Laboratory for wireless remote control

This version has been prepared for
either offline use under COVID-19 restrictions
or live use in a laboratory.

This laboratory will teach you how to receive and decode radio signals, akin to those commonly used by IoT devices. Utilising a LimeSDR Mini, you will decode a modulation scheme. You will then form a payload to attack your IoT device, and you will have an opportunity to explore how to combine this with fuzzing to compromise multiple IoT devices.

Submission Instructions

For each numbered section in each task, please write what you did, what you learnt, and aim to answer each sub-question on the template provided. You should aim to write no more than 100 words per numbered section, with a total of around 1000–1200 words. Bullet points or brief sentences are fine; there is no need to write an essay, What matters is the content, and the understanding that you are showing.

Getting Started

During this laboratory, you will utilise two main pieces of software. The first of these, *SDR Console*, is a universal Software Defined Radio client for Windows. Being compatible with most major SDRs and very simple to use, we will utilise this for basic spectrum monitoring and visualisation.

Universal Radio Hacker is a very useful piece of software for analysing digital signals. We will be using this to intercept and replay digital signals to a common "Garage Door Opener"-type IoT device. It imposes a potentially counter-intuitive workflow which may lead to something of a "steep learning curve".

Both of these pieces of software are available on the laboratory machines. You can also install them on your own Windows system.

It is recommended to use the virtual machine image in the lab to get started quickly.

Task 1 - Understanding the target device

You have been issued a standard radio controlled relay module and fob as part of this laboratory. Remove the top cover of the relay board, but **do not attempt to dismantle the fob**. You may unscrew the relay board from the bottom cover if you wish, but reinstall it before continuing.

Using this information:

1. Identify the relay board and describe its components.
2. Without inspecting the keyfob, calculate the operating frequency of the relay board.
3. What is the operating frequency of the keyfob? Is this legal in the United Kingdom?
4. Arrange to supply the appropriate power to the board utilising the bench apparatus. What voltage is used?

5. Select appropriate resistors, and connect four LEDs so that they turn on when the corresponding relay is engaged, Show your circuit.
6. Power the system, and check and record the current consumption. If necessary, follow the appropriate learning sequence to provision the relay board for the keyfob, then check that relay "A" operates with the remote control.

With your relay board now confirmed working, we will use a LimeSDR Mini to monitor the spectrum in which the device operates.

First, you will need to use the provided SMA connector to build a $\lambda/4$ antenna for the frequency you identified above. If you have not yet been risk assessed to use the laboratory soldering irons, please consult the laboratory technicians. With your antenna built, screw it gently into the **RX** port on the LimeSDR. Do not use a spanner! You are now ready to begin using the SDR to investigate your IoT Relay.

Task 2 - Spectrum Monitoring

For this task we will be using SDRConsole, a windows-only utility which facilitates spectrum monitoring using the LimeSDR.

1. Establish the best PC USB port to use with your SDR. Are some ports more or less noisy than others? Why?
2. What is the exact frequency on which the garage door opener transmits?
3. Do other traces appear when you push the button? Which, and why? Take an appropriate "screen capture"
4. Can you learn anything by listening to the audio output from SDR Console? What?

You may be able to see from the waterfall, and hear on the audio output, that this is an on-off keyed signal.

Task 3 - Capture, Analysis and Replay

For this next task, you will need to use Universal Radio Hacker, an open-source tool for analysing digital signals. The x64 Windows installer can be found at
https://github.com/jopohl/urh/releases/download/v2.9.2/Universal.Radio.Hacker-2.9.2-x64.exe.
If SDRConsole is already working with your LimeSDR Mini, you should not need to install or configure any additional software.

Capture the signal

Open Universal Radio Hacker by selecting the icon on the Desktop.

In order to capture your signal, make sure that the **SDRConsole is closed** so that URH can access the radio device, then select *File* → *Record Signal...*.

There are a variety of options on this page; you will want to set the following:

- Device: LimeSDR
- Device Identifier: *Press the refresh button and this should populate.*
- Antenna: Wide (RX_W)
- Frequency: About 500kHz below the frequency you identified above.
- Sample Rate and Bandwidth: By default these will be 2MSps (2MHz Bandwidth) which should be fine.
- Gain: 70 is a good initial value
- DC Correction: On

When happy, press the "Start" button to begin capture. The command trace will show in the bottom left, and you will need to wait approximately 10 seconds for the capture to begin.

When happy that you have captured your signal, press "Stop" to end.

Then, press "Save" to save the capture to disk. Give it a useful name; you might include the button you pressed on the garage door opener.

When happy, close the capture window and your trace will populate in the "Interpretation" tab of Universal Radio Hacker. You can take multiple captures; these will show up underneath each other as part of the interpretation tab.

Replay the signal

Now that you have a signal captured, you can use the LimeSDR to play back an exact replica and see if the device recognises it.

First, move your antenna from the 'RX' port to the 'TX' port.

In the *Interpretation* tab, press the small play button (▶) above the filename of the trace you wish to replay. This will open the transmission window where, again, you will need to set the following:

- Device: LimeSDR
- Device Identifier: *Press the refresh button and this should populate.*
- Channel: TX1
- Antenna: Band 1 (TX_1)
- Frequency: The same frequency that was used for receiving.
- Sample Rate: The same sample rate that was used for receiving.
- Gain: 50 is a good initial choice
- Repeat: Change this from "Infinite" to e.g. 5.

You may need to place the receiver close to the LimeSDR for this to work. You may also need to re-capture the original signal you are replaying, if it is too noisy.

Questions:

1. How effective is this kind of attack?
2. What advantages and disadvantages do replay attacks have compared with fully reconstructing the signal?
3. Would this type of attack work on a wireless car key?

Task 4 - Analysis of the Signal

Traces can be manipulated by using the mouse to select regions of the trace, and unwanted information can be deleted with the delete key. The mouse wheel is used to zoom, and the Y scale can be adjusted at the right hand side of the trace.

Questions:

1. What is the repeating unit of the signal? What is the bit period?
2. What is the duration of a single bit within a message?
3. Can you identify the encoder chip that is used?
4. Does the chip identity help you to modify your answer to 4.1?
5. Can you modify the messages to operate button "B"?
6. Using the chip information, how many different ID codes are supported by this system?

7. How many repeats of the message are required to trigger a relay?
8. To what frequency is the receiver *actually* tuned?
9. If you were not able to capture a signal from a target system, what is the maximum amount of time it would take to fuzz every ID code combination for a single relay channel, and thus open the door without a keypad?