Enhanced Vehicular Roll-Jam Attack using a Known Noise Source

Zachary Depp
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, OH, USA
depp.18@osu.edu

Halit Bugra Tulay
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, OH, USA
tulay.1@osu.edu

C. Emre Koksal
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, OH, USA
koksal.2@osu.edu

***Abstract*—The traditional vehicular roll-jam attack is an effective means to gain access to the target vehicle by jamming and recording key fob inputs from a victim. However, it requires specific knowledge of the attack surface, and delicate tuning of software-defined radio parameters. We have developed an enhanced version of the roll-jam attack that uses a known noise signal for jamming, in contrast to the additive white Gaussian noise that is typically used in the attack. Using a known noise signal allows for less strict tuning of the software-defined radios used in the attack, and allows for digital noise removal of the recorded input to enhance the replay attack.**

Keywords*—*software-defined radio, roll-jam, wireless security of vehicular systems

I. Introduction

Fundamentally, the vehicular roll-jam attack works by having an adversary target a victim whose vehicle they want to access without authorization. In the attack model, the adversary jams and records the signals transmitted from a key fob to access a target vehicle. It was developed specifically to defeat the rolling code security measures that modern vehicles use to protect against normal replay attacks. The vehicular roll-jam attack has been around publicly since at least 2015, and has proven to be situationally effective at gaining unauthorized access to modern vehicles that use key fob rolling code security [1]. The increased availability of software-defined radios (SDR) to hobbyists have made this attack well-known, although it has not fundamentally changed since it first emerged, and car manufacturers have yet to implement any kind of real mitigation strategy against it.

The attacker uses one or more SDRs to send a jamming signal to the vehicle to block the reception of legitimate key fob inputs, while simultaneously recording that legitimate input, typically an unlock signal, with the intention of replaying it at a later time to gain access to the vehicle, as seen in Fig. 1. This attack bypasses the rolling code security of the key fob, which synchronizes key fob inputs with a cryptographic counter that is shared with the vehicle's onboard computer. The vehicle will interpret the replayed input as legitimate since it has yet to receive that message, and unlock the vehicle. There have been several proposed defense strategies against this attack, including adding timestamps to the rolling code, but vehicle manufacturers have yet to make any widespread changes [2], [3].

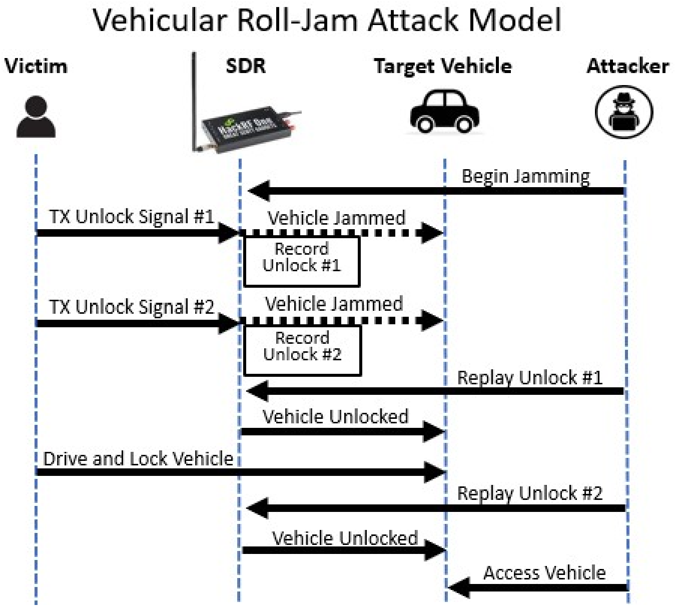


Fig. 1: Roll-Jam Attack Model.

While the roll-jam is a well-known attack, it still requires information from the victim before the adversary can execute

Symposium on Vehicles Security and Privacy (VehicleSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-88-6
https://dx.doi.org/10.14722/vehiclesec.2023.23037
www.ndss-symposium.org

the attack. The attacker needs to know the exact frequency that the vehicle key fob operates at, and then must adjust their SDR to jam either slightly above or below that frequency and then must find an appropriate level of transmit gain for the noise signal such that the vehicle is jammed, but not so much that it renders the captured key fob signal unusable in the subsequent replay attack. Certain modern vehicles are also starting to incorporate anti-theft security features which can prevent the vehicle from receiving any key fob inputs if it receives an already used code. This means if the attack is not executed perfectly the first time, further attempts are blocked. This tuning and configuration of the SDR can take a significant amount of time, during which the attacker could lose their window of exploitation.

In this paper, we propose an enhanced roll-jam attack that uses a known noise sequence at the exact same frequency as the key fob. Unlike the traditional roll-jam, our new attack does not require prior knowledge of the key fob signal. It conceptually works for any signal, and even those with modern encryption practices with rolling codes. Our new attack method allows us to jam the vehicle at the exact frequency with even higher transmit power than the traditional roll-jam attack. Subsequently, we record the key fob input and perform noise removal techniques to obtain the original input signal. The obtained signal is later replayed to gain access to the target vehicle.

II. Enhancing the Roll-Jam Attack

One of the biggest setbacks with the traditional roll-jam attack is that it requires simultaneous jamming and recording within a relatively narrow bandwidth, usually 1.5MHz, in the spectrum of either 315MHz or 433MHz [4]. The jamming signal is usually additive white Gaussian noise (AWGN), as it is effective at jamming over a narrow bandwidth and is easy to generate with most SDR software [5]. If the attacker jams at a frequency further than 1.5MHz from the key fob operating frequency, however, they run the risk of jamming outside of the receive window of the vehicle, and not jamming the vehicle at all. If the attacker jams too closely to the key fob frequency, they risk distorting the recorded signal to the point of being unable to replay it later.

Modern vehicles have been shown to be incredibly vulnerable to being wirelessly jammed by a variety of techniques [6]. In our approach, the attacker uses a known noise sequence transmitted at the same frequency as the key fob for jamming. Since the noise sequence is known, the attacker uses noise removal techniques to maintain a sufficient signal-to-noise ratio (SNR) that enables the signal to be replayed. By removing the noise component from the recorded message, the attacker could replay the attack from much further away and with greater efficiency. The noise source being known also gives the attacker greater flexibility in the amount of power they use to jam the target vehicle. As the noise signal is known to them, they will be able to identify and digitally remove noise sources transmitted at higher power than the traditional attack. This enables the attacker to have greater confidence that the target vehicle is in fact being jammed from receiving legitimate messages, and potentially interrupting the attack. Fig. 2 details our enhanced roll-jam model.

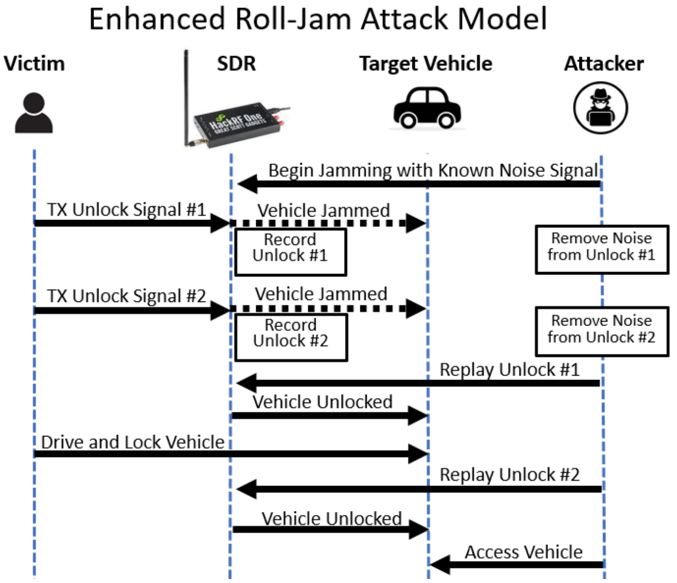


Fig. 2: Enhanced Roll-Jam Attack Model.

The attacker begins jamming the target vehicle with the known noise signal as soon as they are in position to wait for the victim to attempt to unlock their vehicle. Once they capture the unlock signals, they immediately digitally remove the noise component from them, and then carry out the replay attack using the enhanced messages. Due to the noise removal process, the vehicle is more likely to accept the replayed signals as legitimate, and give the attacker access to their target. We propose to use a legitimate key fob message as a template for the noise source. However, simply capturing and replaying old signals multiple times is not a viable option, due to anti-theft security features on certain modern vehicles. These features automatically lock-down the car and its accompanying key fob if the car receives previous rolling code messages. These constraints meant we had to record a legitimate signal, and then modify it sufficiently such that it was efficient at jamming the vehicle and was easily identifiable by the attacker.

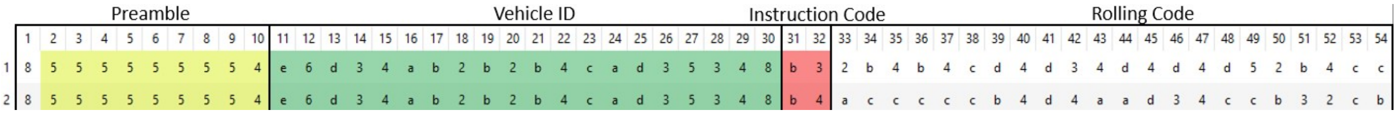
III. Evaluating the Enhanced Roll-Jam Attack

A. *Creating the Known Noise Source*

The first step to generate the known noise signal was to capture several legitimate key fob messages from a modern vehicle that used rolling code security. For the purposes of this research, a 2020 Kia Sorento EX was used, and the software Universal Radio Hacker (URH) was used in conjunction with a Great Scott Gadgets HackRF One as our SDR for collecting, analyzing, and replaying data [7]. Fig. 3 and Fig. 4 show several captured and demodulated signals from the target vehicle's key fob in hexadecimal form.

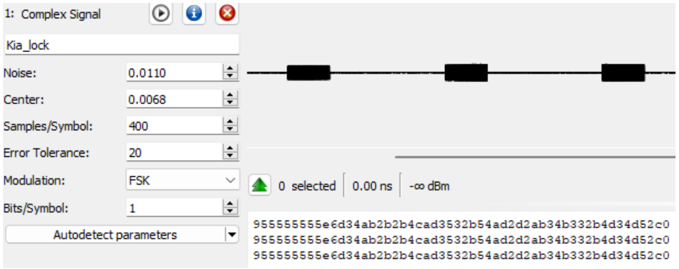
This key fob operates in the 433MHz range, specifically at 433.92MHz, and uses frequency-shift keying for modulation. URH has a helpful auto-detection setting that attempts to determine the signal parameters as long as the SNR is sufficiently high enough.

The key fob transmits three identical message pulses, each separated by approximately 120ms. Fig. 3 shows the demodu-



	Preamble										Vehicle ID																				Instruction Code		Rolling Code																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
1	8	5	5	5	5	5	5	5	5	4	e	6	d	3	4	a	b	2	b	2	b	4	c	a	d	3	5	3	4	8	b	3	2	b	4	b	4	c	d	4	d	3	4	d	4	d	4	d	5	2	b	4	c	c
2	8	5	5	5	5	5	5	5	5	4	e	6	d	3	4	a	b	2	b	2	b	4	c	a	d	3	5	3	4	8	b	4	a	c	c	c	c	c	b	4	d	4	a	a	d	3	4	c	c	b	3	2	c	b

Fig. 3: Breakdown of demodulated hexadecimal key fob message.



1: Complex Signal

Kia_lock

Noise: 0.0110

Center: 0.0068

Samples/Symbol: 400

Error Tolerance: 20

Modulation: FSK

Bits/Symbol: 1

Autodetect parameters

0 selected | 0.00 ns | -∞ dBm

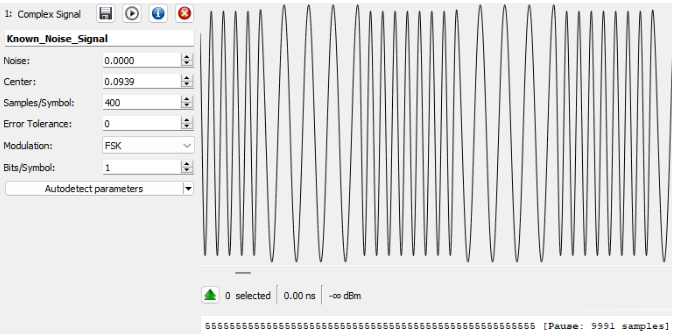
```
955555555e6d34ab2b2b4cad3532b54ad2d2ab34b332b4d34d52c0
955555555e6d34ab2b2b4cad3532b54ad2d2ab34b332b4d34d52c0
955555555e6d34ab2b2b4cad3532b54ad2d2ab34b332b4d34d52c0
```

Fig. 4: Demodulated key fob message from 2020 Kia Sorento.

lated frames of data consist of a preamble, vehicle ID number, instruction code, and the rolling code. Each individual message is 54 hexadecimal digits long, and each button press on the key fob generates three new identical pulses. From here, URH has a function which allows generation of new data frames by using captured data and copying the modulation technique, carrier frequency, sample rate, and symbol size. Now we are free to change any of the bits in the message, and then compile a brand new payload consisting of specific data. To make visual analysis of the decoded signal easier, we decided on generating a known noise payload consisting entirely of hexadecimal 5, as seen in Fig. 5. The actual demodulated bit values of the known noise signal do not matter as long as they are known to the attacker and do not match an old rolling code message from the vehicle. From here, we can transmit this signal indefinitely from the SDR to act as the jamming signal in the enhanced attack.

B. Executing the Enhanced Roll-Jam Attack

After generating the known noise sequence, we can execute the entire enhanced roll-jam attack. The first part of the attack is carried out almost identically to the traditional roll-jam. The attacker selects a target vehicle, the 2020 Kia Sorento in this case, and transmits the known noise signal to jam the vehicle from one SDR while simultaneously recording the legitimate key fob input. The differences are that instead of a randomly generated noise source, the known noise source is used, and the noise is transmitted at the exact frequency of the key fob, 433.92MHz. The URH output in Fig. 6 shows what this captured input looks like alongside the known noise signal.



1: Complex Signal

Known_Noise_Signal

Noise: 0.0000

Center: 0.0939

Samples/Symbol: 400

Error Tolerance: 0

Modulation: FSK

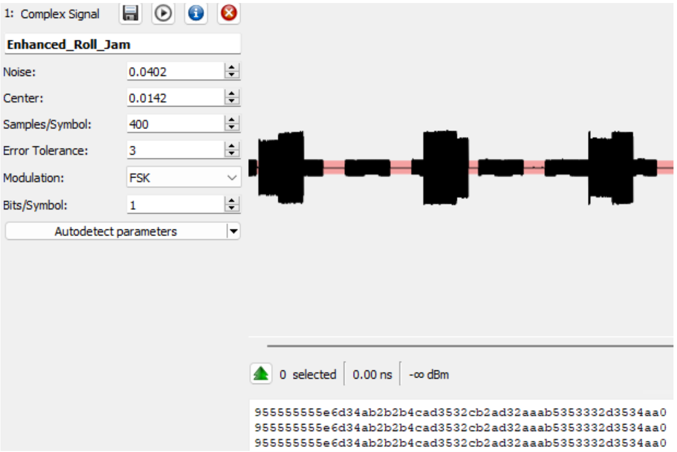
Bits/Symbol: 1

Autodetect parameters

0 selected | 0.00 ns | -∞ dBm

```
555555555555555555555555555555555555555555555555555555 [Pause: 9991 samples]
```

Fig. 5: Known noise signal generated from legitimate message.



1: Complex Signal

Enhanced_Roll_Jam

Noise: 0.0402

Center: 0.0142

Samples/Symbol: 400

Error Tolerance: 3

Modulation: FSK

Bits/Symbol: 1

Autodetect parameters

0 selected | 0.00 ns | -∞ dBm

```
955555555e6d34ab2b2b4cad3532cb2ad32aaab5353332d3534aa0
955555555e6d34ab2b2b4cad3532cb2ad32aaab5353332d3534aa0
955555555e6d34ab2b2b4cad3532cb2ad32aaab5353332d3534aa0
```

Fig. 6: Key fob message recorded while jamming with known noise source.

For our experiments, the jamming SDR was placed directly next to the receiving SDR, and the key fob was placed close to the SDRs in order to keep our transmitting power at a reasonably low level, as seen in Fig. 7. After recording the key fob input while jamming, the URH autodetect function was able to automatically isolate the key fob message from the known noise signal. Even if the autodetect function had not worked, the attacker would be able to visually locate the captured message and could then manually adjust the parameters to fully isolate the input. With the noise floor set to the maximum amplitude of the known noise source, the attacker is left with the key fob message alone in its entirety.

The next step is to generate a new payload for the replay portion of the attack using URH, with a similar method as was used to generate the known noise source. After removing the noise in URH, the outcome is a message signal with the entirety of the noise component removed, as seen in Fig. 8. The attacker can then transmit this noise-removed signal when they want to access the victim's vehicle. With the noise component completely removed from the payload, the SNR is improved significantly, and the attacker has the ability to replay the message with more flexibility.

In the traditional roll-jam attack, the replayed message still contains the added noise component that the attacker used to jam the vehicle. While the jamming is at an adjacent frequency to the key fob frequency, the sidebands generated can be significant, and makes the replay attack difficult to alter if the attacker needs to transmit with additional power. The enhanced attack allows for the message to be replayed at even higher power than the original key fob recording, and from a further range than the key fob's operating distance.

Our experiments have demonstrated that this attack works on every vehicle we have had available for testing. This includes vehicles with key fob frequencies operating in the 315MHz range, and that use amplitude-shift keying for modulation rather than frequency-shift keying. The attack performs with highest efficiency when the jamming SDR is closer in proximity to the vehicle than the recording SDR and key fob, but the attack also works well even when the jamming SDR, recording SDR, and key fob are co-located. Fig. 9 shows the frequency domain signals before and after jamming. This attack has been tested and verified successful on the following modern vehicles from the United States, Asian, and European markets:

- 2013 Ford F-150
- 2015 Honda HRV
- 2015 Nissan Rogue
- 2015 Audi A3
- 2020 Kia Sorento
- 2020 Toyota Tacoma

The traditional roll-jam works situationally on these vehicles as well, however, significant configuration changes to the SDR are required for every different vehicle. With the enhanced attack, all the adversary needs to know is the key fob frequency and then they can implement the attack with a high rate of success.

C. *Comparing the Traditional and Enhanced Attacks*

To directly compare our enhanced roll-jam attack with the traditional version, we used the Linux-based GNU Radio Companion (GRC), a framework that contains signal processing blocks for SDRs [8]. Fig. 10 depicts the flow chart derived for this analysis.

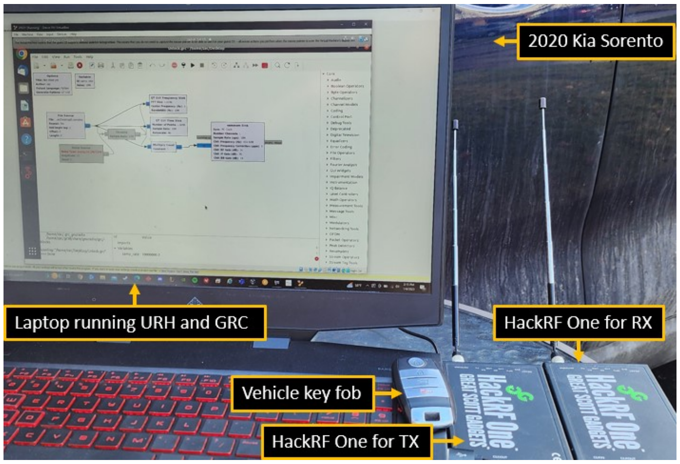


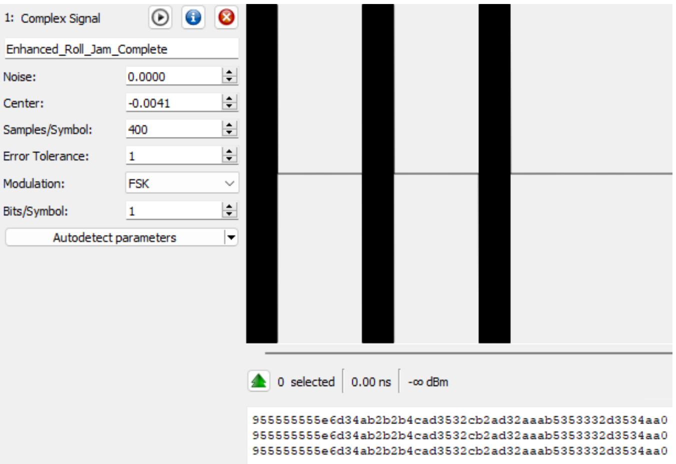
Fig. 7: Executing the enhanced roll-jam on a 2020 Kia Sorento.

GRC contains a block for generating AWGN, which we use to compare against our generated known noise source. For our first analysis, we measure the average SNR of a captured key fob signal when using AWGN and the known noise sequence to jam. We also measure the SNR of the complete enhanced roll-jam attack after we have used URH to remove the noise, leaving just the key fob message. While the SNR of the attack is not completely indicative of its success, attacks with a higher SNR have more flexibility to replay the attack under conditions favorable to the attacker, such as being able to unlock the target vehicle from a further distance, and giving the attacker more confidence that the attack will succeed.

The HackRF operates in half-duplex, so two were used to collect this data, one for transmitting the noise source using GRC, and one for collecting the key fob input using URH. Both SDRs used the same low power settings as seen in Fig. 10 in the osmocom Sink block. For each noise type, ten consecutive unlock signals were sent from the key fob and captured by the SDR connected to URH. An average SNR was calculated using the analysis tools available in URH, and the results are shown in Table 1. We observe that using AWGN as the noise source provided slightly better results than when just using the known noise sequence alone. However, when implementing the noise removal in the enhanced attack, the SNR is significantly higher, as the only noise component remaining is the ambient noise of the environment. This allows the attacker to easily replay the signal from further away and with a high degree of confidence.

For the next analysis, we compare the highest level of transmit gain for each noise type that still allows for noise removal in URH. The setup for data collection is the same as the previous analysis, and we simply raise the transmitter power in GRC until URH could no longer automatically detect the captured key fob input apart from the noise.

The HackRF has two transmitter gain settings that can be adjusted, a radio frequency (RF) gain and an intermediate frequency (IF) gain. The RF gain controls the front-end amplifier of the HackRF, and it is either on or off with gain values of 0dBm and 14dBm, respectfully, and the IF gain can be



Enhanced_Roll_Jam_Complete

Noise: 0.0000
Center: -0.0041
Samples/Symbol: 400
Error Tolerance: 1
Modulation: FSK
Bits/Symbol: 1

```
955555555e6d34ab2b2b4cad3532cb2ad32aaab5353332d3534aa0
955555555e6d34ab2b2b4cad3532cb2ad32aaab5353332d3534aa0
955555555e6d34ab2b2b4cad3532cb2ad32aaab5353332d3534aa0
```

Fig. 8: Recorded key fob message with noise component removed.

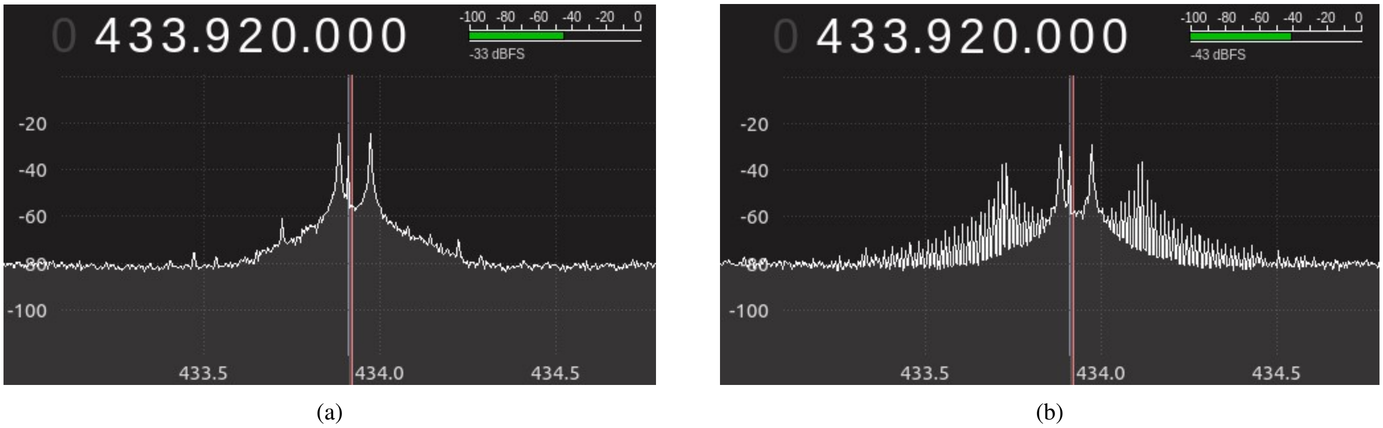
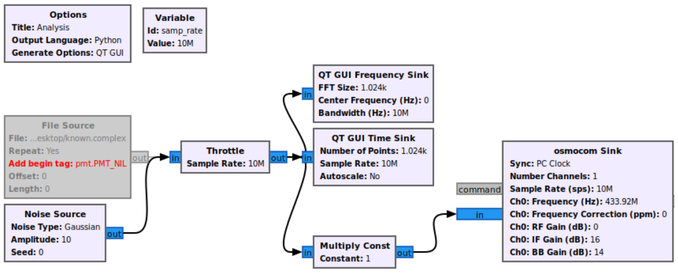


Fig. 9: a) Frequency domain key fob signal before jamming b) Key fob signal while jamming with known noise source.


Options
Title: Analysis
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 10M

File Source
File: ...esktop/known.complex
Repeat: Yes
Add begin tag: pmt.PMT_NIL
Offset: 0
Length: 0

Noise Source
Noise Type: Gaussian
Amplitude: 10
Seed: 0

Throttle
Sample Rate: 10M

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 10M

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 10M
Autoscale: No

Multiply Const
Constant: 1

osmocom Sink
Sync: PC Clock
Number Channels: 1
Sample Rate (sps): 10M
Ch0: Frequency (Hz): 433.92M
Ch0: Frequency Correction (ppm): 0
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 16
Ch0: BB Gain (dB): 14

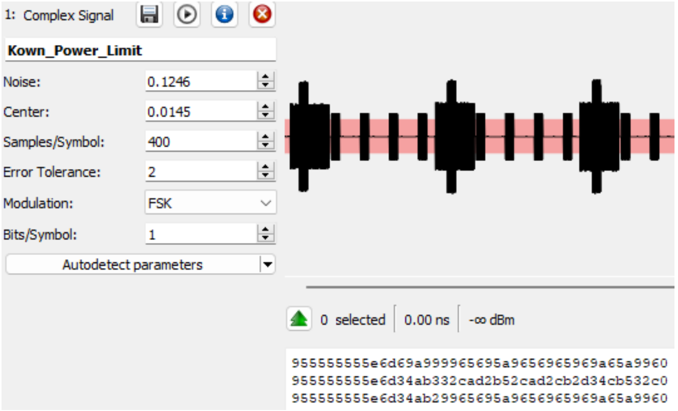
Fig. 10: GNU Radio Companion Block Diagram used for Analysis.

set from 0dBm to 47dBm [9]. Initial testing confirmed that both the AWGN and known noise sequence begin effectively jamming the vehicle at the same transmit power from the same distance. For this analysis, the front end amplifier was turned on for both noise sources, and the IF gain was adjusted for comparison.

The AWGN source had a maximum transmit gain of 17dBm before the signal became undecipherable by URH. The key fob signal is still visibly recognizable on the recording, but above 17dBm the noise distorts the signal beyond recognition. Any recorded sequence above this gain threshold is not suitable for noise removal in URH. The known noise sequence, however, could be transmitted at up to 26dBm before URH was unable to detect a message, as seen in Fig. 11. This 9dBm difference represents the ability to transmit the known noise sequence with approximately eight times more power than the AWGN signal. This allows the attacker to jam at higher power and have greater confidence that the vehicle is in fact being jammed.

TABLE I: Signal-to-Noise Analysis

	Attack Type		
	AWGN Noise Source	*Known Noise Source*	*Enhanced Attack*
SNR (dB)	8.786	7.899	40.115



Kown_Power_Limit

Noise: 0.1246
Center: 0.0145
Samples/Symbol: 400
Error Tolerance: 2
Modulation: FSK
Bits/Symbol: 1

```
955555555e6d69a999965695a9656965969a65a9960
955555555e6d34ab332cad2b52cad2cb2d34cb532c0
955555555e6d34ab29965695a9656965969a65a9960
```

Fig. 11: URH detects key fob message with known noise transmitted at 26dBm.

While executing the entire enhanced attack takes longer than the traditional roll-jam due to the noise removal and signal generation process in URH, attackers would generally perform the collection part of the roll-jam attack first, and then execute the replay portion of the attack at a later time when the vehicle is unattended. This means that there is no loss in attack efficacy as long as the replay occurs sometime after the collection process. The collection process itself is greatly improved by being able to jam the vehicle with an appropriate level of noise as soon as the attacker knows the key fob frequency.

IV. Conclusion

In this paper, we propose the enhanced vehicular roll-jam attack that uses a known noise source. We demonstrate the effectiveness of the attack on different vehicles using a software-defined radio. Specifically, we show a significant SNR improvement over the traditional roll-jam attack. This provides the adversary incredible flexibility to carry out the attack without requiring prior knowledge of the transmitted signal. While AWGN serves as an appropriate noise source for the attack, jamming becomes significantly more potent when using a noise source created and known by the attacker. Indeed, cryptographic security approaches will not be able to mitigate this new attack, as we have shown that the key fob signal can be decoded simultaneously during smart jamming in a full-duplex like operation. These types of attacks will remain prevalent in our society as long as vehicle manufacturers are unable to update their security mechanisms to defend against them.

REFERENCES

[1] C. Kraft, “Anatomy of the Rolljam Wireless Car Hack,” Make: DIY Projects and Ideas for Makers, Aug. 11, 2015. https://makezine.com/article/maker-news/anatomy-of-the-rolljam-wireless-car-hack/ (accessed Jan. 04, 2023).

[2] K. Greene, D. Rodgers, H. Dykhuizen, K. McNeil, Q. Niyaz and K. A. Shamaileh, ”Timestamp-based Defense Mechanism Against Replay Attack in Remote Keyless Entry Systems,” 2020 IEEE International Conference on Consumer Electronics (ICCE), 2020, pp. 1-4, doi: 10.1109/ICCE46568.2020.9043039.

[3] K. Greene, D. Rodgers, H. Dykhuizen, Q. Niyaz, K. Al Shamaileh and V. Devabhaktuni, ”A Defense Mechanism Against Replay Attack in Remote Keyless Entry Systems Using Timestamping and XOR Logic,” in IEEE Consumer Electronics Magazine, vol. 10, no. 1, pp. 101-108, 1 Jan. 2021, doi: 10.1109/MCE.2020.3012425.

[4] G. Nespral, “How to hack a car,” Hackaday.io, Mar. 26, 2019. https://hackaday.io/project/164566-how-to-hack-a-car/details. (accessed Jan. 04, 2023).

[5] “Noise and dB — PySDR: A Guide to SDR and DSP using Python,” pysdr.org. https://pysdr.org/content/noise.html (accessed Jan. 04, 2023).

[6] Y. O. Basciftci, F. Chen, J. Weston, R. Burton and C. E. Koksal, ”How Vulnerable Is Vehicular Communication to Physical Layer Jamming Attacks?,” 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), 2015, pp. 1-5, doi: 10.1109/VTCFall.2015.7390968.

[7] J. Pohl, “Universal Radio Hacker,” GitHub, Mar. 22, 2022. https://github.com/jopohl/urh/ (accessed Jan. 04, 2023).

[8] “GNU Radio - The Free Open Source Radio Ecosystem” GNU Radio. https://www.gnuradio.org/ (accessed Jan. 04 2023).

[9] M. Ossmann, “HackRF Documentation,” hackrf.readthedocs.io, Sep. 22, 2022. https://hackrf.readthedocs.io/en/latest/index.html (accessed Jan. 04, 2023).