak authentication mechanisms**: Many trainers accept connections without verifying the legitimacy of the software.
- **Lack of encryption**: If the trainer sends unencrypted power data or accepts unprotected control commands, an attacker can easily modify them.
- **Predictable communication patterns**: Power data is transmitted at regular intervals, making it easy for an attacker to modify without disrupting normal operation.



Bluetooth communication through Man-in-the-Middle device

4. Spoofing Power Values (Trainer → Software)

4.1 How Power Data Is Structured and Transmitted

- Power values from the trainer are sent via the **Cycling Power Service (0x1818)** using the **Power Measurement Characteristic (0x2A63)**.
- This characteristic is **not always encrypted**, making it possible to intercept and modify its contents.
- The data format typically includes:
    - Instantaneous Power (in watts)
    - Pedal Balance (optional)
    - Accumulated Torque (optional)

4.2 How Power Spoofing Works

1. **Interception**: The attacker intercepts power data from the trainer before it reaches the cycling software.
2. **Modification**: The attacker alters the power value to:
    - **Increase power** (e.g., faking higher wattage for virtual racing advantages).
    - **Decrease power** (e.g., making a competitor's trainer appear weaker in a race scenario).
3. **Forwarding**: The manipulated data is sent to the cycling software, which accepts it as valid.

4.3 Potential Impacts

- **Unfair Competitive Advantage**: A rider could appear to generate much higher power than they actually do, gaining an unfair advantage in virtual online races.
- **False Training Feedback**: If power readings are artificially inflated, training sessions may become ineffective.
- **Unintended Performance Adjustments**: Some smart trainers automatically adjust resistance based on power output. Fake data could disrupt this feedback loop.

5. Spoofing Control Values (Software → Trainer)

5.1 How Control Commands Are Structured and Transmitted

- Virtual cycling software sends control commands using the **Fitness Machine Service (0x1826)**.
- The **Fitness Machine Control Point (0x2AD9)** characteristic is responsible for adjusting resistance, incline, and other parameters.
- The command structure typically includes:
    - Command Type (e.g., Set Incline, Set Resistance, Set Target Power)
    - Parameter Value (e.g., incline percentage, target power in watts)

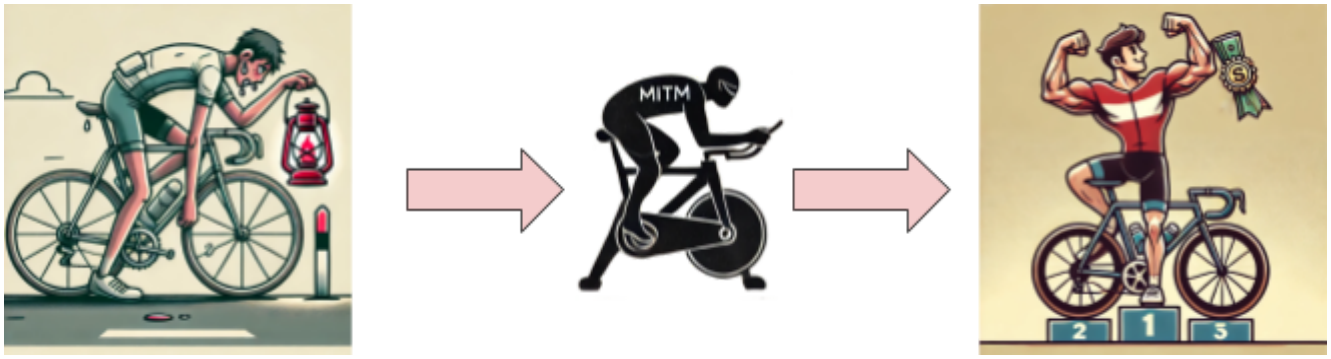
5.2 How Control Spoofing Works

1. **Interception**: The attacker captures control commands from the cycling software before they reach the trainer.
2. **Modification**: The attacker alters key parameters, such as:
    - **Increasing incline** (forcing an extreme uphill simulation).
    - **Reducing resistance** (making pedaling unnaturally easy).
    - **Changing rider weight** (affecting virtual performance calculations).
3. **Forwarding**: The modified command is sent to the trainer, which applies the manipulated settings.

5.3 Potential Impacts

- **Disrupting Workouts**: Resistance changes can make structured training ineffective or overly difficult.
- **Causing Physical Strain**: Unexpected resistance spikes could lead to muscle strain or loss of control.
- **Creating an Unfair Advantage**: Lowering weight or resistance can manipulate in-game physics for competitive gain.

6. Practical Methods for Executing a MitM Attack



6.1 Using the QZ App for Data Manipulation

The **QZ app (qdomyos-zwift)**[11] is a widely used third-party application designed to **enhance compatibility** between various fitness devices and virtual cycling platforms. It provides a valuable solution for users with non-standard or older trainers that lack direct integration with apps like Zwift, TrainerRoad, and Rouvy. By acting as a **bridge**, QZ allows users to customize their training experience, fine-tune power readings, and optimize resistance settings for a smoother ride.

Benefits of QZ App:

- **Expands Compatibility:** Enables users to connect non-supported trainers, making virtual cycling accessible to a wider audience.
- **Customization Options:** Allows users to fine-tune power and resistance settings to match real-world conditions.
- **Bridging Between Protocols:** Converts and relays data between Bluetooth devices, improving connectivity.
- **Easy to operate** without special technical knowledge or understanding

How QZ Could Be Exploited for Spoofing:

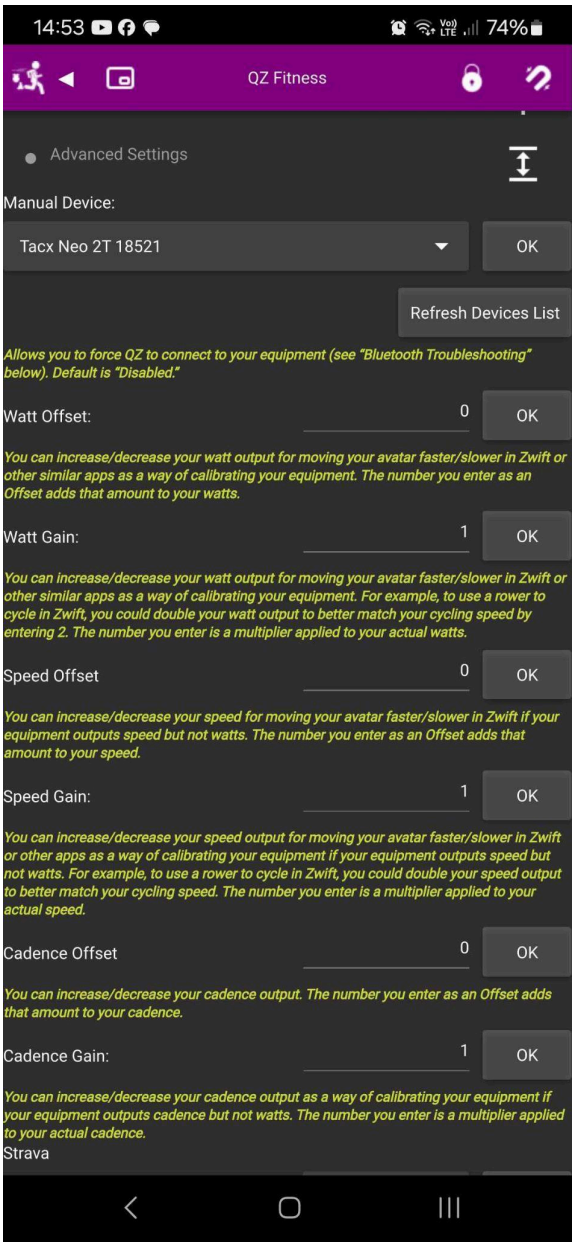
While QZ is designed for legitimate use, its flexibility also introduces potential for **misuse**:

- **Acts as an intermediary** between the trainer and software, similar to a MitM setup.
- **Allows modification of performance metrics** before forwarding data to the virtual cycling platform.
- **Can dynamically alter** power values, resistance settings, or rider weight, affecting in-game performance.

Example of Potential Manipulation:

1. QZ connects to the trainer and receives **real** power and resistance data.
2. The app **modifies** these values (e.g., doubling power output) based on user settings.
3. The manipulated data is then sent to the virtual cycling platform, appearing legitimate.

While QZ is primarily intended for **enhancing accessibility and user control**, its ability to modify real-time data makes it a tool that, if misused, could enable unfair advantages in competitive environments.



QZ Fitness App at Android

6.2 Using an ESP32 for BLE Interception and Spoofing

The **ESP32**[12] is a widely used microcontroller with built-in BLE support, making it a powerful tool for intercepting and modifying Bluetooth traffic.

Why ESP32?

- **Low-cost and accessible**: Available for under $10.
- **Dual BLE roles**: Can act as both a BLE central (connecting to the trainer) and a BLE peripheral (spoofing the trainer to the software).
- **Programmability**: Supports **Arduino, ESP-IDF, and Micropython**, allowing customization of BLE behavior.

How an ESP32 Can Be Used for a MitM Attack:

1. **Scan for nearby BLE devices** to identify the cycling trainer.
2. **Mimic the trainer's Bluetooth identity** (spoof MAC address, services, and characteristics).
3. **Connect to the real trainer** and relay data between the trainer and the software.
4. **Modify power or control values in real-time** before forwarding them.

There are open-source ESP32-based BLE projects (e.g., **ESP32 BLE Keyboard**, **BLE GATT Server**) that can be adapted for this purpose.

6.3 Using a PC for Bluetooth MitM Attacks

A **PC with a Bluetooth adapter** offers a more flexible and powerful environment for conducting MitM attacks compared to ESP32 or mobile apps.

Required Hardware & Software

- **Bluetooth 4.0+ USB Adapter** (with HCI support for raw packet capture)
- **Linux-based OS** (Kali Linux, Ubuntu, or a customized environment)
- **Software Tools:**
    - **Wireshark**[13] (for Bluetooth packet sniffing)
    - **gatttool** (for BLE communication)
    - **Bettercap**[14] (for BLE MitM and spoofing)
    - **btproxy**[15] (for Bluetooth packet relay and modification)
    - **BtleJuice**[16] (for advanced MitM attacks on BLE devices)

BtleJuice is a powerful framework specifically designed for **BLE MitM attacks**. It enables:

- **Live interception and modification** of BLE packets.
- **Relaying and spoofing** of GATT characteristics in real-time.
- **Easy setup of an intermediate proxy**, allowing full control over BLE communication.

How a PC-Based MitM Attack Works with BtleJuice

1. **Sniffing Bluetooth Communication**
    - The attacker places a Bluetooth adapter in **monitor mode** to capture BLE traffic.
    - **Wireshark** can be used to inspect raw BLE packets.
2. **Spoofing the Cycling Trainer**
    - The PC **clones the trainer's MAC address and GATT services**.
    - Using **BtleJuice**, the attacker **relays and manipulates BLE characteristics** before forwarding them to the software.
3. **Intercepting and Modifying Data**
    - The attacker can alter power values, resistance settings, or other control parameters in real-time.
    - **BtleJuice** allows modifying BLE requests and responses dynamically.
4. **Relaying Manipulated Data to the Cycling Software**
    - The PC **forwards fake power or resistance data** to the software.
    - The software **accepts the spoofed values as if they were legitimate**.

Advantages of Using a PC for MitM Attacks

- **Greater processing power** – Can handle real-time modifications with precision.
- **Advanced debugging tools** – Easier to analyze and manipulate BLE packets.
- **BtleJuice's modular approach** – Allows injecting, modifying, and filtering BLE data with ease.

7. Real-World Impact of MitM Attacks on Training & Competition

While MitM attacks are often associated with **virtual cycling platforms**, their impact extends beyond esports. Many athletes and coaches rely on **smart trainers paired with cycling computers (e.g., Garmin Edge**[17]**), training apps (e.g., Wahoo SYSTM**[18]**, TrainerRoad**[19]**), and structured workout platforms**. A successful MitM attack could manipulate critical training data in **any scenario where a trainer communicates via Bluetooth Low Energy (BLE).**

7.1 Impact on Professional and Competitive Events

MitM attacks could disrupt **both virtual and real-world racing events**. Professional cyclists often use **smart trainers for warm-ups** before time trials or key races, such as the **Tour de France** or **UCI-sanctioned events**. A targeted attack could:

- **Distort power data**, causing a rider to miscalculate their pacing or effort.
- **Modify resistance settings**, leading to ineffective warm-ups or unnecessary fatigue.
- **Cause delayed or incorrect feedback**, reducing confidence in power readings during crucial moments.

7.2 Training and Performance Analysis Risks

Many athletes use **Garmin, Wahoo, or other cycling computers** to track **power output, cadence, and heart rate** during structured workouts. A MitM attack in this setting could:

- **Manipulate power readings**, making an athlete believe they are training harder or easier than they actually are.
- **Falsify cadence or heart rate data**, disrupting structured interval training.
- **Interfere with real-world ride tracking**, affecting post-ride analysis and long-term performance assessments.

7.3 Coaching and Training App Vulnerabilities

Coaches and training platforms like **TrainingPeaks, TrainerRoad, and Wahoo SYSTM** rely on accurate power data to **set training zones, prescribe workouts, and analyze progress**. If power data is spoofed:

- **Training plans could be compromised**, leading to incorrect load progression.
- **Athletes might train at the wrong intensity**, resulting in undertraining or overtraining.
- **Performance benchmarking becomes unreliable**, affecting goal setting and race preparation.

7.4 Potential Safety Risks

Beyond training effectiveness, **incorrect trainer resistance adjustments** due to a MitM attack could pose **physical risks**:

- **Unexpected resistance increases** could strain muscles or lead to knee injuries.
- **Resistance drops mid-effort** might cause loss of control, especially during **sprint intervals or out-of-saddle efforts**.
- **Overexertion risks**—if power data is manipulated to underreport effort, an athlete may push harder than intended, risking **heat exhaustion or cardiovascular stress**.

7.5 Conclusion

MitM attacks on BLE-connected trainers pose risks **beyond virtual cycling**. Whether used for **race preparation, structured training, or performance analysis**, **athletes, coaches, and event organizers depend on accurate data**. Ensuring secure BLE communication is crucial for **maintaining training effectiveness, competitive fairness, and rider safety**.

8. Mitigation Strategies

To protect against MitM attacks on Bluetooth communication between cycling trainers and virtual cycling software, both **hardware manufacturers** and **software developers** need to implement security measures [10].

8.1 Strengthening Authentication Mechanisms

- **Enforce pairing with passkeys or numeric comparison** instead of the insecure "Just Works" mode.
- **Use device whitelisting** to restrict connections to authorized software only.

8.2 Enabling and Enforcing Asymmetric Encryption with Known Secure Keys

In an ideal setup for preventing **man-in-the-middle (MitM) attacks**, **asymmetric encryption** should be used, where each device (trainer and software) has a **private key** and a **public key**. The public keys can be freely distributed, while the private keys are securely stored. This ensures that the data is **securely encrypted** and can only be decrypted by the intended recipient.

Why Asymmetric Encryption Is Better:

- **Non-repudiation and authenticity**: The use of private keys to sign data ensures that the sender can be verified, and the communication remains tamper-proof.
- **Stronger security**: Unlike symmetric key encryption, asymmetric encryption doesn't require sharing a secret key, which makes it more secure and immune to key interception attacks.
- **Encrypted data protection**: Even if an attacker intercepts the transmission, they cannot decrypt the data without access to the private key.

How to Implement Asymmetric Encryption:

- **Public key distribution**: Each trainer hardware and virtual cycling platform software has a **pre-distributed, securely stored public key**. **There should be no public key exchange during connection** to prevent MitM key injection attacks. The **private keys remain hidden and never transmitted**.
- **Use secure storage (e.g., hardware security modules or secure enclaves)** to protect private keys.
- **Data encryption**: Data sent from the trainer to the software (and vice versa) is encrypted with the recipient's public key. Only the recipient can decrypt it using their private key.

- **Digital signatures**: Data can be signed with the sender's private key, and the recipient can verify the signature using the public key, ensuring authenticity.

For example, the trainer could encrypt the power data with the **software's public key**, and the software would decrypt it using its private key. Similarly, the software could encrypt control commands with the **trainer's public key**.

8.3 Regulatory Compliance: The EU Cyber Resilience Act (CRA)

The **Cyber Resilience Act (CRA)**[20] enforces mandatory cybersecurity requirements for connected devices, including cycling trainers and virtual cycling platforms. Under the CRA, manufacturers must:

- Implement **stronger encryption and authentication mechanisms** to prevent attacks like MitM.
- Ensure **lifecycle security**, including vulnerability patching and secure software updates.
- Prove compliance to maintain market access in the **EU**, ensuring that future BLE-based trainers meet cybersecurity standards.

While the CRA enforces encryption and security best practices, **it does not fully eliminate MitM risks**. **If a device exchanges public keys during the BLE handshake, an attacker can intercept and replace them**. Therefore, manufacturers must go beyond CRA requirements by ensuring **pre-distributed public keys and secure key storage mechanisms**.

8.4 Implementing Signed Data and Timestamps

- **Digitally sign power values and control commands** so any tampering is detectable.
- **Include timestamps in BLE messages** to prevent replay attacks.

8.5 Randomizing and Rotating Bluetooth Addresses

- **Use private resolvable addresses (RPA)** that change periodically to make device tracking harder.

8.6 Restricting Access to GATT Characteristics

- **Set proper read/write permissions** on power and control characteristics.
- **Require authentication for modifying trainer settings** to prevent unauthorized control.

8.7 Detecting and Preventing Anomalous Data

- **Software should flag sudden, unrealistic power changes** that could indicate spoofing.
- **Use server-side validation in virtual cycling platforms** to verify expected power and resistance trends.

9. Conclusion

This paper has demonstrated the potential for **man-in-the-middle (MitM) attacks** on Bluetooth communication between cycling trainers and virtual cycling software, highlighting the risks of **data manipulation**, particularly for **power values and control commands**. As shown, vulnerabilities in BLE communication protocols—such as **weak authentication, lack of encryption, and unprotected GATT characteristics**—expose users to the risk of **malicious interference**. These weaknesses not only **threaten the integrity of competitive events**, enabling unfair advantages, but also **raise concerns about user trust** in virtual cycling platforms.

The **Cyber Resilience Act (CRA)** is expected to **push manufacturers toward stronger BLE security**, ensuring that cycling trainers and virtual platforms adopt **robust encryption, authentication, and update mechanisms**. While regulatory pressure will likely **accelerate the adoption of more secure BLE communication protocols**, it does **not fully eliminate** MitM risks. Specifically, **vulnerabilities like public key exchange during handshake** remain a concern, highlighting the need for additional security measures **beyond CRA compliance**.

Implementing **stronger security measures**, such as **asymmetric encryption, enhanced authentication, and periodic address randomization**, can **significantly reduce** the likelihood of a successful MitM attack. However, the challenge lies in **balancing security with accessibility** - not all users or devices may support advanced security features due to **hardware limitations**. Therefore, a **nuanced approach** is required.

One potential solution is the **implementation of configurable security settings**, allowing virtual cycling platforms to **strengthen security for high-impact events** (e.g., races with monetary prizes) or **adjust security measures based on the severity of potential risks**. This targeted approach ensures that **participants with capable hardware are protected**, while still allowing the **broader community to engage** without imposing excessively high security requirements.

Further research should explore the **feasibility of adaptive security mechanisms** and the development of **lightweight cryptographic solutions** suitable for resource-constrained devices. Ultimately, securing BLE communication in virtual cycling is **not just about fair competition** - it is **critical for user trust, long-term platform viability, and the continued growth of the sport**. Manufacturers and software developers must **proactively improve security beyond regulatory mandates**, ensuring a **secure and fair environment for all users**.

10. References

[1] Bluetooth SIG, “Bluetooth Low Energy (BLE) Overview,” [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/. [Accessed: Feb. 16, 2025].

[2] Zwift, “Zwift: The at-home training game connecting cyclists around the world,” [Online]. Available: https://www.zwift.com/. [Accessed: Feb. 16, 2025].

[3] MyWhoosh, “MyWhoosh: Your Ultimate Indoor Cycling App,” [Online]. Available: https://www.mywhoosh.com/. [Accessed: Feb. 16, 2025].

[4] TrainingPeaks Virtual, “TrainingPeaks Virtual (formerly IndieVelo),” [Online]. Available: https://www.trainingpeaks.com/virtual/. [Accessed: Feb. 16, 2025].

[5] Rouvy, “Rouvy: Indoor Cycling Reality,” [Online]. Available: https://www.rouvy.com/. [Accessed: Feb. 16, 2025].

[6] Bluetooth SIG, “GATT Specification Supplement” [Online]. Available: https://www.bluetooth.com/specifications/gss/. [Accessed: Feb. 16, 2025].

[7] Bluetooth SIG, “Cycling Power Service” 2025. [Online]. Available: https://www.bluetooth.com/specifications/specs/cycling-power-service/. [Accessed: Feb. 16, 2025].

[8] Bluetooth SIG, “Fitness Machine Service” 2025. [Online]. Available: https://www.bluetooth.com/specifications/specs/fitness-machine-service-1-0-1/. [Accessed: Feb. 16, 2025].

[9] Bluetooth SIG, “Device Information Service” 2025. [Online]. Available: https://www.bluetooth.com/specifications/specs/device-information-service/. [Accessed: Feb. 16, 2025].

[10] Bluetooth SIG, “Bluetooth Security & Privacy,” 2025. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/key-attributes/bluetooth-security/ . [Accessed: Feb. 16, 2025].

[11] QZ App, “qdomyos-zwift: QZ App Documentation,” GitHub, [Online]. Available: https://github.com/cagnulein/qdomyos-zwift & https://www.qzfitness.com. [Accessed: Feb. 16, 2025].

[12] Espressif Systems, “ESP32 Technical Documentation,” [Online]. Available: https://www.espressif.com/en/products/socs/esp32/. [Accessed: Feb. 16, 2025].

[13] Wireshark, “Wireshark: The World's Premier Network Protocol Analyzer,” [Online]. Available: https://www.wireshark.org/. [Accessed: Feb. 16, 2025].

[14] Bettercap, “Bettercap: The Swiss Army Knife for Hacking,” [Online]. Available: https://www.bettercap.org/. [Accessed: Feb. 16, 2025].

[15] Btproxy, “Btproxy: Man in the Middle analysis tool for Bluetooth.” GitHub, [Online]. Available: https://github.com/conorpp/btproxy. [Accessed: Feb. 16, 2025].

[16] BtleJuice, “BtleJuice: Bluetooth Smart Man-in-the-Middle Framework,” GitHub, [Online]. Available: https://github.com/DigitalSecurity/btlejuice. [Accessed: Feb. 16, 2025].

[17] Garmin, “Garmin Cycling Devices & Accessories,” [Online]. Available: https://www.garmin.com/. [Accessed: Feb. 16, 2025].

[18] Wahoo, “Wahoo SYSTM: Personalized Training for Cyclists,” [Online]. Available: https://www.wahoofitness.com/systm/. [Accessed: Feb. 16, 2025].

[19] TrainerRoad, “TrainerRoad: Cycling’s Most Effective Training System,” [Online]. Available: https://www.trainerroad.com/. [Accessed: Feb. 16, 2025].

[20] European Commission, “The Cyber Resilience Act (CRA)” [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act & https://www.european-cyber-resilience-act.com. [Accessed: Feb. 16, 2025].