LAB4- Ring the Bell

1. **What is the frequency on which the remote control is sending its messages?**

   434 Mhz +- 75 Khz [peaks seen at 434.128, 434.116, 433.977]
   We sent our replay signal on 434.116 Mhz frequency.

2. What modulation scheme is used to modulate the data?
   Amplitude shift keying

3. How many symbols did you count? What is the symbol period? (1 point)

4. What are the bits you have recovered from the signal? (1 point)
   0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0

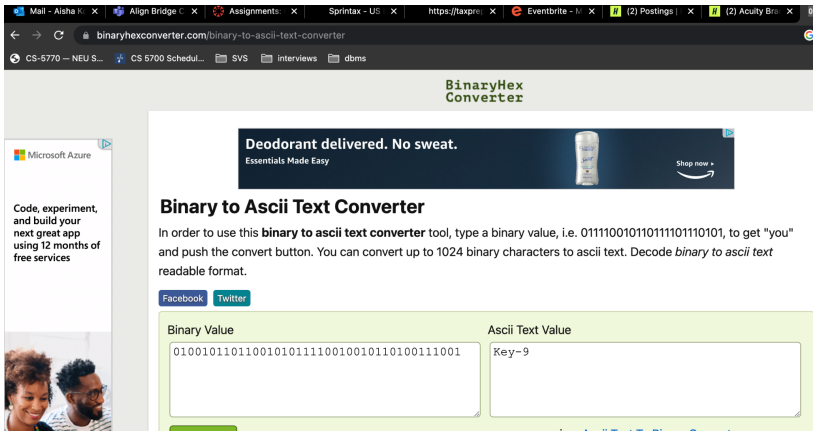
5. **Discuss at least two countermeasures along with potential limitations that can prevent an attacker from performing this replay attack? (1 point)**

   A. **Using session IDs**- session IDs are unique random numbers that are used for communication between a transmitter and receiver. They are different for each session in the communication system. These session IDs can't be predicted, thereby it's not possible for the attacker to generate session keys the same as the transmitter for the replayed signal. Therefore receiver can invalidate the replayed signal from an adversary.
   Limitation - This increases transmitter and receiver complexity and may not be suitable for use cases where the cost of the product is minimal such as doorbells.

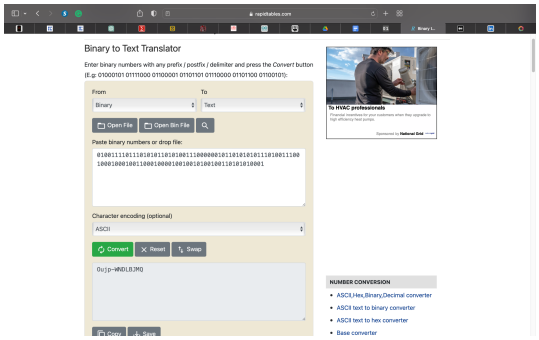
   B. **Using timestamps and TTL (time-to-live)** - Adding timestamps for each packet and determining the appropriate time to live for each packet can help filter out replayed packets. TTL is the maximum amount of time a packet is valid. If a packet is received after the TTL is expired, the receiver can discard the packet.
   Limitation - This is a huge performance burden to add a timestamp to each packet. And there should be synchronization between the transmitter and receiver for this to work properly.

6. The signal transmitted in the lab also carries additional data (key and secret message). These symbols are different from that of the bell. Decode the symbols to extract the message. (1 point)
   Key is : 9
   Message is : Flag-NEUCSADH

   Explanation:

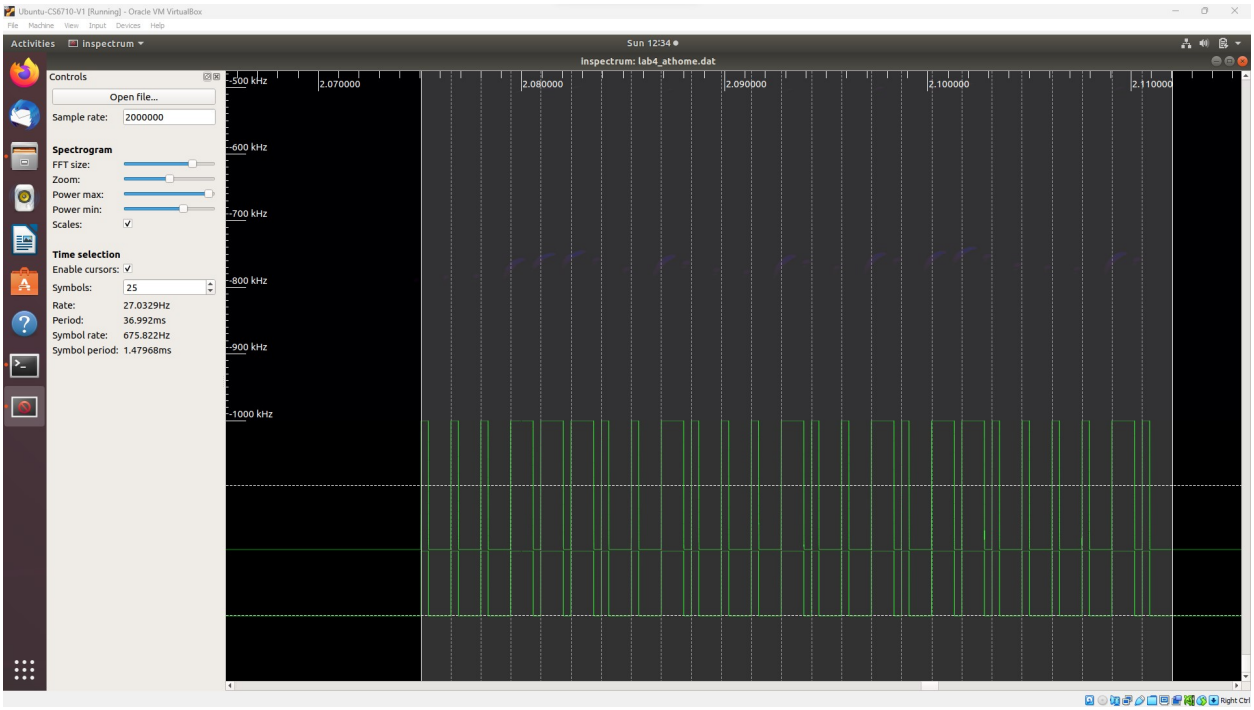
a. On capturing signal for a longer duration we saw it contained packets of 2 different lengths as well in addition to the bell packet.
b. Of these two packets one was 40 symbols and another was 104 symbol long.
c. On exporting symbols of the 40 symbol packet th following sequence was seen:
    i. 0011001001011010011100 1
    ii. 

    iii. On exporting the 104 symbol packet the following sequence was seen:
        1. 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1
    iv. Decoding the sequence to Ascii resulted in: sequence **Oujp-WNDLBJMQ**



    v. This sequence did not make any particular sense and thats when key came handy. Our first inuituion was that some cipher has been applied. The key said 9, we first tried to rotate all letter by 9 but that again gave a vague key. Next thought was to subtract 9 from 26 since english has 26 alphabets and that resulted in 17.
    vi. Rotating each letter by 17 resulted in the final flag: **Flag-NEUCSADH**

BONUS Challenge Earlier you rang the bell by replaying previously captured data. Instead now, use the decoded data (symbol sequence) to generate your packet from the scratch using gnuradio and use it to ring the bell. Hint: Try using vector source block to input the symbols stream. (1 point)



These are the bits that we extracted for the bell that was given to us.
[0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0]

Using these bits above, we constructed the symbols and eventually the signal using the python code for ASK modulation. Please refer to the ipynb file attached for the code.

We then use this data file in the grc source. Please find the grc file attached as well.

(*Demoed to Professor during office hours on Monday, but we can demo to the TAs later again if necessary*)