How do we attack rolling codes?

Rolling code systems are vulnerable to a number of attacks that have been presented over the years. A number of these are attacks on the hardware itself through things like power analysis to determine the seed number or maths algorithm used. One example is the side channel attacks (see https://en.wikipedia.org/wiki/KeeLoq#Side-channel_attacks)
I will only look at the basic attacks that are against the radio transmissions rather than the hardware. In some ways this makes these more generic as they are hardware neutral, and also a lot easier to implement as the hardware and software requirements are a lot lower. The two attacks are one based on the previous replaying and one to look at the vulnerabilities of the rolling code system itself.

Used equipment:

To attempt those experiments you need a device that can record the msg sent from the victim remote and at same time can transmits a jamming signals at multiple frequencies which also should work with FSK AND ASK(OOK) Modulation

For more information on this topic visit:

https://www.andrewmohawk.com/2016/02/05/bypassing-rolling-code-systems/

https://web.stanford.edu/class/ee26n/Assignments/Assignment5.html

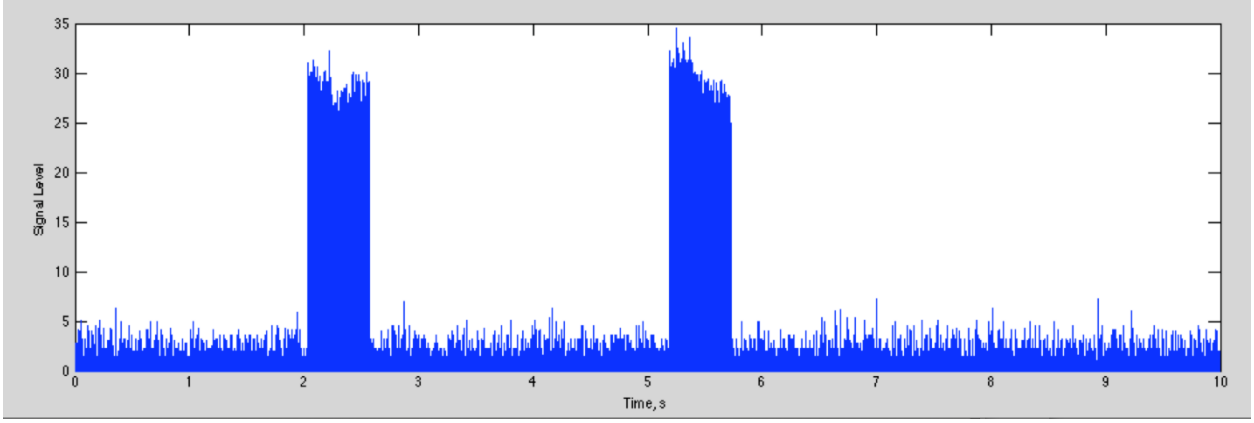
How to know the frequency and transmitted code of the used keyfob:

There are lots of different key fobp systems. We'll start by looking at the key fob for my 2006 Prius. Key fobs use something called a Remote Keyless System (RKS). In the U.S. these operate at 315 MHz, +/- 2.5 MHz. My Prius key turned out to be at 312.590 MHz.

You can figure out what frequency your key fob transmits on using your SDR and use GQRX or SDR# to monitor the spectrum. When you push a button on the fob, you should see a brief jump in the spectrum. You may need to shift the frequency band up or down by a couple of MHz to find the signal, mine was almost 2.5 MHz low.

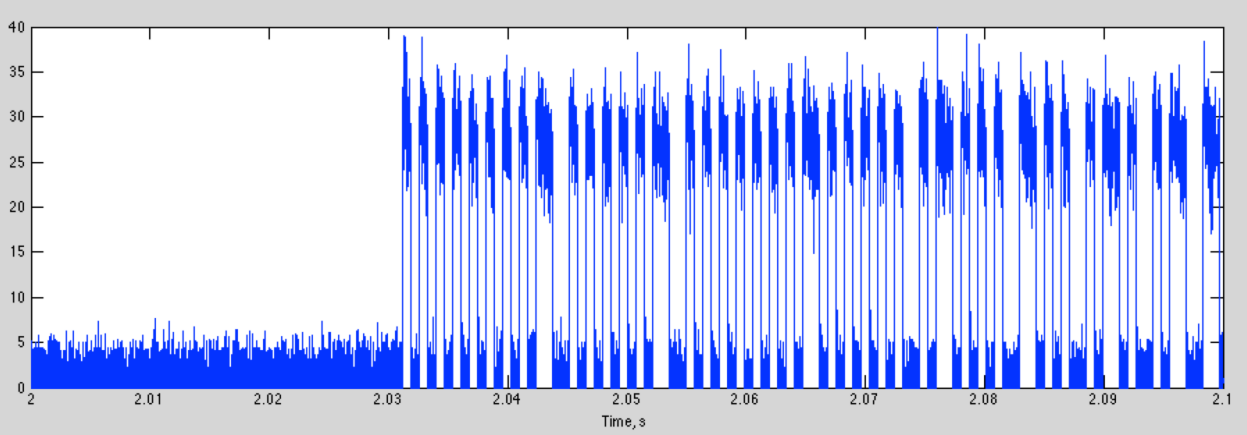
One word of caution. Don't get too carried away pushing the button! The RKS system uses a rolling pseudo-randomly generated code. Both the key fob and the car keep in synch, so that the car recognizes the next code. However, if the key fob gets too far ahead in the sequence (100s of button pushes) the car won't recognize it. That makes the key (and the car) considerably less useful!

If we capture the signal the result is shown below

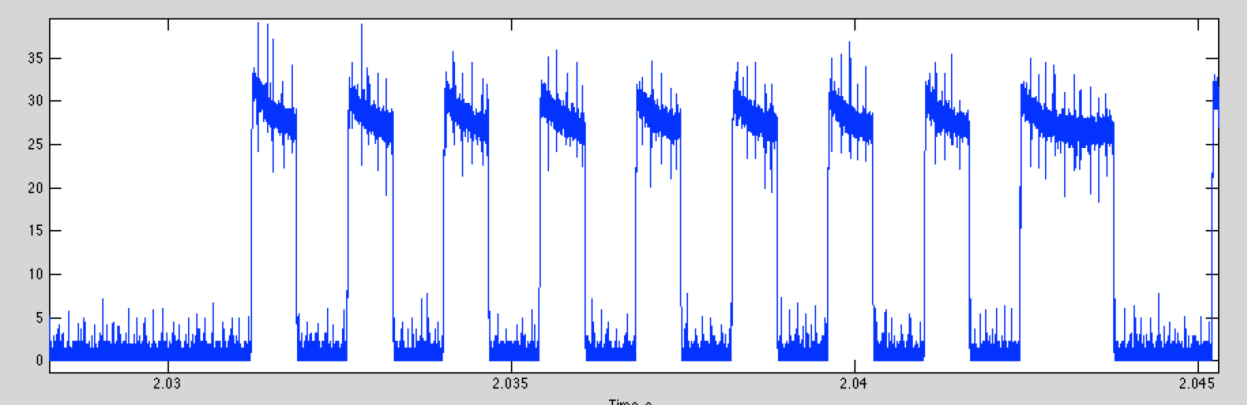


The total width of the plot is 10 seconds, so you can see there is one key press shortly after 2 seconds, and another shortly after 5 seconds.

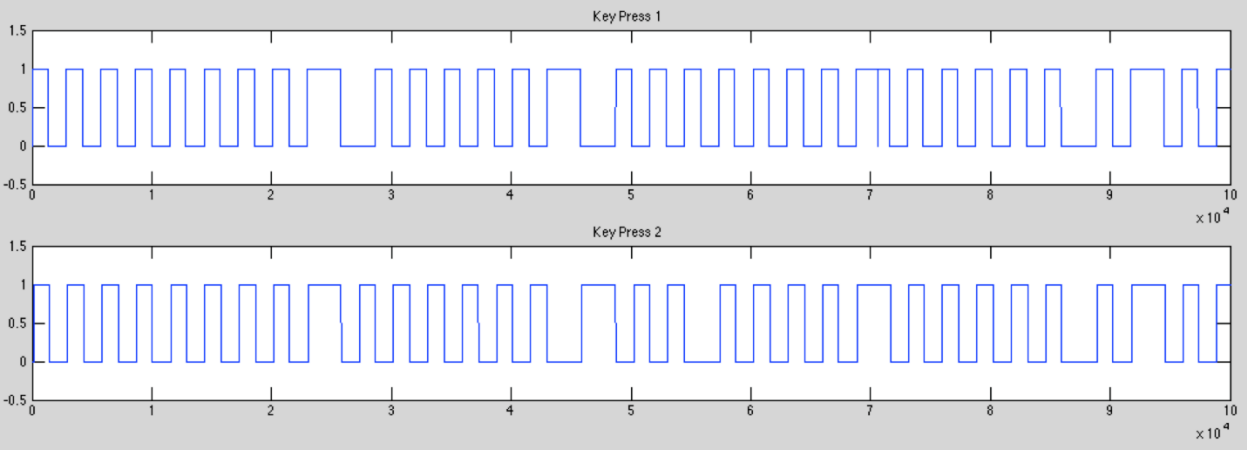
If we plot 100 ms starting at 2 seconds, we can see the digital signal we are looking for:



Zooming in to the first couple of bits, we get



The bits are easy to identify. A decision threshold of 15 will give almost perfect detection. If we do this, and then plot first part of the digital data for the two key presses, we get this



Although the two start the same, they rapidly diverge. This is fortunate, because if the signal was the same every time, you'd have enough information to steal my car now!

The data is again on-off keying (OOK). It is also almost certainly split phase (or Manchester) encoding. Instead of a “1” being high, and a “0” being low, the information is encoding in the transition from high to low or low to high. That means that a “0” bit is a rising transition, and a “1”

bit is a falling transition. A good way to recognize split phase encoding is that you can only have one or two low or high segments in a row. The nice thing about Manchester endcoding is that every symbol has a transition, and these are easier to find then whether the signal has been high or low for several intervals.

This example is OOK, which is the most common for car remotes. Some use frequency-shift keying (FSK), where each bit is transmitted as a different frequency, and the envelope is constant.

So now we can use the recorded data to be retransmitted later to unlock car or decode the msg.

Attack Types:

1-Missing Link Attack (for lack of a better name) or Replay attack :

a-no jamming

b-with jamming

2-Code Grabbing Attack (aka 'Roll Jam')

3-How TO disable a remote control or force the receiver to get out of synch.

4-Attacking Passive Keyless Entry and Start (PKES) Systems.

5-Attacking the Rolling Key System

1-Missing Link Attack:

a-no jamming:

- **Remote control should be out of range of the receiver this condition should be fulfilled for the attack to success.**
- **you press the button of remote and record it.**
- **move to the car or garage door and open**

b-With Jamming:

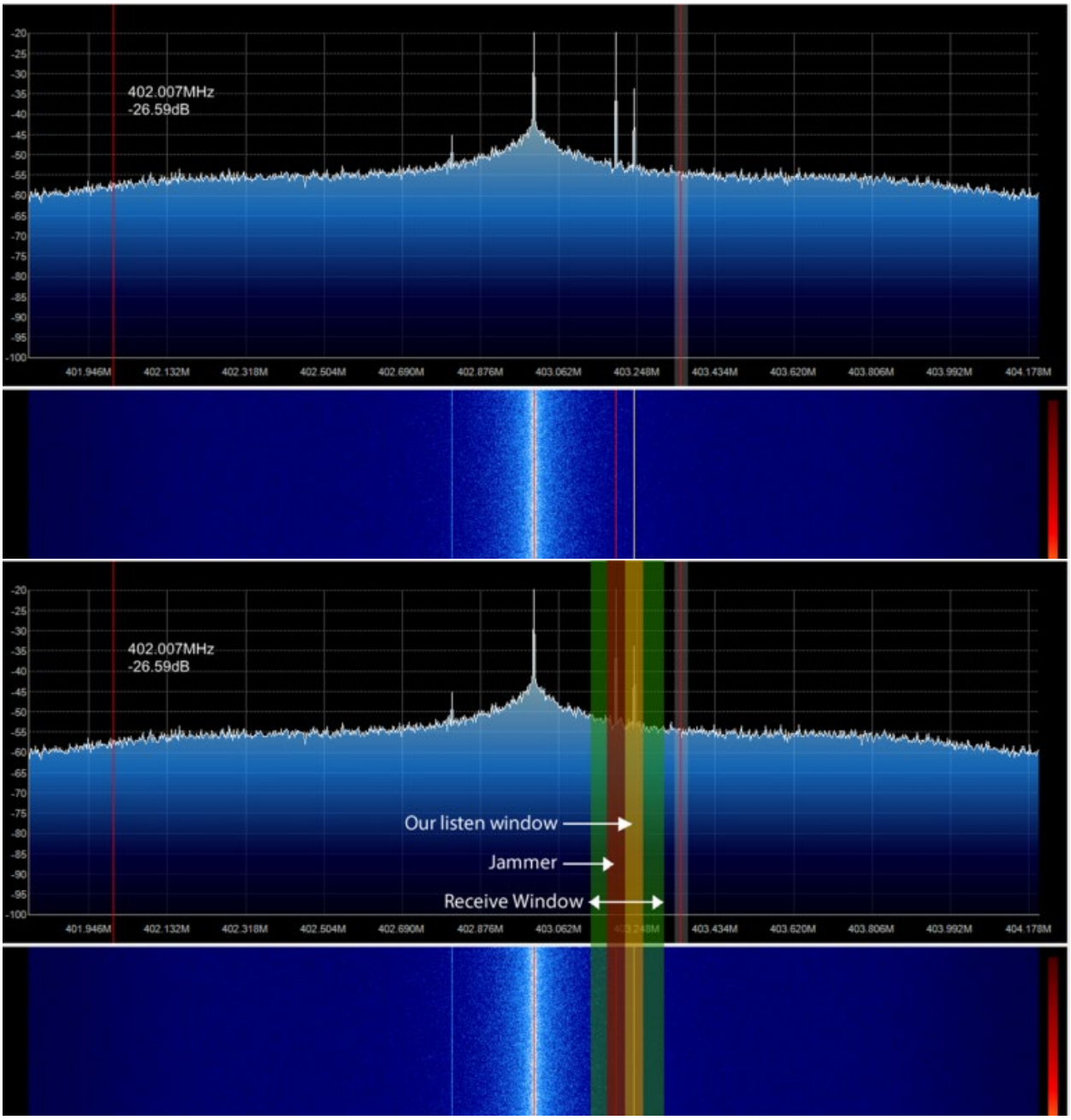
- **Use a jamming device to jam** signal near the vehicle or receiver so the receiver cannot actually 'hear' the code , you can use any remote that works on same frequency since usually receiver will have a wide reception window not only the frequency of the main remote so remote within that receive window will jam signal effectively near the car and prevent it to unlock the car
- **Then record the transmitted msg to then unlock the car or garage door.**
- **Problems:**
  While remote jamming works, it is of course noticable as if the person locking the car simply tests the doors to ensure they are locked they would notice the car unlocked. Additionally if they were aware of such attacks they could even listen to the fact that the doors never made the lock sound or the cars lights never flashed when they pressed the 'lock' button.

Note: this two types of attacks only work one time with rolling code while fixed as much as you like.

2- Code Grabbing Attack (aka 'RollJam'):

This is another technique that if done correctly will be mostly indistinguishable from the standard operating procedure of the car.

- This technique relies on using both the jamming and replaying discussed above. Take a look at the standard signal of a car remote (in this case around 403hmz) of a rolling code lock:



In the image the car remote is effectively being jammed, however at the same time we can also capture the code from the remote. This is because the jamming is not on the exact frequency of the remote. For the party trick, what usually happens when someone cannot get a remote to work is that they will press the button a second time, most people do this instinctively and don't even notice it happening. As soon as this happens we can capture not just a single key press but a secondary one as well.

Armed with two 'valid' codes, what the attacker can then do is transmit the codes back in the captured order (the first captured code sent first), this will perform the action that the target user intended to happen (lock/unlock sequence) and as such the end user will be unaware of anything suspicious happening. However this also means that as an attacker we have a secondary code that is still valid and can be used on the car at a later stage.

Steps:

1. Target parks their car, gets out the car
2. Attacker launches a jammer that prevents the car from receiving the code from the remote
3. Target presses the remote, car does **NOT** lock and the attacker obtains the **first** keypress
4. Target presses the remote a **second** time and the attacker obtains the **second** keypress
5. Attacker then sends the **first** key press to lock the car, car locks as per normal
6. Target assumes all is well and carries on about their day
7. Attacker then sends the **second** keypress to the car, unlocking it
8. Profit.
9. Target returns to the vehicle and remote works as per normal

- Problems:
  Many modern cars have different frequencies for lock and unlock car this means the attacker can only unlock the door on lock frequency if only the system use the same rolling code for lock and unlock the frequencies which is not common for good systems
  One way to solve this is to jam the lock frequency always which is noticeable which will let the victim to lock the car manually or another way around this also that the rolling code implementation on both frequencies is poorly implemented for example and attacker could record the code from the lock frequency with a rolling code x and then send the unlock msg with rolling x using unlock frequency, Other implementations seen in specifications show that the rolling code is a portion of the total code sent. let the code sent is a 24 bit key where the first 12 are the rolling code, the second 8 are the command (such as lock or unlock) and the last 4 is the checksum. Vehicles implementing this type are also naturally susceptible as the attacker merely needs to replace the rolling code segment to be able to use any rolling code on both frequencies which is also a poor implementation of rolling code.

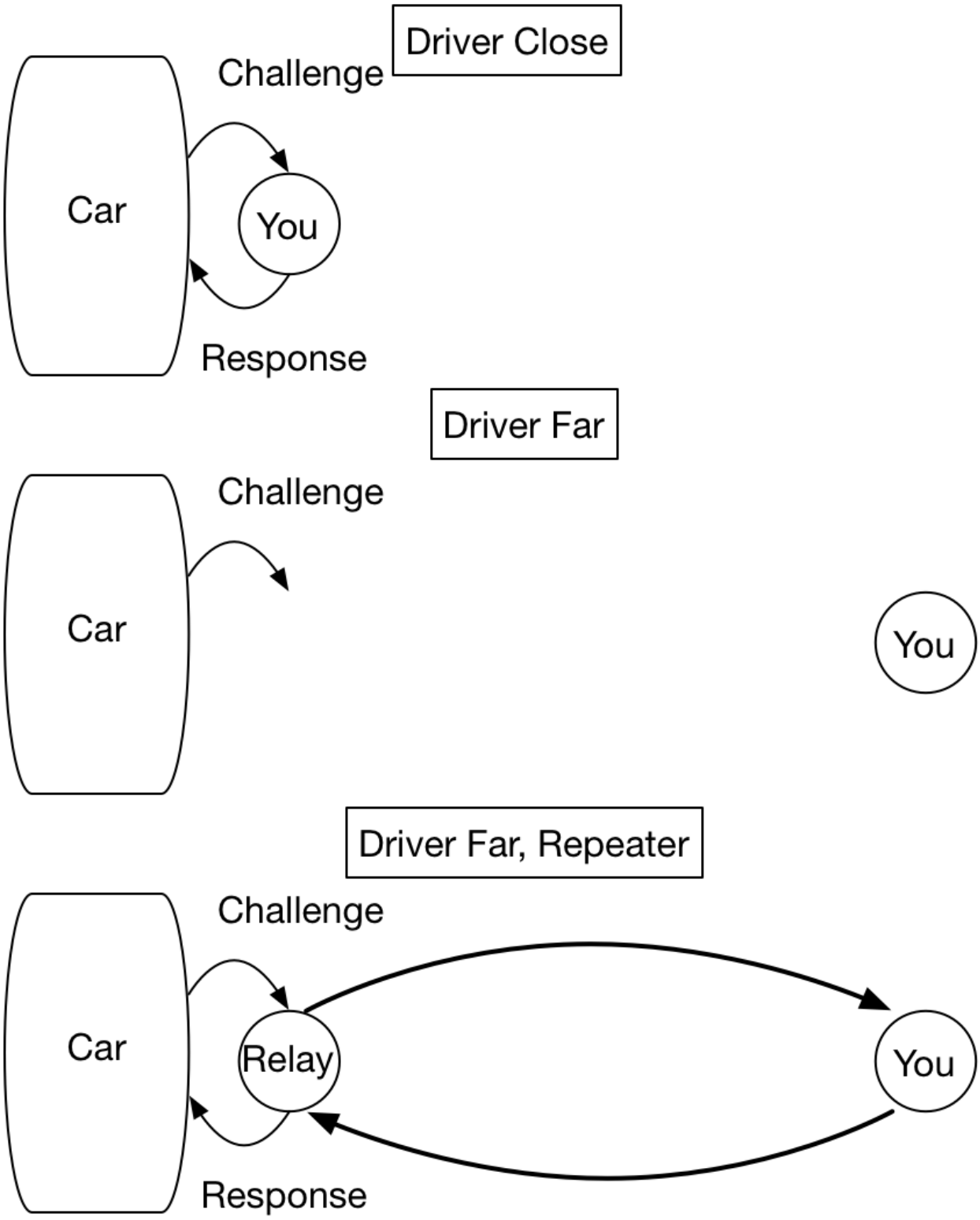
3-How TO disable a remote control or force the receiver to get out of synch:

1. Capture unlock codes
2. Send unlock code **one** (car unlocks)
3. Lock car with specific VW remote
4. Send unlock code **two** (car does not unlock)
5. **Remote no longer functional for locking/unlocking car**

4-Attacking Passive Keyless Entry and Start (PKES) Systems:

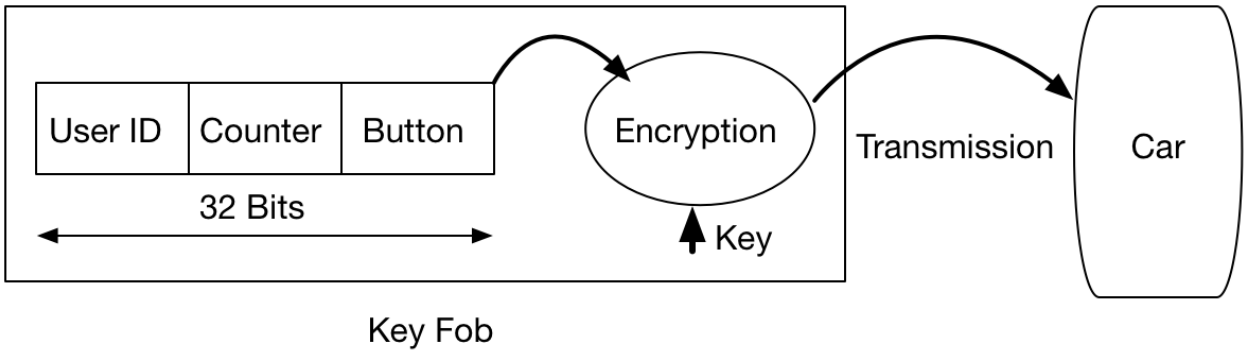
Many higher end cars use a passive system for opening the car when the driver approaches. A low power signal is transmitted from the car as a challenge. The key fob then responds with an authentication. Because the power is so low, the car assumes the driver must be in close proximity if it receives a response.

These systems can be hacked by building a repeater that placed near the car. It captures the car's signal and retransmits it at higher power. The remote can be anywhere with in a couple hundred meters, and it will still hear it. The remote responds, and that is again captured by the repeater, and retransmitted.


Driver Close
Challenge
Car
You
Response
Driver Far
Challenge
Car
You
Driver Far, Repeater
Challenge
Car
Relay
You
Response

5-Attacking the Rolling Key System:

The next attacks go after the rolling key system itself. The way this generally works is that the key fob sends an ID, along with a counter of how many times a key has been pressed. This is encrypted, and transmitted to the car when you push the button.



If the encryption is strong, it is extremely difficult to figure out what the userid and counter is. There are several interesting cases. One is for the last 20 years of VW's (and Audi's, Porsche's, etc), that we'll look at here. Another is for Subarus, that you can look at for your assignment this week.

How to protect your self against such attacks:

- **2 different frequencies**
- **2 way communication** means that you can set up a secure session between the remote and the car , this however is far more expensive to implement and has its own problems .
- **Codes that expire**
  Because there is no timeout on the codes it means that an attacker can use these at any stage. However implementing a timeout on the code means that should you be away from your vehicle for a while, say on holiday, your remotes would lose sync.
- **Code Hopping / FHSS**
  Implementing these methods on how the codes are sent on the physical level means that it is a lot harder to jam and to transmit valid codes. This would set the barrier to having hardware that can match the vehicles and remotes much higher and hopefully mitigate a number of attacks (for a brief period of time).
- **Smaller Receive Windows**
  Having higher quality components means that the receive window could be made much smaller
- **Lower Transmission power (Low range operation):**
  Which means the car will only unlock on very close range will make unlocking the car more difficult for the hacker .