

Hardware Experiment Operation Manual

IoT 重放攻击防御实验：实机操作手册 (Definitive Operation Manual)

警告：本手册涉及无线电设备操作。在日本境内实验请务必优先使用 **SMA 同轴电缆 + 衰减器** 直连方式 (Conducted Test)，或在电波暗室进行，以严格遵守电波法。本文档是本项目的**唯一官方操作指南**。请严格按照以下步骤，从零开始搭建 Hardware-in-the-Loop (HIL) 实验环境。

阶段一：硬件与环境准备

1.1 硬件清单 (Checklist)

请逐一核对以下设备：

- ☐ **PC:** 1 台 (macOS 或 Ubuntu Linux, Windows WSL2 不推荐因 USB 透传问题)
- ☐ **SDR 设备:** 3 台 HackRF One (分别扮演 Alice, Bob, Mallory)
- ☐ **Micro-USB 数据线:** 3 根 (必须是支持数据传输的高质量线材)
- ☐ **SMA 同轴电缆:** 3 根 (用于设备间射频信号直连, **强力推荐**以获得稳定结果)
- ☐ **SMA 衰减器 (Attenuator):** 至少 1 个, 建议 20dB 或 30dB。
 - 重要提示: 如果使用电缆直连, **必须在发射端 (Alice/Mallory) 的 ANT 接口串联衰减器**, 否则高功率信号会**烧毁**接收端 (Bob) 的芯片!

1.2 物理连接步骤 (Safety First)

绝对红线：HackRF 的 RX 端口最大输入功率为 **-5 dBm**。过强的信号会**直接烧毁**接收芯片！

1. **衰减器安装：**在 Alice 和 Mallory 的 ANT (TX) 端口上, **必须先拧上 30dB 以上** 的衰减器 (如果直连, 建议 40-60dB)。
2. **线缆连接：**将 SMA 同轴电缆连接在衰减器和 Bob 的 ANT (RX) 端口之间。

3. **USB 连接**: HackRF 连电脑。注意尽量别插在同一个 USB Hub 上, 防止带宽拥堵。

1.3 软件环境安装

打开终端 (Terminal), 逐行执行以下命令:

```
# 1.
cd ~/Desktop/ /Replay

# 2.
python3 -m venv .venv
source .venv/bin/activate

# 3.
pip install pyzmq matplotlib numpy

# 4.  GNU Radio  HackRF  (macOS)
#  soapyhackrf  GRC  HackRF
brew install gnuradio soapyhackrf

# 5.
hackrf_info
```

1.4 设备序列号登记表

角色	序列号 (Serial)	用途
Alice	-----	发送端
Bob	-----	接收端
Mallory	-----	攻击端

2. 核心概念与协议约定 (必须读!)

在开始连线之前, 有两个“导致实验失败”的坑必须避开:

2.1 空口“标签”不传输

GNU Radio 内部的 Tag (如 packet_len) 不会随无线电波飞到接收端。
* **错误做法**: TX端打个 Tag, 指望 RX 端自动知道包长度。 * **正确做法**: 必须使用 **物理层帧格式 (PHY Frame)**。 * 本手册推荐 **固定包长 (Fixed Length)** 策略: 简单粗暴但有效。 * Alice 每次发 100 字节, Bob 每次解 100 字节。

2.2 ZMQ 消息格式 (PMT)

GNU Radio 的 ZMQ Message 块使用的是 PMT (Polymorphic Type) 序列化格式。* Python 脚本必须发送/接收 **Serialized PMT String**。* 我们在 `hardware_experiment.py` 中已经封装好了这层转换，但请确保你不要随意改动 ZMQ 相关的底层代码。

阶段三：搭建 GNURadio 流图 (GRC)

任务 A: 发送端 `alice_tx.grc` (Fixed Length)

1. ZMQ PULL Message Source: 地址 `tcp://127.0.0.1:5555`。
2. PDU to Tagged Stream: Tag `packet_len`。
3. Repack Bits: 8 到 1 (把字节流变成比特流)。
4. GFS Mod: 调制。
5. HackRF Sink:
 - RF/IF/BB Gain: 全部设为 0 (直连测试起步越低越好)。
 - Sample Rate: 2M。
 - Device Args: `hackrf=ALICE_SERIAL`。

任务 B: 接收端 `bob_rx.grc`

1. HackRF Source: Device Args `hackrf=BOB_SERIAL`。
2. Low Pass Filter: 降噪。
3. GFSK Demod: 解调为字节流。
4. Correlate Access Code – Tag Stream:
 - Access Code: (默认32位)。
 - Threshold: 0 或 1。
 - Tag Name: `packet_len`。
5. Repack Bits: 1 到 8。
6. Tagged Stream to PDU:
 - Tag: `packet_len`。
 - PDU Length: 手动填 100 (或者你的固定包长)。这里不能依赖 Tag 的值，必须硬编码长度！
7. ZMQ PUSH Message Sink: `tcp://127.0.0.1:5556`。

任务 C: 攻击者 (Mallory) – 分两步走

由于 HackRF 是 半双工 (Half-Duplex)，必须拆分成两个独立文件：

1. `mallory_rx.grc` (监听用):
 - 配置同 Bob，ZMQ 输出到 5557。
 - Device Args: `hackrf=MALLORY_SERIAL`。
2. `mallory_tx.grc` (重放用):

- 配置同 Alice, ZMQ 输入从 5558。
- Device Args: hackrf=MALLORY_SERIAL。

操作技巧: 在 GRC 界面里, **不要同时点击两个流图的运行**。* 想听的时候: 运行 rx, 停止 tx。* 想发的时候: 停止 rx, 运行 tx。* 或者使用脚本 `attacker_relay.py` 的 `--strategy post` 模式, 它会配合你的手动切换。

阶段四: 自动化实验执行

终端 1: Bob

```
python scripts/gnuradio_adapter.py rx --port 5556 --sim-mode window
```

终端 2: Alice

```
python scripts/hardware_experiment.py --runs 50 --num-legit 20 --p-loss 0.3 --mode window
...(后续步骤同上)... python scripts/gnuradio_adapter.py rx -port 5556 -
sim-mode window
```

```
* *:      "Connecting..."
```

```
### [Terminal 2]   GRC
```

```
GNURadio
```

```
1.      `bob_rx.grc` ->
```

```
2.      `alice_tx.grc` ->
```

```
3.      `mallory.grc` ->
```

```
* *:      HackRF   TX/RX      /
```

```
### [Terminal 3]   Python
```

```
Alice
```

```
```bash
```

```
cd ~/Desktop/ /Replay
```

```
source .venv/bin/activate
```

```
50 20 30%
```

```
python scripts/hardware_experiment.py --runs 50 --num-legit 20 --p-loss 0.3 --mode window
```

```
状态: 你会看到数据开始滚动: [Run 1 Msg 1] Sent=1 | Recv=1 ->
```

```
ACCEPTED [Run 1 Msg 2] Sent=2 | Recv=2 -> ACCEPTED
```

### [Terminal 4] 启动 Mallory 攻击脚本 (可选)

如果想测试三方攻击:

```
cd ~/Desktop/ /Replay
```

```
source .venv/bin/activate
```

```

python scripts/attacker_relay.py --strategy random_delay
```

状态: 你会在 Terminal 1 (Bob) 中看到红色的警告: REJECTED (Replay Detected)。

---

## 阶段四: 数据收集与分析

实验完成后, Terminal 3 会输出最终报告:

```
=== HARDWARE RESULTS ===
Legit: 980/1000 Accepted (98.0%)
Attack: 0/100 Successful (0.0%)
Real Timeouts: 15
Artificial Drops: 300
```

**你的任务:** 1. **截图:** 将这个终端的一页绿字/红字截图, 作为“实机实验运行证明”。  
2. **填表:** 将 Legit Acceptance Rate 和 Attack Success Rate 填入论文的实验结果表格。  
3. **分析:** \* 如果 Real Timeouts 很高 (>5%), 说明物理连接有问题 (线没拧紧) 或干扰太大。  
\* 如果 Attack 成功率 > 0%, 说明你的防御代码 (Window/Counter) 有 bug, 或者 Mallory 碰巧猜对了 (极低概率)。

---

## Q&A 常见故障急救

- **Q: 报错 Address already in use?**
  - A: 上次程序没关干净。关掉所有终端, 甚至重启电脑。
- **Q: HackRF 灯不亮?**
  - A: 也就是 GRC 没点运行, 或者 USB 接触不良。
- **Q: Bob 收不到任何数据?**
  - A: 检查 GRC 里的序列号是不是填反了 (Alice 用了 Bob 的号)。检查频率是否一致 (都是 2.45G)。
- **Q: GRC 报错 xterm executable not found?**
  - A: 在 GRC 的 Options 块里, 把 Generate Options 改为 No GUI。我们在终端看结果, 不需要图形波形。