1 (2018 USENIX)Injected and Delivered: Fabricating Implicit Control over Actuation Systems by Spoofing Inertial Sensors

1.1 Summary

In this paper, the author utilize the resonance effect of MEMS inertial sensors to devise two different non-invasive spoofing attack toward embedded sensors. To conduct these attacks, the authors analyze the sampling process and find the way to control the digital signal output via changing the acoustic signal's amplitude, frequency and phase. To assess the performance of these attacks, the authors evaluate their attacks on 25 different devices and find 23 of them affected by the acoustic signal.

1.2 Challenge

Previous works have utilized the ultrasound signal to attack the MEMS sensors in UAVs, however, they have a couple of deficiencies which are the main challenge of this paper.

1. The output of the inertial sensors are digital signals sampled from the analog signal generated by resonance and in real scenarios the sample rate of the ADC converters have some drift. The authors find that even little drift can affect the output of sensors significantly. So to control the output, the drift in sampling rate must be considered.

1.3 Main Idea

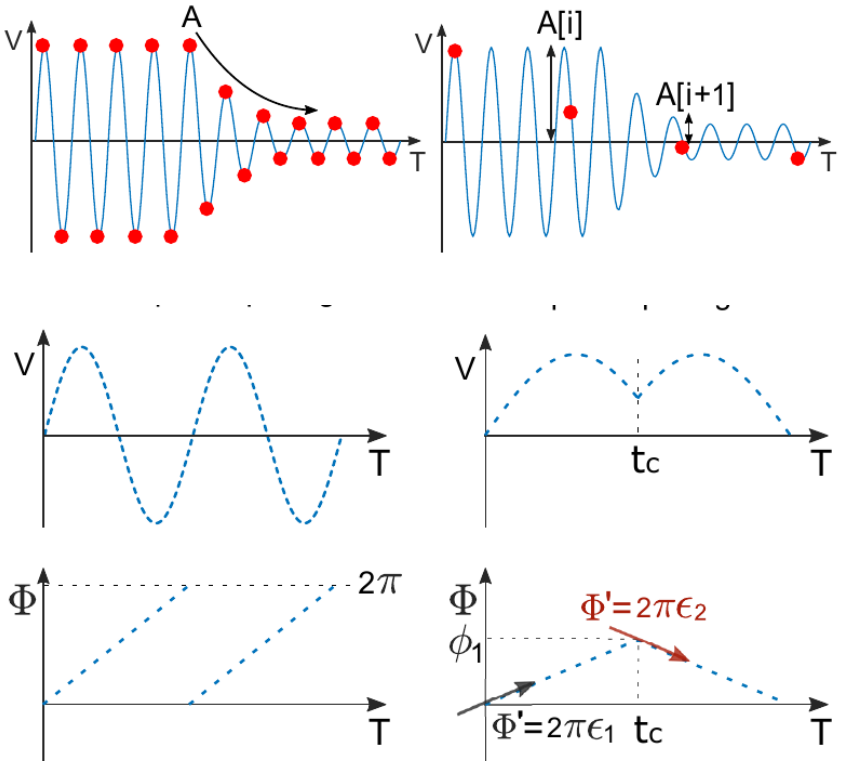
1.3.1 Attack Model

The oscillation induced by resonant sound wave has the same frequency with it and there is an digitization process before generating the final reading of the sensor. Because the acoustic signal belongs to out-of-band signal, its frequency is much higher than the sampling rate which leads to the alias effect. The digitization signal can be illustrated below:

$$V[i] = A \cdot sin(2\pi\varepsilon\frac{i}{F_s} + \phi_0)$$

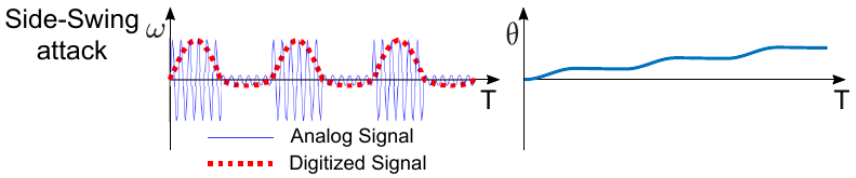
where $F = n \cdot F_s + \varepsilon(-0.5 \cdot F_s < \varepsilon < 0.5 \cdot F_s)$ is the frequency of digital signal after alias. From the formulation above we can find out that if the sampling rate F_s increase with a small amount δF_s , the change in the frequency of

the digital signal is n times more than it. So even little offset in sampling rate can cause significant affect. we can also manipulate the amplitude and phase of the digital signal by change the amplitude and frequency of acoustic signal respectively as the picture show below.



1.3.2 Attack Methods.

Based on the analysis on the digitization process, the authors propose two kinds of attacks named side-swing attack and switching attack respectively. Because the pattern of the oscillation is a sinusoid signal, the accumulative offset induced by it is 0. So the basic idea of these two attack is trying to keep the digital in one direction and decrease it in the opposite orientation. In side-swing attack, the authors modulate the analog signal with two amplitude. When the digital signal is in target direction, the higher amplitude is used and when in inverse direction the other is used. The picture below show this process.



1.3.3 Evaluation

To evaluate the effect of the embedded inertial sensor spoofing attack, The author conduct experiments on 25 different sensors under close-loop and open-loop systems. The result of the experiment can be shown in the two tables below.

Device	Sensor		Resonant	Affected/	Max	Control
	Type	Model†	Freq. (kHz)	Func. Axes	Dist. (m)	Level
Megawheels scooter	Gyro	IS MPU-6050A	27.1~27.2	y/y	2.9	Implicit control
Veeko 102 scooter	Gyro	Unknown	26.0~27.2	x/x	2.5	Implicit control
Segway One S1	Gyro	Unknown	20.0~20.9	x/x	0.8	Implicit control
Segway Minilite	Gyro	Unknown	19.2~20.0	x/x	0.3	DoS
Mitu robot	Gyro	N/A SH731	19.0~20.7	x/x	7.8	Implicit Control
MiP robot	Acce	Unknown	5.2~5.4	x/x	1.2	DoS
DJI Osmo stabilizer	Gyro	IS MP65	20.0~20.3	x,y,z/x,y,z	1.2	Implicit control
WenPod SP1 stabilizer	Gyro	IS MPU-6050	26.0~26.9	z/y,z	1.8	Implicit control
Gyenno steady spoon	Gyro	Unknown	Not found	Unknown	N/A	Not affected
Liftware level handle	Acce	IS MPU-6050	5.1	x/x	0.1	DoS

Device	Sensor		Resonant	Affected/	Max	Control
	Type	Model†	Freq. (kHz)	Func. Axes	Dist. (m)	Level
IOGear 3D mouse	Gyro	IS M681	26.6~27.6	x,z/x,z	2.5	Implicit control
Ybee 3D mouse	Gyro	Unknown	27.1~27.3	x/x,z	1.1	Implicit control
ES120 screwdriver	Gyro	ST L3G4200D	19.8~20.0	y/y	2.6	Implicit control
B&D screwdriver	Gyro	IS ISZ650	30.3~30.6	z/z	0	Limited control
Dewalt screwdriver	Gyro	Unknown	Not found	none/y	N/A	Not affected
Oculus Rift	Gyro	BS BMI055	24.3~25.6	x/x,y,z	2.4	Implicit control
Oculus Touch	Gyro	IS MP651	27.1~27.4	x/x,y,z	1.6	Implicit control
Microsoft Hololens	Gyro	Unknown	27.0~27.4	x/x,y,z	0	Limited control
iPhone 5	Gyro	ST L3G4200D	19.9~20.1	x,y,z/x,y,z	5.8	Implicit control
iPhone 5S	Gyro	ST B329	19.4~19.6	x,y,z/x,y,z	5.6	Implicit control
iPhone 6S	Gyro	IS MP67B	27.2~27.6	x,y,z/x,y,z	0.8	Implicit control
iPhone 7	Gyro	IS 773C	27.1~27.6	x,y,z/x,y,z	2.0	Implicit control
Huawei Honor V8	Gyro	ST LSM6DS3	20.2~20.4	x,y,z/x,y,z	7.7	Implicit control
Google Pixel	Gyro	BS BMI160	23.1~23.3	x,y,z/x,y,z	0.4	Implicit control
Pro32 soldering iron	Acce	NX MMA8652FC	6.2~6.5	Unknown	1.1	DoS

With these experiments, the author find that 17 of the 25 inertial sensors can be implicitly controlled by the resonant attack.

1.4 Strength

1. Consider the effect of the sampling process.
2. Propose methods to control generated signal to be in one direction.

3. Use the feedback of the target device to choose parameters.

1.5 Weakness

1. The resulting digital signal is always oscillating signal and the methods can't generate arbitrary waveform.

2 (2019 Wireless Communication)Localization and navigation in autonomous driving: Threats and countermeasures

2.1 Solved Problem

The paper contains two major parts.

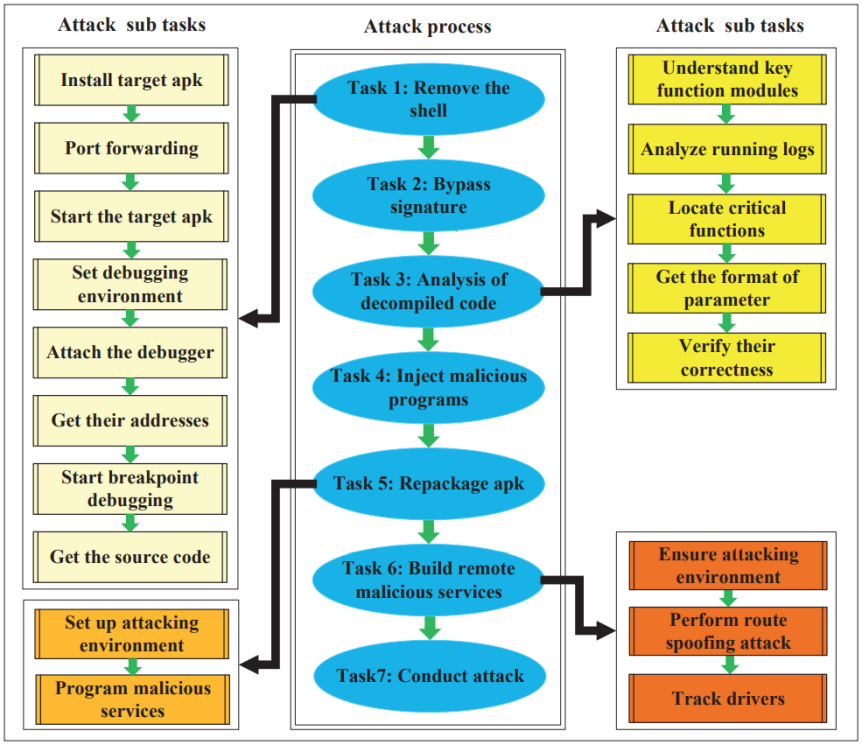
- In the first part the authors make a comprehensive survey toward security vulnerabilities in today's state-of-art navigation and localization technologies.
- In the second part, the authors devise a novel route spoofing attack by injecting malicious code to modify the parameter of the route planning function in navigation apps.

2.2 Main Idea

The authors first construct a taxonomy about differentattacks on localization and navigation system. The maincomponents in these two systems are GPS and HD map. So,the authors classify different attacks base on their targetcomponent. For each sort of attack, the authors also discuss the countermeasures toward it.

Attack surfaces	Reference	Attack point	Attack schemes	Threats	Countermeasures
GPS localization	Gorkem *et al.*, [5] Psiaki et al., [7] Huang *et al.*, [8] Parkinson *et al.*, [6] Kexiong *et al.*, [9]	GPS signal	GPS spoofing attack, GPS jamming attack, GPS replay attack	Hijack valuable devices, goods, or even target persons; manipulate the car's localization and navigation; cause GPS to not worknormally.	Optimize GPS signal system; encrypt GPS signals via private key; improve GPS terminal's resiliency to interference
	Nighswander *et al.*, [11]	GPS receiver software	GPS software attack, GPS jamming attack, GPS spoofing attack	Cause GPS to not work normally; spoof and damage GPS receivers toward GPS-dependent systems	Redundant setup of key facilities; encrypt telemetry and communication links; GPS software security
HD maps	Jeske [2] Sinai *et al.*, [3]	Traffic information	Sybil attack	Fabricate fake traffic information to update maps; influence self-driving's route planning; track users; detect users' habits and favorite hotspots	Ensure the accuracy of data collection
	Karnouskos *et al.*, [4]	Map update	Message falsification attack	Forged user identities, car locations, or road accidents and update them to HD maps; threatens users' privacy;	Use safe update framework; apply safe cloud platforms

After investigating current attack methods the authors conduct their own route spoofing attack and prove them to be effective to the most popular navigation applications. The basic idea in the attack is trying to change the parameters of the route planning functions(i.e. the staring, destination and waypoint positions). To achieve this goal, the author use the workflow below:



A rough description is to first disassemble the code of the navigation and inject malicious code to communicate to a malicious server after finding out the logic of the code. The authors conduct experiments on three popular navigation apps and find all of them affected significantly by the attack above. The author also prove that all of the apps that has a route planning functions may be vulnerable to the attack too.

2.3 Highlights Worth Learning

1. Use reverse engineering to explore the scheme of source code.
2. The taxonomy about current attacks on localization modules in autonomous vehicles.

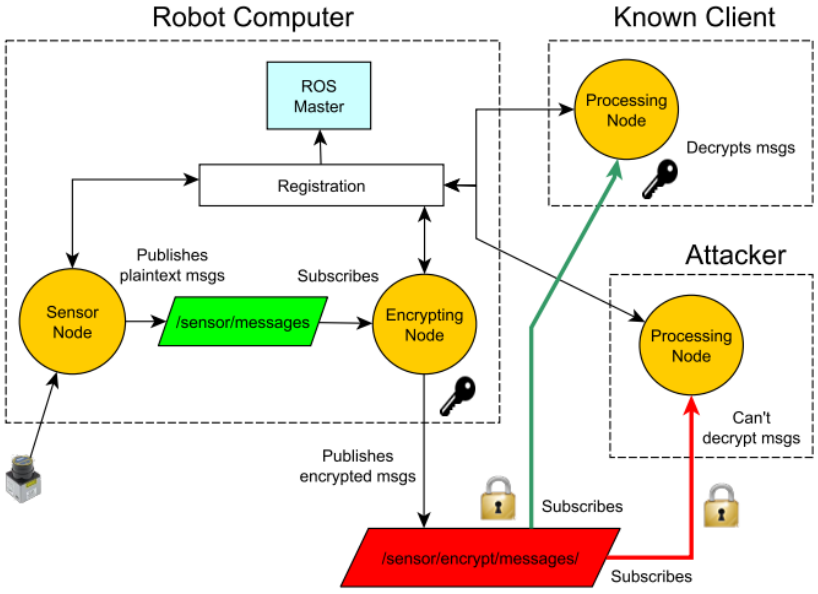
3 Cybersecurity in Autonomous Systems: Evaluating the performance of hardening ROS

3.1 Solved Problem

The ROS is designed to be composed with several nodes and a master that communicate by topic with each other. The communication model of ROS is a simple publisher and subscriber model. The simple model make the design and implementation of functionality easy, but it also makes the ROS system vulnerable to a variety of attacks because of the lack of authentication and encryption. In this paper, the author propose a altered ROS model that use cryto algorithms to protect the communication between different nodes.

3.2 Main Idea

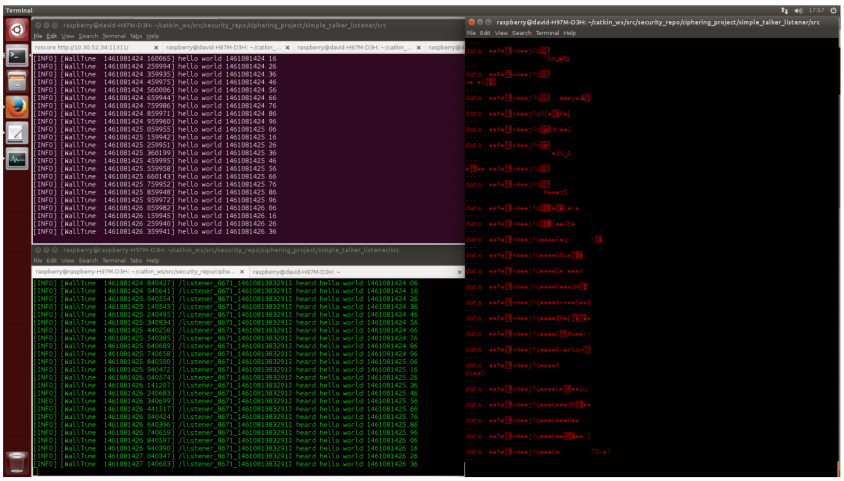
The encryption framework proposed in this paper can be illustrated below.



Instead of using the original node and topic that manage and publish sensor readings, the author create an intermediate node that subscribes the original topic, encrypt it and then publish data as another topic. After the encryption process, nodes without a valid key can't access the information conveyed.

In the feasibility study, the authors use the hello world demo provided by the ROS official. The author compare the performance in the same computer

with and without encryption. The results show that the even a malicious node can receive the message from the topic, it can't understand it without a key. The performance lost of this encryption is relatively low.



Then, the authors use real string message to test the performance and find that the performance of this system decrease geometrically when the size of the messages increase.

3.3 Highlights Worth Learning

1. The elaborate illustration of ROS's workflow.
2. Clear experimental steps.