Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing

Junjie Shen
UC Irvine
junjies1@uci.edu

Jun Yeon Won
UC Irvine
junyeonw@uci.edu

Zeyuan Chen
UC Irvine
zeyuac4@uci.edu

Qi Alfred Chen
UC Irvine
alfchen@uci.edu

Abstract

For high-level Autonomous Vehicles (AV), localization is highly security and safety critical. One direct threat to it is GPS spoofing, but fortunately, AV systems today predominantly use Multi-Sensor Fusion (MSF) algorithms that are generally believed to have the potential to practically defeat GPS spoofing. However, no prior work has studied whether today's MSF algorithms are indeed sufficiently secure under GPS spoofing, especially in AV settings. In this work, we perform the first study to fill this critical gap. As the first study, we focus on a production-grade MSF with both design and implementation level representativeness, and identify two AV-specific attack goals, off-road and wrong-way attacks.

To systematically understand the security property, we first analyze the upper-bound attack effectiveness, and discover a take-over effect that can fundamentally defeat the MSF design principle. We perform a cause analysis and find that such vulnerability only appears dynamically and non-deterministically. Leveraging this insight, we design FusionRipper, a novel and general attack that opportunistically captures and exploits take-over vulnerabilities. We evaluate it on 6 real-world sensor traces, and find that FusionRipper can achieve at least 97% and 91.3% success rates in all traces for off-road and wrong-way attacks respectively. We also find that it is highly robust to practical factors such as spoofing inaccuracies. To improve the practicality, we further design an offline method that can effectively identify attack parameters with over 80% average success rates for both attack goals, with the cost of at most half a day. We also discuss promising defense directions.

1 Introduction

Today, various companies are developing high-level self-driving cars [1] such as Level-4 Autonomous Vehicles (AV) [2], and some of them are already providing services on public roads such as self-driving taxi from Google's Waymo One [3] and self-driving trucks from TuSimple [4]. To enable such high-level driving automation, the Autonomous Driving (AD) system in an AV needs to not only perform the perception of surrounding obstacles, but also *centimeter-level localization* of its own global positions on the map [5,6]. Such localization function is highly security and safety critical in the AV context, since positioning errors can directly cause an AV to drive off road or onto a wrong way. Since in high-level AD systems the perception module is only designed for obstacle detection and the localization module is in full charge of identifying road deviations [7–11], even when the perception module is functioning perfectly, it cannot prevent a variety of road hazards specific to localization errors such as driving off road to hit road curbs, falling down the highway cliff, or being hit by other vehicles that fail to yield, especially when the AV is on the wrong way. However, recent security research in AD systems concentrates on AD perception, *e.g.*, malicious stickers on traffic signs [12–15], which leaves the security of AD localization an open problem.

For outdoor localization in general, GPS is the *de facto* location source, and thus a direct threat to it is GPS spoofing, a long-existing but still unsolved security problem with practicality proven on a wide range of end systems [16–24], including low-autonomy AVs such as Tesla cars [22]. Fortunately, to achieve robust localization, real-world high-level AD systems today predominantly use Multi-Sensor Fusion (MSF) algorithms that combine GPS input with position inputs from other sensors, typically IMU (Inertial Measurement Unit) and LiDAR (Light Detection and Ranging) [7, 25–33]. Since in such design GPS input alone can not dictate the localization output, it is generally believed to have the potential to practically defeat GPS spoofing [18, 23, 34–36]. However, state-of-the-art MSF algorithms are mainly designed for improving accuracy and robustness, instead of security. This thus makes it largely unclear how secure they can be under GPS spoofing. Given its widespread use in AVs and high importance to road safety, it is thus imperative to systematically understand this as early as possible.

To fill this critical research gap, in this work we perform the first study on the security property of MSF-based localization in AV settings. As the very first study in this direction, we focus on GPS spoofing as the attack vector since it is one of the most mature attack vectors to the MSF input sources.

We focus on a production-grade MSF implementation, Baidu Apollo MSF (BA-MSF), due to its high representativeness in both design (KF-based MSF) and implementation (centimeter-level accuracy evaluated by real-world AV fleet), which will be detailed later in §2.1. We consider the attack goal as using GPS spoofing to cause large *lateral* deviations in the MSF output, *i.e.*, deviating to the left or right. This can cause the AV to drive off road or onto a wrong way, which we call *off-road attack* and *wrong-way attack* respectively.

To systematically understand the security property, we first analyze the upper-bound attack effectiveness via a dynamic blackbox analysis since BA-MSF is released in the binary form. We find that in the real-world trace, the majority (71%) of even such upper-bound attack results can only cause less than 50 cm deviation, which is far from causing either off-road or wrong-way attacks (need over 90 cm and 2.4 m respectively). This shows that MSF can indeed generally enhance the security against GPS spoofing. Interestingly, we also observe that there still exist a few upper-bound attack results that can cause over 2 meters deviations. For all of them, we find that GPS spoofing is able to cause *exponential growths* of deviations. This allows the spoofed GPS to become the dominating input source in the fusion process and eventually cause the MSF to reject other input sources, which thus *fundamentally defeats the design principle of MSF*. In this paper, we call it a *take-over effect*. We then perform a cause analysis and find that this only appears when the MSF is in relatively *unconfident* periods due to a combination of dynamic and non-deterministic real-world factors such as sensor noises and algorithm inaccuracies.

Such take-over vulnerabilities are highly attractive for attackers since they can exploit the exponential deviation growths to achieve *arbitrary* deviation goals. However, as discovered earlier, the vulnerable periods are created dynamically and non-deterministically. Thus, we design *FusionRipper*, a novel and general attack that opportunistically captures and exploits take-over vulnerabilities with 2 stages: (1) *vulnerability profiling*, which measures when vulnerable periods appear, and (2) *aggressive spoofing*, which performs exponential spoofing to exploit the take-over opportunity.

We implement FusionRipper and evaluate it on 6 real-world sensor traces from Apollo and the KAIST Complex Urban dataset. Our results show that when the attack can last 2 minutes, there *always* exists a set of attack parameters for FusionRipper to achieve *at least* 97% and 91.3% success rates in *all* traces for the off-road and wrong-way attacks respectively, with less than 35 seconds success time on average. To understand the attack practicality, we evaluate it with practical factors such as (1) spoofing inaccuracies, and (2) AD control taking effect, and find that for both cases the attack success rates are affected by less than 4%. Attack demos showing the end-to-end attack impact are available at **https://sites.google.com/view/cav-sec/fusionripper**.

In addition, we observe that the attack effectiveness is sensitive to the selection of the attack parameters. Thus, to improve the practicality, we further design an offline attack parameter profiling method that can collect effective parameters without causing obvious safety problems during such profiling to stay stealthy. Our results on real-world traces show that our method can effectively identify attack parameters with 84.2% and 80.7% success rates for off-road and wrong-way attacks respectively, with the profiling cost of at most half a day.

Considering the critical role of localization for safe and correct AV driving, the discovered attack against the state-of-the-art MSF algorithm requires immediate attention and defense discussion. To facilitate this, we also discuss both long-term and short-term defense directions.

In summary, this work makes the following contributions:

- We perform the first security study on MSF-based localization in high-level AV settings under GPS spoofing. We focus on a production-grade MSF with both design and implementation level representativeness, and identify two attack goals specific to the AV settings.
- We analyze the upper-bound attack effectiveness, and discover a take-over effect that can fundamentally defeat the MSF design principle. We further perform a cause analysis and find that such vulnerability only appears dynamically and non-deterministically.
- We design FusionRipper, a novel and general attack that opportunistically captures and exploits the take-over vulnerability we discover. We evaluate it on 6 real-world sensor traces, and find that it can achieve high effectiveness (over 97% and 91.3% success rates) for both off-road and wrong-way attacks. We also find that such high effectiveness is robust to various practical factors.
- To improve the attack practicality, we further design an offline attack parameter profiling method that can effectively identify attack parameters with 84.2% and 80.7% success rates for off-road and wrong-way attacks respectively, with the profiling cost of at most half a day. We also discuss promising defenses directions.

2 Background

2.1 AD Localization and Multi-Sensor Fusion

In real-world high-level (*e.g.*, Level 4 [2]) AD system design, localization is a critical module that needs to compute global vehicle positions on the map in the real time based on positioning sensor inputs [7–11]. As shown in Fig. 1, its output is used by various other modules in the AD system, *e.g.*, the perception module for detecting obstacles, the planning module for driving decision making, and the control module for executing these decisions. Such direct impact on various critical decision making steps in AD driving thus makes localization outputs highly security and safety critical.

To ensure safe and correct driving, AD localization needs to not only have *centimeter-level* accuracy to localize the AV at traffic lane level [5, 6, 37], but also have high robustness under various road and weather conditions [37]. Thus,
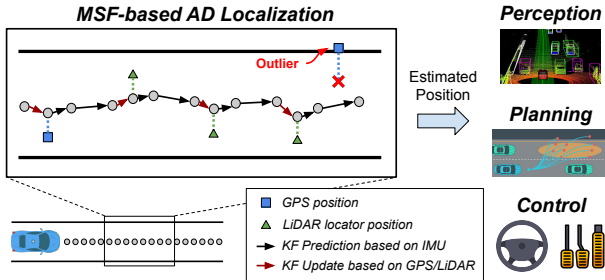
Figure 1: MSF-based localization and its use in high-level AD systems.

Multi-Sensor Fusion (MSF) based design has become the mainstream in both academia and industry since it can fuse results from multiple independent positioning sensors, typically GPS, IMU, and LiDAR, and thus produce results with overall higher accuracy and robustness [7–9, 25–33]. For example, modern AV-grade GPS receivers can achieve centimeter-level positioning accuracy with the error correction from ground stations [38]. However, GPS signal quality can be easily degraded due to natural phenomena such as atmosphere delays and multi-path effect [39]. LiDAR-based localization algorithms, or LiDAR locators [26, 40–42], match laser scans to pre-generated ones in a High Definition Map (HD Map) [43] in order to provide highly accurate positioning. However, the performance of such matching is susceptible to poor weather conditions such as rain or fog and the outdatedness of the HD Map. Thus, the goal of MSF is to leverage the strengths of these different sources while compensating their weaknesses.

Kalman Filter (KF) based MSF and its representativeness. Among MSF-based localization algorithms for AD systems, KF-based MSF is adopted most extensively in both academia and industry [25, 26, 28, 29, 31–33], and shown to have the state-of-the-art performance [25]. To concretely show its representativeness, we survey the MSF-based localization papers from top-tier robotics conferences [44] in the most recent 2 years (2018, 2019). As shown in Table 1, 14 (77.8%) of the total 18 papers adopt KF-based MSF, showing a clear predominance in today's MSF designs. Such representativeness can also be shown by the fact that it is taught in all Self-Driving Car courses from Udacity [7,8] and Coursera [9].

KF is a Bayesian filter that calculates an *optimal* state distribution with the *lowest* uncertainty from the sensor measurement distributions. In the context of AD localization, the state is composed of the vehicle's *position*, *velocity*, and *attitude* (PVA) and their *uncertainties* (or co-variance or variance matrices). Specifically, KF iteratively applies two steps: *prediction* and *update*, as illustrated in Fig. 1. In the prediction step, the acceleration and angular velocity from IMU are integrated in the KF to generate an intermediate state (black arrows in Fig. 1). In the update step, KF takes the position measurements from GPS or LiDAR locator, and updates a fraction of it to the KF state based on the uncertainties of the KF state and the measurement. A larger KF state uncertainty or a smaller measurement uncertainty will cause more updates

Table 1: Survey of MSF-based localization designs in papers published in top-tier robotics conferences (IROS, ICRA, and RSS) [44] in the most recent 2 years (2018 and 2019).

MSF Design		Papers	Percentage	
Category	Name			
KF-based	Linear KF	[25, 46–51]	7/18 (38.9%)	
	Extended KF	[52–55]	4/18 (22.2%)	14/18 (77.8%)
	Unscented KF	[56–58]	3/18 (16.7%)	
Others (*e.g.*, Particle Filter)		[59–62]	4/18 (22.2%)	

to the KF state. Please refer to the extended version [45] of this paper for more details.

Outlier detection. To prevent KF state from being easily disrupted by occasional measurements that are too noisy in the real world, the KF update is usually bounded by an *outlier detector*. Fig. 1 shows an example where a GPS measurement is discarded since its position deviates too much from the KF state. Chi-squared test is one of the most widely used outlier detectors for KF [29, 33, 63], which considers a measurement as an outlier if the Chi-squared test value is larger than a statistical significance threshold (usually 3.841 [64]). An outlier measurement can be either discarded or partially updated.

Targeted MSF implementations and representativeness. In this paper, we perform our security study on concrete MSF implementations for practicality and realism. In particular, our main target is an MSF design and implementation from the Baidu Apollo team, which we call *BA-MSF*. It is published in ICRA 2018 [25], a top-tier robotics conference [44], and follows the KF-based MSF design using high-end GPS, LiDAR, and IMU, with the Chi-squared test as the outlier detector conforming to the common practice [29, 33, 63]. As described earlier, such design is the most representative in today's MSF-based AD localization (Table 1).

Besides its design, the implementation of BA-MSF is also highly representative in today's MSF-based AD localization: it has been tested using a large AV fleet in various challenging scenarios such as urban downtown, highways, and tunnels [25], and shown the *highest* localization accuracy (0.054 m) among *all* MSF-based localization papers (including both KF-based and non KF-based) in the top-tier robotics conferences [44] of the most recent 2 years. Today, it is already adopted in Baidu Apollo [10], a production-grade AD system currently providing self-driving taxi services in China [65].

Besides BA-MSF, we also consider two other publicly-available KF-based MSFs for generality evaluations (§6.4). We follow the common parameter tuning process [66] but can only reach at most 1-2 meter accuracy, which is far from the centimeter-level accuracy required by AD systems [5, 6]. Thus, in the majority of our experiments, we target BA-MSF as it is much more representative for AD systems.

2.2 GPS Spoofing and the Practicality

GPS spoofing has been a fundamental problem for civilian GPS systems due to the lack of signal authentication in the infrastructure. In GPS spoofing, the attacker transmits fabri-

cated GPS signals with stronger power than the authentic ones, and thus causes the victim receiver to lock onto the attacker's signals and resolve positions controlled by the attacker. GPS spoofing has been proven feasible theoretically [16] and empirically [17]. So far, it has been demonstrated on various end systems such as smartphones [18, 19], drones [20, 21], yachts [23], and recently also low-level AVs such as Tesla cars [22]. Recently, a year-long investigation identified 9,883 spoofing events that affected 1,311 civilian vessel systems in Russia since 2016 [67]. Although GPS spoofers are illegal to be sold in the U.S., they can be made cheaply from commercial off-the-shelf components. For example, a low-end spoofer is as cheap as $223 [18], and higher-end ones that can simultaneously track 10+ satellites and transmit 10+ fake GPS signals only cost similar to a laptop [17, 68]. Considering such high realism, in this paper we consider it as a practical attack vector to AD localization.

3 Attack Model and Problem Formulation

3.1 Attack Goal and Incentives

Attack goals. In this paper, we target an attack scenario where an attack vehicle tailgates a victim AV while launching a GPS spoofing attack, which is both practical and effective as evaluated by previous work using real cars [18]. In such a scenario, we consider an attack goal of introducing large *lateral* deviations to the localization output of the victim AV, *i.e.*, deviating to the left or right. Since all vehicles need to drive within their designated road lanes for safety protections, such lateral deviations can pose a direct threat to road safety.

In particular, in this paper we consider two concrete attack goals specific to the AV context: *off-road attacks* and *wrong-way attack*. As illustrated in Fig. 2, the former aims at deviating to either left or right until the victim drives off the road pavement, while the latter aims at deviating to the left until the victim drives on the opposite traffic lane. Table 2 lists the required deviations to achieve these two goals, which will be used in our subsequent security analysis.

In the AV context, these two attack goals can cause various *safety hazards specific to localization errors* such as driving off road to hit road curbs or falling down the highway cliff. Since in high-level AD systems the perception module is only designed for obstacle detection and the localization module is in full charge of identifying road deviations [7–11], these hazards cannot be prevented even when the perception module is functioning perfectly. Moreover, such hazards cannot be prevented even if high-level AD systems directly use perception sensors, e.g., cameras and ultrasonic sensors, for collision avoidance. These two attack goals can also cause *vehicle collisions*, *e.g.*, with vehicles in adjacent or opposite traffic lanes. Even when the AV can perform automatic emergency brake, it cannot avoid being hit by other vehicles that fail to yield on time, especially those human driving ones with over 2 sec average driver reaction time [69].

Attack incentives. No matter whether road accidents are caused, the victim AVs under the two attack goals are already violating the traffic rules [70, 71] and exhibiting unsafe driving behaviors. These can already damage the reputation of the corresponding AV company. Thus, a likely attack incentive is *business competition*, which can allow one AV company to deliberately damage the reputation of its rival AV companies and thus unfairly gain competitive advantages. This is especially realistic today considering that there are over 40 companies competing in the AV market [1]. Meanwhile, considering the direct safety impact, we also cannot rule out the possible incentives for terrorist attacks or targeted murders, *e.g.*, against civilians, or controversial politicians or celebrities.

3.2 Threat Model

Attacker's capability. We assume that the attacker can launch GPS spoofing (§2.2) to control the positioning measurements of the victim's GPS receiver, with a similar level of measurement uncertainty as the natural GPS signals. We also assume that the attacker can track the physical positions of the victim AV in the real time during the tailgating. This can be achieved by computing the attack vehicle's own position and offsetting it with the relative position between the attack vehicle and the victim. One concrete scenario is that the attack vehicle is also an AV with a similar set of sensors and run state-of-the-art AD localization algorithms for its own position and AD perception algorithms for the relative position. Under this scenario, the attacker can thus accurately track the victim positions since for AVs precisely tracking the positions of surrounding obstacles in the real time is one of the most basic tasks for ensuring correct and safe driving. Such a scenario is especially realistic when the attack is from rival AV companies (incentive discussed in §3.1).

AV control assumption. We assume that AD systems are designed to drive on the center of traffic lanes, and constantly tries to correct any deviation to the center. State-of-the-art AD systems from both the academia [72] and industry [10, 11] follow such design and use lateral controllers to enforce it at a high frequency in the control module (*e.g.*, 100 Hz in Apollo [10]). This means that when the attacker introduces a deviation to the MSF output (*e.g.*, to the right in Fig. 2), the victim AV will actively correct it and thus cause its physical-world position to have the same amount of deviation but to the *opposite* direction (*e.g.*, to the left in Fig. 2).

3.3 Attack Formulation

Based on the attack model above, the attack in our study can be formulated as the following optimization problem:

$$\max_{\{\delta_k^a | k=1,...,n\}} \mathcal{D}(x_n^a, \{x_k | k=1,...,n\})$$
$$\text{where} \quad x_k^a = \mathcal{M}(x_{k-1}^a, r_k + \delta_k^a, z_k^{\text{lidar}}, imu_k), x_0^a = x_0, \quad (1)$$

where δ_k^a is the GPS spoofing distance to the victim's physical-world position r_k on the road plane, x_k is the MSF output without the attack, x_k^a is the MSF output with the attack, z_k^{lidar} is the LiDAR locator output, imu_k is the IMU measurement, $\mathcal{D}(\cdot)$ denotes the lateral deviation between a position and
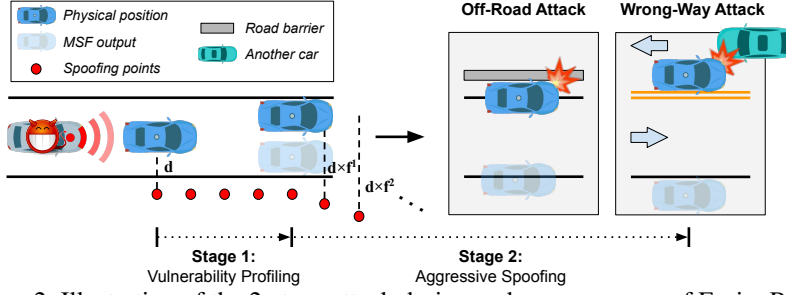
Figure 2: Illustration of the 2-stage attack design and consequences of FusionRipper.

Table 2: Required deviations for the two attack goals considered in this paper. The values are calculated based on common AV, lane, and road shoulder widths (detailed in Appendix A).

Attack Goal	Required Deviation (m)	
	Local	Highway
Off-Road Attack	0.895	1.945
Wrong-Way Attack	2.405	2.855

a trajectory, and $\mathcal{M}(\cdot)$ denotes an iteration in the KF-based MSF algorithm (introduced in §2.1), and k is the iteration index. As shown, mathematically our attack on MSF is to find a sequence of spoofing distances $\{\delta_k^a | k = 1, ..., n\}$ that can maximize the deviation of the n-th attacked MSF output to the original trajectory $\{x_k | k = 1, ..., n\}$.

4 Security Analysis of MSF Algorithm

To systematically understand the security property of MSF-based AD localization, we start with the necessary first step: understanding the upper-bound attack effectiveness, *i.e.*, the maximum possible deviation, under the attack formulation.

4.1 Upper-Bound Attack Effectiveness

Analysis methodology. To analyze the upper-bound attack effectiveness, we perform exhaustive search of possible attack inputs $\{\delta_k^a | k = 1, ..., n\}$ to the representative MSF implementation, BA-MSF, to find the one that can maximize Eq. 1. We did not choose to use an optimizer since the BA-MSF implementation is released in the binary form and thus we cannot directly get its analytical formula. For a given sensor input trace in our analysis, there are multiple possible *attack windows*, *i.e.*, from one GPS input to another later. For each attack window, we iteratively search for the δ_k^a that can deviate the most from x_k, which is a method also used in previous theoretical work on the security of single-source KF [73–76]. In accordance with our threat model, we set the measurement uncertainty of GPS spoofing inputs as the median value in real-world sensor input traces of BA-MSF.

We perform the analysis above on two types of sensor input traces: (1) real-world trace, and (2) synthetic noise-free trace. The former is obtained by directly recording the run-time MSF input while the AV is driving in the real world. Analysis results from this type of traces have the highest realism, but the types of analysis we can perform are limited since we cannot easily modify the sensor data without violating the consistency among different sensor inputs, and the analysis insights can be less clean due to real-world sensor noises. Thus, we complement it with the latter, which synthesizes MSF inputs following a given driving trajectory, with all the LiDAR locator and non-spoofed GPS inputs set to the ground truth positions, their measurement uncertainty set to the medium value in the real-world trace, and the IMU measurements calculated according to the driving trajectory.

Experimental setup. We obtain the official BA-MSF implementation from the Apollo AD system code base [10]. For the real-world trace, we use the BA-MSF input trace released by Apollo, which is recorded in Sunnyvale, CA and 4-min long [77]. In this paper, we denote it as *ba-local*. For the synthetic trace, we generate one for a common driving trajectory: driving on a straight road with a constant velocity of 45 mph. In our analysis, we use an attack window of 10 attack inputs, which is 10 seconds since the GPS input is 1 Hz in Apollo. In the exhaustive search, we enumerate δ_k^a from 0 to 10 meters with step size of 0.04 meters on both left and right sides, since we find that in our experiments GPS input deviations larger than that are identified as outliers by the Chi-squared test in BA-MSF. The medium measurement uncertainty values for GPS and LiDAR locator are calculated from trace *ba-local*.

Results. Fig. 3 (a) shows the distribution of the upper-bound deviations achieved in the 10-point attack windows for each trace. As shown, in both real-world and synthetic traces, even such maximum possible attack effectiveness is very limited: majority (76.0%) of the attack windows in the real-world trace and *all* of those in the synthetic trace cannot reach even the lowest required deviations (0.895 m) in Table 2. The main reason behind such poor attack performances is as follows. First, due to outlier detection, the maximum deviation achievable by the first attack input is very small, *e.g.*, at most 0.06 meters. Next, such tiny deviation can be quickly corrected by LiDAR locator inputs since in between two GPS attack inputs there are 5 LiDAR locator inputs (5 Hz in Apollo). This makes it highly difficult for subsequent attack inputs to build upon the deviations achieved by previous attack input. Thus, production-grade KF-based MSF algorithms today can indeed generally enhance the security against GPS spoofing.

At the same time, we also observe that the results between the real-world trace and the synthetic trace have very sharp differences: in the synthetic trace, the upper-bound deviations for all attack windows are at most 0.076 meters, while those in the real-world trace is generally larger, with 90.3% of them larger than 0.076 meters. This suggests that *sensor noises in the real world can generally degrade the security of MSF*. As shown later, such real-world factors can actually enable highly effective attacks that fundamentally break MSF in practice.

Observation: take-over effect. While our results show a general lack of attack capability to achieve even the easiest
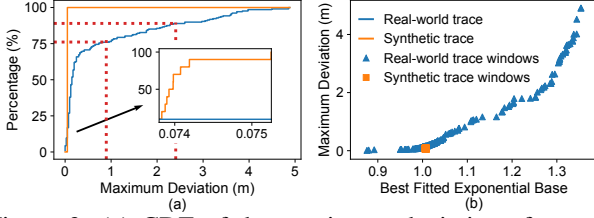
Figure 3: (a) CDF of the maximum deviations for attack windows in real-world and synthetic traces. Attack goals are marked in red dotted lines. (b) Maximum deviations and best fitted exponential bases of attack windows in the two traces.
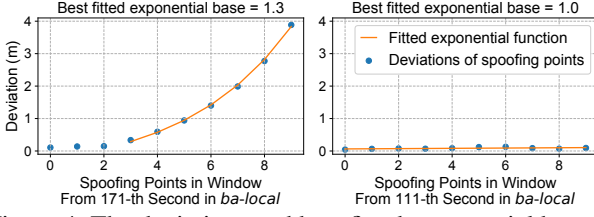

Figure 4: The deviations and best fitted exponential bases of two example attack windows in the real-world trace. Left is with take-over effect; Right is without take-over effect.

attack goal in Table 2, we also observe that for the real-world trace there still exist 14% attack windows that can actually achieve over 2 meters deviations, which are large enough for some of our attack goals. For all of these windows, we find that GPS spoofing is able to cause *an exponential growth* of deviations, and one such example is shown on the left of Fig. 4. As shown, its deviation trend is very different from those in majority of other attack windows as shown on the right of Fig. 4, which is almost flat.

To more quantitatively measure such observation, for each window, we fit an exponential function $f(x) = a^x + b$ to the deviations, where x is the x-th attack point and $f(x)$ is the deviations. For each 10-point window, we use the exponential base a in the best fitted function (based on the mean squared error) to measure the exponential growth trend. As shown in Fig. 3 (b), such exponential growth trends have strict positive correlation with the upper-bound deviations in the attack windows, and all windows that can have very large deviations, *e.g.*, over 3 meters for achieving *all* attack goals in Table 2, have very clear exponential growth trend, *e.g.*, with a being at least 1.3 (the trend on the left of Fig. 4).

Such exponential growth trend is very similar to the situation when the spoofed GPS is the only positioning source in KF updates, which is confirmed by re-running the upper-bound attack analysis in the synthetic trace without LiDAR locator inputs as shown in Fig. 5. This means that for these windows with exponential deviation growths, GPS inputs somehow become the dominating KF update source (we will analyze the cause later). In fact, according to the Chi-squared test values in the analysis logs, we find that LiDAR locator inputs actually become outliers in the latter parts of these windows and then can not provide corrections any more. This thus *fundamentally defeats the design principle of MSF, i.e.,*
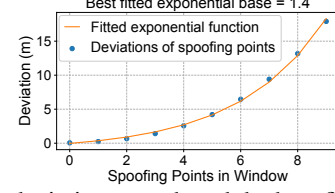

Figure 5: The deviation growth and the best fitted exponential base for BA-MSF with only the spoofed GPS input in KF updates (or a *single-source* KF-based MSF) in the synthetic trace under exhaustive search.

the fusion of multiple input sources for more robustness and accuracy. In this paper, we call it *take-over effect*.

For an attacker, such take-over effect is the most desired attack outcome, since it can efficiently cause *arbitrary deviations* and thus lead to both off-road and wrong-way attacks, and even larger ones if desired. Thus, in the next section we perform a cause analysis to understand why such take-over effect appears in the real-world trace.

4.2 Cause Analysis

Since take-over effect does not appear in all attack windows, there must be some factors other than the attack input δ_k^a that contribute to the take-over opportunity. To analyze the causes for take-over effect, we first identify possible contributing factors using theoretical analysis and experimental validation, and then use correlation analysis to identify the most important factors for the observed take-over effect in our analysis.

Contributing factor identification. To identify the set of possible contributing factors to the deviations in MSF, we first perform theoretical analysis based on the general KF-based MSF design (§2.1). From the analysis (mathematical derivations in the extended version [45]), we identify 4 theoretical contributing factors besides the attack input δ_k^a: (1) initial MSF state uncertainty $\mathbf{P}_0$, (2) LiDAR measurement uncertainty $\mathbf{R}^{\text{lidar}}$, (3) difference between LiDAR position and the original MSF output without attack Δ_{lidar}, and (4) IMU measurement *imu*. To validate that these 4 factors indeed affect the actual BA-MSF implementation, we model each of them in the synthetic trace, and experimentally measure their relationship with the deviation. Our results show that all 4 factors can positively affect the deviation. More details are in the extended version [45].

Factor importance analysis. With the 4 contributing factors identified, we then use popular causality analysis methods to understand the importance of these factors on causing the take-over effect observed in §4.1. Specifically, we perform the exponentiation function fitting as described in §4.1, and label the windows with exponential base a over 1.1 as windows with take-over effect. As shown in Fig. 3 (b), for windows without any take-over effect, *e.g.*, the ones for the synthetic trace, the exponential base a is way below 1.1. With the exponential fitting results, we identify the first point of the exponential growth to obtain $\mathbf{P}_0$. For $\mathbf{R}^{\text{lidar}}$, Δ_{lidar}, and *imu*, we use the average values from the first point of the exponential growth to the end of the window. We use 2 statistical test-

Table 3: Correlations between the contributing factors and the take-over vulnerability. Results with statistically strong correlation are highlighted in bold.

Correlation Method	Factor Importance			
	$\mathbf{P}_0$	$\mathbf{R}^{\text{lidar}}$	Δ_{lidar}	*imu*
Pearson's Correlation	**0.42 (2.0e-10)**	**0.44 (3.5e-11)**	0.12 (8.4e-2)	0.01 (8.6e-1)
Fisher's Exact Test	**21.09 (8.6e-6)**	**11.78 (5.2e-8)**	5.91 (3.2e-4)	1.95 (1.1e-1)

Pearson's correlation: *r* (*p*-value), where r is the correlation coefficient
Fisher's exact test: *or* (*p*-value), where *or* is the odds ratio

ing methods commonly used for causality analysis [78–80]: Pearson's Correlation and Fisher's Exact Test.

Analysis results. Table 3 shows the experiment results. For the two statistical testing methods, $p < 0.05$ is considered statistically significant, and $r > 0.5$ and $or > 9$ are considered strongly correlated for Pearson's Correlation and Fisher's Exact Test respectively [81]. As shown, only the *p* values for $\mathbf{P}_0$ and $\mathbf{R}^{\text{lidar}}$ are statistically significant for both methods, with their *r* values very close to showing strong correlations, and their *or* values showing strong correlations. In contrast, neither of the *r* or *or* values for Δ_{lidar} and *imu* show strong correlations, and for *imu*, the results are not even statistically significant. This suggests that the take-over effect we observe in our upper-bound analysis is most likely caused by relatively large $\mathbf{P}_0$ and $\mathbf{R}^{\text{lidar}}$ in the corresponding attack windows.

For these two most important contributing factors, $\mathbf{R}^{\text{lidar}}$ reflects the lack of confidence in the LiDAR-based localization algorithm during the attack window, and $\mathbf{P}_0$ reflects the lack of confidence in the KF states at the beginning of the attack window. This means that *take-over opportunities, or vulnerabilities, appear when the MSF is in relatively unconfident periods*. Because of this, the MSF algorithm needs to take more updates from the GPS inputs, the relatively most confident input source in that period, which thus allows GPS inputs to dominate KF updates and trigger the take-over effect.

Since $\mathbf{R}^{\text{lidar}}$ is the uncertainty reported by LiDAR locator, a large $\mathbf{R}^{\text{lidar}}$ is caused by the inaccuracies of such locator algorithm in practice. From the KF equations (detailed in the extended version [45]), a large $\mathbf{P}_0$ is mainly caused by larger uncertainties from the LiDAR locator and GPS updates before the attack window, which is thus due to algorithm inaccuracies in LiDAR locator and noises in GPS signals. Thus, unconfident periods in MSF are mainly created by practical factors such as algorithm inaccuracies and sensor noises. This also explains why we cannot observe any take-over effect in synthetic noise-free trace. These practical factors are fundamentally difficult to avoid in practice, which is exactly why MSF is designed to compensate such inaccuracies and noises from individual sources [7, 25–33]. However, as shown in our analysis, *even for the high-end sensors used in AVs today, these inaccuracies and noises are unfortunately large and frequent enough for GPS spoofing to exploit and fundamentally break MSF in practice*.

5 Attack Design: FusionRipper

Although our analysis in §4 reveals that there do exist take-over vulnerabilities for MSF in the real world, such vulnerabilities only appear in the unconfident periods created by dynamic and non-deterministic practical factors such as algorithm inaccuracies and sensor noises, which is not observable by the attacker in a tailgating attack vehicle (§3) and are highly difficult, if not impossible, to directly control. Thus, the attacker has to *opportunistically* capture and exploit such vulnerable periods in the actual attack time.

Leveraging this idea, we propose a novel attack design against MSF-based AD localization, called *FusionRipper*, which consists of 2 stages as depicted in Fig. 2:

Stage 1: Vulnerability profiling. In this stage, the attacker performs GPS spoofing and measures the feedback from the victim AV to profile when vulnerable periods appear. In our design, we aim for as fewer attack parameters as possible to maximize the ease of implementation and robustness, and thus choose to use *constant spoofing* for this stage, *i.e.*, always setting δ_k^a to a constant *d* as shown in Fig. 2. Although such profiling method is simple, our evaluation results later in §6 show that it is able to achieve a high attack success rate that is very close to the theoretical upper bound.

While performing constant spoofing, the attacker tracks victim's physical positions in real time and measures their deviations to the center of traffic lane (described in §3). If such deviation is as large as causing the AV to exhibit unsafe driving behaviors, *e.g.*, about to have unnecessary lane straddling, the victim AV is considered as in the vulnerable period. Our design uses the deviation that can touch the left or right lane line on local roads (0.295 meters, detailed in Appendix A) as the threshold to determine vulnerable periods. The intuition is that a properly designed and tested AD system should very rarely have large position deviations that can cause unsafe driving behaviors under normal fluctuations of sensor inputs. For example, the errors of BA-MSF evaluated by Baidu Apollo AVs on real roads are within 0.054 meters [25], which is far less than 0.295 meters. Thus, when such rare deviation appears, it is very likely caused by the constant spoofing, and the MSF algorithm is very likely in an unconfident period since it takes larger update from the spoofed GPS inputs.

Stage 2: Aggressive spoofing. After the vulnerable period is identified, the attacker can then perform aggressive spoofing to trigger the take-over effect and thus quickly induce large deviations. As shown in our security analysis in §4.1, the deviations grow exponentially during the take-over effect, and thus we choose exponential spoofing in the aggressive spoofing stage. As shown in Fig. 2, as soon as the attacker identifies a vulnerable period, she switches to use spoofing distance $d \times f^i$, where an exponential base *f* is cumulatively multiplied to previous spoofing distance at each of the spoofing points, and *i* is the index of the aggressive spoofing inputs.

Generality. Since FusionRipper is designed to exploit the

take-over vulnerability that is general to any KF-based MSF as discussed in our cause analysis based on the general form of KF-based MSF (§4.2), its design is generally applicable to any KF-based MSF algorithms. As shown in our generality evaluation later (§6.4), FusionRipper is highly effective on different KF-based MSF designs and implementations.

6 Attack Evaluation

6.1 Evaluation Methodology

Experimental setup. Following the common practice among AV companies [82, 83], we evaluate FusionRipper on real-world sensor traces. Specifically, we use the real-world trace *ba-local* used in our security analysis earlier (§4), and also traces from KAIST Complex Urban [84], a dataset for evaluating AD systems. Since *ba-local* is collected by the Apollo team and is designed specifically for evaluating MSF-based localization algorithms for Apollo, it is by default compatible with BA-MSF with a complete positioning sensor set as well as the HD Map for running the LiDAR locator[1].

Similar to *ba-local*, the traces in the KAIST dataset are also collected by high-end AV-grade positioning sensors [84]. But unfortunately, they do not provide the HD Map for running the LiDAR locator in BA-MSF. To address this, we assume an *ideal* LiDAR locator which always outputs the ground truth positions provided in the KAIST dataset, with their measurement uncertainty set to the median value of that in *ba-local*. Considering that one of the likely causes for the take-over effect is the LiDAR locator inaccuracies, especially the measurement uncertainty values (§4.2), this assumption only makes the attack harder and thus the results will provide the worst-case attack effectiveness on the KAIST traces.

The KAIST dataset includes 18 local traces and 2 highway traces that are compatible with BA-MSF, and we select 3 local ones and both the 2 highway ones. We truncate them to the first 5 minutes to keep the evaluation time manageable. In the selection of local traces, we select the ones with the smallest average MSF state uncertainty (*i.e.*, most confident). Considering that state uncertainty is one of the two most important contributing factors to the take-over effect (§4.1), the evaluation results on these traces provide the worst-case attack effectiveness for the KAIST traces. The detailed trace selection process can be found in the extended version [45].

Evaluation metrics. To evaluate the attack effectiveness, we apply attack parameters d and f from all possible attack starting points, *i.e.*, when the GPS input comes, in each trace, since the attacker can discover the victim at any moment in the trace and start performing the attack. As described earlier in §5, the attacker switches to aggressive spoofing when the lateral deviation between the spoofed MSF output and the non-spoofed MSF output is over 0.295 meters, which is just about to have lane straddling on local roads.

---

[1] Apollo released 8 sensor traces recorded with localization, but only *ba-local* has both the complete sensor set and compatible format with BA-MSF.

We consider the attack as successful when the lateral deviation of the MSF output is over the required deviations for the off-road and wrong-way attacks according to Table 2. This follows our AD control assumption (§3), which can directly considers the amount of deviation at the MSF output level as the amount of physical position deviations in the opposite direction to the center line. Later in §7.2, we will concretely evaluate this assumption using an end-to-end evaluation with the AD control taking effect. The *success rate* is calculated as the fraction of the successful attack starting points out of all starting points. For each attack starting point, we enumerate the combinations of d from 0.3 to 2.0 meters, with step size 0.1 meters, and f from 1.1 to 2.0, with step size 0.1. We choose these ranges because we do not find the values out of these ranges can improve the attack effectiveness in our experiments. Each d and f combination is then applied to both the left and right side of the driving direction, since both sides are valid for achieving off-road attack (detailed in §3.1). Since it takes time to (1) capture a take-over vulnerability, which is created dynamically and non-deterministically, and (2) reach the required deviations even during take-over effects (§4.1), we also consider *minimum attack duration* when calculating success rate, *i.e.*, how much time the attack can last when tailgating the victim AV. Intuitively, the longer such duration is, the higher chance she can have to hit a vulnerable period.

6.2 Attack Effectiveness

Attack success rates. Fig. 6 shows the best success rates of FusionRipper among all the combinations of d and f for the two attack goals. It shows both the results for individual traces and the average result among all traces (the thick pink line). As shown, for all traces, the average success rate is always over 75% for both attack goals even when the minimum attack duration is as low as 30 seconds. When the minimum attack duration increases, the success rates for all traces increase accordingly, which is expected since the attacker has higher chance to capture a vulnerable period. In particular, when the attack can last 2 minutes, *there exists at least one combination of d and f that can achieve over 97% success rate (98.6% on average) for the off-road attack and over 91% success rate (95.9% on average) for the wrong-way attack, for all traces in our evaluation*. Note that this is in fact the worst-case results for KAIST traces as discussed in §6.1. Since a normal taxi or truck trip is usually at least 10 minutes, it is highly likely that an attacker can find such a 2-minute tailgating opportunity in practice to launch the FusionRipper attack.

Among all the traces, *ka-local08* and *ka-highway17* shows the lowest success rate in general, especially when the required deviation is large. As shown in the extended version [45], both traces have smallest average MSF state uncertainty in their categories (*i.e.*, local and highway). This means that their MSF outputs have the highest confidence and thus are the most difficult to attack as we expect in §6.1. This also confirms that we are evaluating the worst-case attack effectiveness on KAIST traces.

Table 4: Real-world sensor traces used in our evaluation.

Source	Trace Label	Road Type	Duration	HD Map
Apollo	*ba-local*	Local	257s	Yes
KAIST Complex Urban	*ka-local08*	Local	289s	No
	ka-local31	Local	1014s	
	ka-local07	Local	553s	
	ka-highway17	Highway	1186s	
	ka-highway06	Highway	1937s	

Figure 6: Average attack success rates of (a) *off-road* attack and (b) *wrong-way* attack under different minimum attack duration.

Table 5: Ablation study results on *ba-local* trace.

Attack Config.	Off-Road		Wrong-Way	
	Succ. Rate	Succ. Time	Succ. Rate	Succ. Time
FusionRipper	98.0%	29s	97.0%	33s
Vulnerability Profiling Stage Only	14.1%	26s	7.0%	29s
Aggressive Spoofing Stage Only	10.1%	8s	5.0%	13s

Figure 7: Average success rate under different required attack deviations when the minimum attack duration is 2 minutes.

Figure 8: Average success time for reaching required deviations in off-road and wrong-way attacks under different minimum attack duration.

Between the two attack goals, the success rates only slightly drop for wrong-way attack since it has a larger required deviation. This means that the majority of the captured vulnerable periods have a successful take-over effect that can be exploited to cause different required deviations. To confirm this, we further evaluate the success rates of FusionRipper for even larger required deviations, and find that when the minimum attack duration is 2 minutes, FusionRipper is able to maintain an average success rate over 91.3% even when the required deviation is 10 meters as shown in Fig. 7.

Sensitivity to attack parameters. Table 6 lists the top 3 combinations for each trace. As shown, the attack effectiveness of FusionRipper is sensitive to the combinations of d and f. For example, the best d and f combinations are *all different* for the 6 traces. This motivates us to design an offline method to identify effective d and f combinations to increase the attack practicality, which is detailed later in §8.

Ablation study. The high attack effectiveness is a result of the combination of the two attack stages. To concretely understand this, we conduct an ablation study on *ba-local*, where we remove one of the two stages in the experiments. For *Vulnerability Profiling Stage Only*, we apply the constant spoofing distance d from each starting point. For *Aggressive Spoofing Stage Only*, we directly scale the spoofing distance using different combinations of d and f from each starting point. For both configurations, we obtain the *highest* success rates by enumerating d or f in the range specified in §6.1.

Table 5 shows the experiment results for *ba-local* when the minimum attack duration is 2 minutes. As shown, both configurations can only achieve at most 14% and 7% for the two attack goals, which is far less than 98% and 97% by Fusion-Ripper. This means that there are still some very unconfident periods that even stage 1 or stage 2 alone can succeed, but as
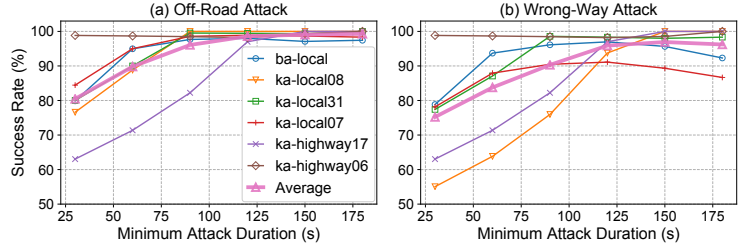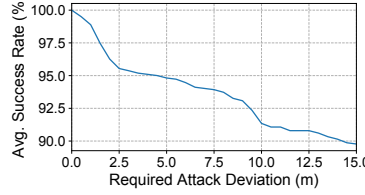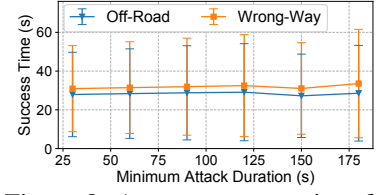
shown, without the help of each other, the success rate is very limited. This concretely demonstrates the necessity of the current 2-stage design of FusionRipper. Note that FusionRipper has longer attack success time than *Aggressive Spoofing Stage Only* due to the time spent on the vulnerability profiling stage. However, since the current ∼30 seconds attack time on average is already quite affordable for a tailgating attacker in practice, such advantage is much less important than the much higher success rates by FusionRipper.

Attack success time. For the attack success time, overall the average success time and the standard deviations are very similar under different minimum attack duration as shown in Fig. 8. When the minimum attack duration is 2 minutes, the average success time is less than 30 seconds with a standard deviation of around 25 seconds for both off-road and wrong-way attacks. This shows that FusionRipper can generally succeed very fast, *e.g.*, within a minute, even when the attacker plans to attack for over 2 minutes.

6.3 Comparison with Naive Attack Method

In this section, we compare FusionRipper with a more naive attack method: *random attack*, which randomly spoofs a deviation within a distance range for each GPS spoofing point.

Experimental setup. We perform experiments by applying FusionRipper and random attack on *ba-local*. In the random attack, we uniformly sample the position deviation between 0 to 10 meters for each spoofing point. The experiments are repeated for 30 trials. In each trial, the spoofing is performed for each attack starting point and on both the left and right. The higher success rate between that of the left and that of the right is taken as the final success rate for each trial.

Results. The first row in Table 7 shows the experiment results when the minimum attack duration is 2 minutes. We find that the random attack can barely reach any large deviation,

Table 6: Top 3 attack parameters with the highest attack success rates when minimum attack duration is 2 min.

Attack	Rank	ba-local			ka-local08			ka-local31			ka-local07			ka-highway17			ka-highway06		
		d	f	Succ. Rate	d	f	Succ. Rate	d	f	Succ. Rate	d	f	Succ. Rate	d	f	Succ. Rate	d	f	Succ. Rate
Off-Road	Top 1	0.6	1.5	98.0%	0.7	1.1	100%	0.5	1.2	99.4%	0.3	1.1	98.9%	0.3	1.2	97.0%	1.1	1.5	98.2%
	Top 2	0.6	1.6	98.0%	0.7	1.2	100%	1.0	1.3	99.4%	0.3	1.2	98.3%	0.3	1.3	97.0%	1.1	1.3	98.2%
	Top 3	0.6	1.7	98.0%	0.7	1.3	100%	1.0	1.4	99.4%	0.4	1.2	98.3%	0.3	1.4	94.0%	1.3	1.3	98.2%
Wrong-Way	Top 1	0.6	1.5	97.0%	0.3	1.2	93.8%	1.0	1.3	98.3%	0.3	1.4	91.1%	0.3	1.2	97.0%	1.2	1.3	98.2%
	Top 2	0.6	1.3	95.0%	0.3	1.3	93.8%	1.0	1.2	97.8%	0.3	1.5	90.6%	0.3	1.3	97.0%	1.3	1.3	98.2%
	Top 3	0.6	1.4	95.0%	0.5	1.3	92.1%	1.1	1.2	97.8%	0.3	1.3	88.3%	0.3	1.4	94.0%	1.1	1.3	97.6%

and as shown, its success rates are as low as 3.7% and 0.2% on average for the two attack goals respectively, which are much lower than those from FusionRipper (98.0% and 97%).

6.4 Generality of FusionRipper

In this section, we aim at understanding the generality of FusionRipper by evaluating it on more KF-based MSF implementations. Ideally we hope to find other production-grade implementations for AD systems similar to BA-MSF, but to best of our knowledge, BA-MSF is the only publicly-available one so far. Nevertheless, we still try our best to implement/port and evaluate on two other popular KF-based MSF designs, denoted as *JS-MSF* and *ETH-MSF*, which are both designed for general robotics localization instead of for AVs.

Experimental setup. BA-MSF adopts a Linear KF, the most popular KF design for MSF-based localization (Table 1). Thus, we follow a popular Linear KF based MSF design published by Joan Solà [85] and implement JS-MSF. ETH-MSF [86] is an open-source project developed by researchers from ETH Zürich for drones [87], which implements an Extended KF based MSF, the second popular KF design for MSF-based localization (Table 1). It has received over 500 stars on GitHub, which is the *highest* among the repositories under the search keyword "kalman filter sensor fusion". Both implementations use a Chi-squared test based outlier detector and directly reject outlier measurements. We follow a common parameter tuning process [66] and reach at most 1.91 and 1.17 meters localization accuracies on *ba-local* for JS-MSF and ETH-MSF respectively. Although such accuracies are far from the centimeter-level accuracy required by AD systems, they are common for general robotics localization [47,48,56].

Results. Table 7 shows the attack success rates of FusionRipper and random attack on *ba-local* for all 3 KF-based MSF implementations. As shown, FusionRipper can generally achieve high success rates on all three MSFs, which are 100% on both JS-MSF and ETH-MSF for both attack goals. However, we also notice that even random attack can also achieve over 95% success rates for the off-road attack, and over 70% for the wrong-way attack. This suggests that JS-MSF and ETH-MSF are both very unstable, which can also be seen by the fact that their natural localization errors are already 1.17 and 1.91 meters. In contrast, BA-MSF can achieve 0.054 meters accuracy, which is likely due to additional design features such as zero-velocity update [25], and

Table 7: Attack success rates of FusionRipper and random attack on 3 MSF implementations. The attacks are evaluated on *ba-local* with 2-minute minimum attack duration.

Attacked MSF	FusionRipper		Random Attack (avg. of 30 trials)	
	Off-Road	Wrong-Way	Off-Road	Wrong-Way
BA-MSF	98.0%	97.0%	3.7%	0.2%
JS-MSF	100%	100%	97.4%	92.4%
ETH-MSF	100%	100%†	95.9%	72.5%

†Achieves 100% success rate when using a smaller f (1.02).

better parameter tuning by professional AV engineers. Thus, while our results show that FusionRipper is general for all 3 KF-based MSF implementations, we believe that the results on BA-MSF can more representatively indicate the security status of production-grade MSF-based AD localization today.

7 Practical Attack Considerations

Although FusionRipper already shows very high effectiveness in §6, we haven't considered two factors that may affect the attack effectiveness in practice: (1) the variations in the spoofed positions and their measurement uncertainty at the victim's GPS receiver, and (2) sensor input changes due to AD control during the attack. In this section, we evaluate the robustness of FusionRipper under these two practical factors. The experiments in this section are mainly performed on the *ba-local* trace since it has the complete set of real-world sensor inputs for BA-MSF and thus has the highest realism.

7.1 Robustness Against Spoofing Inaccuracies

In §6, we directly set spoofed GPS inputs $r_k + \delta_k^a$ based on d and f, and set their uncertainty R_k as the medium value in real-world traces. However, in practice both can have variations due to sensor noises. In this section, we denote the variances to $r_k + \delta_k^a$ as σ_{pos}, and those to R_k as σ_{var}.

Inaccuracy sources and modeling. As specified in our threat model (§3), we assume that the attacker can estimate the victim AV's real-time positions based on her own position and the distance to the victim. Thus, there are three possible error sources for σ_{pos}: 1) localization error σ_1 in attacker's self-localization process, 2) distance measurement error σ_2 in the measured distance between the attack vehicle and the victim AV, and 3) GPS receiver error σ_3, *i.e.*, the difference between the position the attacker intended to set and the actual received position at the victim side. Assuming the attacker is equipped with the same sensor set used in an AD system
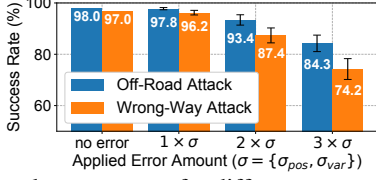
Figure 9: Attack success rate for different amounts of spoofing errors. Experiment of each error amount is repeated 100 times.

and can run an MSF algorithm of similar quality, σ_1 will be similar to the inaccuracies of BA-MSF algorithm, which is reported as 0.054 meters in [25]. Since LiDAR can be used to measure the distance to the victim, σ_2 is thus the distance measurement error in the LiDAR sensor, which is 0.02 meters as specified in the datasheet according to the LiDAR model used in Apollo [88]. For σ_3, we directly use the positioning error, 0.01 meters, as specified in the datasheet of the GPS model used in Apollo [38]. Assuming that these errors are normally distributed with a zero-mean (common practice in robotics [89]), the combined distribution for σ_{pos} is conforming to $N(0, \sigma_1^2 + \sigma_2^2 + \sigma_3^2) = N(0, 0.058^2)$. For the measurement uncertainty error σ_{var} during spoofing, we measure the distribution of GPS measurement uncertainty in the *ba-local* trace, and take the standard deviation $\sigma_{var} = 0.008$.

Experimental setup. We apply these error distributions to the FusionRipper attack in *ba-local* using the best attack parameter in *ba-local* with 2-minute minimum attack duration. For each GPS spoofing input, we randomly sample a position error from $N(0, \sigma_{pos}^2)$ and the error direction from a uniform distribution between 0 to 360 degrees, and apply them to the spoofed input. Similarly, we randomly sample an error value from $N(0, \sigma_{var}^2)$ and apply it to the measurement uncertainty of each spoofing input. To further explore the impact of these errors, we also apply 2× and 3× amounts of the normal error (σ_{pos} and σ_{var}), in our evaluation. We repeat the experiment 100 times for each error amount.

Results. Fig. 9 shows the attack success rates under each error amount. As shown, under normal error amount (1 × $\{\sigma_{pos}, \sigma_{var}\}$), the success rate is only reduced by 0.2% for the off-road attack, and by 0.8% for the wrong-way attack. Even when the error amount is 3× than normal, meaning that the error can be as large as 0.174 meters, the success rate is still 84.3% and 74.2% on average for off-road and wrong-way attacks respectively. This shows that FusionRipper is highly robust to spoofing inaccuracies in practice.

7.2 End-to-End Attack Impact Evaluation

In §6, we assume the amount of deviation in MSF outputs is the same as the amount of physical position deviations to the center line. In this section, we concretely evaluate this assumption by performing an end-to-end attack impact evaluation with the AD control taking effect.

Evaluation methodology. In this evaluation, we adopt two evaluation methods popularly used in AV industry [82, 90]: *trace based* and *simulation based*. In the trace-based evaluation, we still use the original real-world sensor trace *ba-local*,

and synthesize the sensor input changes corresponding to the output of the control module in Apollo. Specifically, the lateral controller in Apollo runs a linear-quadratic regulator algorithm [91] on the lateral deviation in the MSF output, which calculates the amount of steering that will be applied to correct the deviation. We thus mathematically translate such steering into physical position and heading rate changes (detailed in Appendix B), and add them to the original LiDAR locator position and IMU values to get the changed ones due to AD control. The benefit of this method is that it contains real-world sensor noises, which is the key contributor to the take-over vulnerability (§4). However, it does not model more complicated sensing and vehicle motion factors such as raw LiDAR point cloud changes and tire-road frictions, which thus may have limited synthesizing accuracy.

In the simulation-based evaluation, we directly use an AD simulator to dynamically generate raw sensor inputs to Apollo according to its control decisions in the real time, which has more advanced sensor and vehicle motion modelling. However, a common limitation for AD simulators today [92, 93] is that they do not consider generating sensor data with real-world noises. To address this, we model the LiDAR noises as position errors following a normal distribution with a zero mean for each point of the raw LiDAR point cloud generated from the simulator according to the LiDAR datasheet [88].

Experimental setup. In the trace-based evaluation, we run Apollo version 2.5 (the latest version directly compatible with *ba-local*) with the control module enabled on a GPU server, and feed trace *ba-local*. We write a standalone ROS node that feeds the spoofed GPS inputs and also performs the LiDAR locator and IMU input changes described above. For FusionRipper, we use the best attack parameter in *ba-local* with 2-minute minimum attack duration. We do not run the perception module since in Apollo the perception module only outputs detected road obstacles and the system solely relies on the localization module to identify deviations on the road. This is the most popular design modularization for high-level AD systems today [7–11], which lets the localization module to take charge of all aspects related to vehicle positioning.

In the simulation-based evaluation, we use LGSVL, a production-grade AD simulator that can interface with Apollo version 5.0 [93]. Since Apollo version 5.0 replaces the ROS runtime with Cyber [10], we implement the attack logic and noise modeling in a Cyber node instead. Different from the trace-based evaluation, we run the simulation on the *complete* Baidu Apollo AD system with all functional modules enabled, *i.e.*, localization, transform, perception, prediction, planning, routing, and control [10]. We simulate two attack scenarios with one attacking to the left of the road and another to the right, where both have concrete safety consequences such as hitting the road barrier or traffic sign.

Trace-based evaluation results. Our results show that FusionRipper achieves 97.0% and 93.9% success rates for off-road and wrong-way attacks respectively, which is only
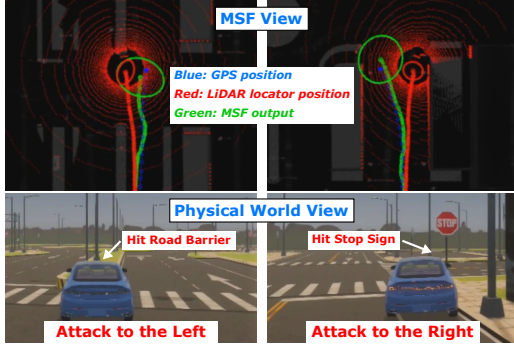
Figure 10: Snapshots of our end-to-end attack demos [94]. MSF View: input sensor positions and MSF outputs; Physical World View: victim AV's physical world position.

slightly lower than those in the MSF algorithm-only analysis (98.0% and 97.0%). Such slightly effectiveness drop may be due to run-time randomness when running the end-to-end Apollo system since it uses multi-threading when feeding the sensor inputs to BA-MSF.

Simulation-based evaluation results and attack demos. Our simulation results show that FusionRipper can successfully deviate the victim AV to hit the road barrier or traffic sign even with the complete end-to-end Baidu Apollo AD system operating. We record attack demo videos for these two simulation scenarios, available at our project website **https:// sites.google.com/view/cav-sec/fusionripper**. Fig. 10 shows a snapshot of the demos. As shown, to correct the MSF output deviation to the right/left of the planned trajectory (*i.e.*, lane center), the AV in the physical world deviates to the left/right and eventually hit the road barrier or the stop sign.

8 Offline Attack Parameter Profiling

Our results so far show that for each trace there always exist an attack parameter combination, *i.e.*, d and f, that can achieve high success rates (§6) with high robustness to practical factors (§7). However, in §6.2 we also observe that such high effectiveness is sensitive to the selection of attack parameters. Thus, it is highly desired if there exists an offline method that can efficiently identify highly effective attack parameters before the actual attack. In this section, we thus explore the possibility of designing such a method to further improve the practicality of FusionRipper.

8.1 Problem Settings and Design

Problem Settings. To find the effective attack parameters offline, we assume that the attacker can perform trials of FusionRipper attacks with different combinations of d and f on AVs of the *same model* as that of the victim AV, *i.e.*, having the same sensor set, AD system, and vehicle model. This is realistic since any AV models developed for commercial purpose need to be mass produced for the ease of management and reducing the development cost for the self-driving taxi or truck services today [65, 95–97]. For example, Waymo's 20,000 self-driving taxis in Phoenix are deployed with the same sensor suite on the same car model [98]. In this process,

the attack trials can be performed *actively*, by requesting self-driving taxi or truck services that use the targeted AV model, or directly purchasing an AV of the same model.

In such profiling process, it is necessary to prevent causing obvious safety problems both for the attacker's own safety and for remaining stealthy. Thus, in such offline profiling we choose a *safe profiling design*, which still performs the FusionRipper attack but stops the attack right after the physical-world deviation of the AV is over a *safe profiling threshold*. This will thus let the non-spoofed GPS and other positioning sources to drag the MSF output deviations back.

Offline profiling algorithm design. Under the problem settings above, our profiling method is designed following a simple strategy: performing attack trials using different combinations of d and f until we find a combination with a sufficiently high success rate. More specifically, the trials are performed for a number of *profiling rounds*. In each round, the attacker picks one combination of d and f and tries it for multiple times. When picking the combinations, the attacker follows the order from the smallest one to the largest one in the parameter space, since larger ones can more easily make the spoofed inputs outliers and thus directly cause attack failure.

Due to the safety requirement, the attacker follows the safe profiling design above, and considers a d and f combination as successful once it reaches the safe profiling threshold. After each profiling round, the attacker can thus obtain a success rate for a d and f combination. Once the success rate of a combination in a round is over a *minimum profiling success rate*, the profiling terminates and such combination is selected for the actual attack. If the attack parameters space is exhausted, the combination with the highest success rate in profiling is selected. The pseudocode of this method is in Algorithm 1.

8.2 Experiments and Evaluation

Experimental setup. In this section, we use the 5 KAIST traces used in §6.2 since this represents the case with attacking the same AV model (the KAIST traces are collected using the same vehicle on different roads [84]). We split the 5 traces into two sets, with 4 as the *profiling traces*, *i.e.*, representing the attack trials in the offline profiling, and 1 as the *evaluation trace* for evaluating the selected d and f from profiling, *i.e.*, representing the actual attack on the victim AV. We evaluate all the 5 possible splittings, and then use their average success rate to measure the offline profiling effectiveness. We use the same parameter space as that in §6.

Algorithm parameter choices. In the profiling algorithm, there are two configurable parameters: *minimum profiling success rate*, and *safe profiling threshold*. Thus, we first perform experiments to understand how to best configure them. In these experiments, for each d and f combination we consider all attack starting points in the profiling traces as its corresponding set of attack trials in the profiling algorithm in order to understand general properties of different parameter values.

We first perform experiments by running the profiling algorithm for different minimum profiling success rates without

Algorithm 1 Offline Attack Parameter Profiling

Notations:
ATTACKTRIALS(d, f, n, t): Profile n attack trials with parameters d, f, returns the number of trials that have deviations larger than t
N: Number of attack trials in each profiling round
S: Minimum profiling success rate
T: Safe profiling threshold
Output: d, f, cost
Initialize $d, d_{best} \leftarrow d_{min}$; $f, f_{best} \leftarrow f_{min}$; SuccRate$_{best}$, cost $\leftarrow 0$

```
 1:  for each f ← f_min to f_max do
 2:      for each d ← d_min to d_max do
 3:          SuccCount ← ATTACKTRIALS(d, f, N, T)
 4:          cost ← cost + N
 5:          SuccRate ← SuccCount/N
 6:          if SuccRate ≥ S then
 7:              return d, f, cost
 8:          else
 9:              if SuccRate > SuccRate_best then
10:                  d_best ← d, f_best ← f
11:                  SuccRate_best ← SuccRate
12:              end if
13:          end if
14:      end for
15:  end for
16:  return d_best, f_best, cost
```

considering safe profiling design. Our results show that the average success rate of the selected d and f does not change significantly overall. Particularly, it peaks when the minimum profiling success rate is 50% for both attack goals and drops after that, maybe due to the overfitting to the profiling traces. More details are in Fig. 15 (a) in the Appendix.

Next, with 50% as the minimum profiling success rate, we vary the safe profiling threshold, and find that reducing the safe profiling thresholds only slightly changes the average success rate of the selected d and f: the success rate differences between profiling threshold 0.3 and 0.9 meters are less than 4% for both attack goals. In particular, using 0.45 meters as the safe profiling threshold has the overall highest average success rate for both attack goals, which are 90.3% and 84.4% respectively. Details are in Fig. 15 (b) in the Appendix. Such 0.45 meters deviation does not cause the AV to drive off road on both local roads and highway (Table 2). On local roads, it will only cause very slightly lane straddling, and on the highway, it is far from even touching the left or right lane line (both visualized in Fig. 13 in Appendix). Thus, the attacker can choose to perform such safe profiling on the highway, or on the local roads with light traffic.

Evaluation results. With the algorithm parameter values decided, we then evaluate the algorithm effectiveness and the profiling cost with limited number of attack trials for each combination of d and f in the profiling round. We define profiling cost as the total number of attack trials spent in the profiling algorithm, since in our problem setting each trial corresponds to a self-driving trip the attacker needs to take, *e.g.*, from a targeted self-driving taxi service. For each attack trial, we limit its maximum duration to 90 seconds, which generally covers over 95% of the successful cases according
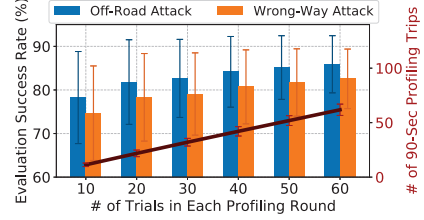

Figure 11: Average profiling effectiveness (bar graph) and costs (line graph) under different numbers of attack trials in each profiling round. Each profiling is repeated for 100 times.

to our earlier evaluation on attack success time (§6.2).

Fig. 11 shows the average success rates of the d and f output by the profiling algorithm and the average numbers of 90-sec profiling trips under different numbers of attack trials in each profiling round. In each profiling round, we randomly sample the corresponding number of attack trials from all attack starting points in the profiling traces. As shown, the average success rate increases as the attacker spends more trials in each profiling round since with more trials, the profiled success rate of a d and f combination in a profiling round is statistically closer to the ground truth. Particularly, when the number of trials in each profiling round is 40, our profiling algorithm can find a d and f combination with over 80% average success rate for both off-road and wrong-way attacks (84.2% and 80.7% respectively). In this case, *the profiling cost is only 42 1.5-minute trips on average, which in total is only slightly over 1 hour*. Since the attackers can actively perform such trials, *e.g.*, by requesting self-driving taxi services themselves, finishing this should take at most half a day.

9 Limitation and Defense Discussions
9.1 Limitations of Our Study
Study representativeness. As the first work to study the security of MSF-based AD localization, we choose to focus on the most representative design, KF-based MSF, and the most representative implementation we can find, BA-MSF (representativeness discussed in §2.1). However, it is still unclear whether other less common MSF designs (*e.g.*, particle filter based [59]) and outlier detection designs (*e.g.*, expectation-maximization based [99]) can be more secure, which can be potential future work directions.

Attack generality. Although our results have shown the generality of FusionRipper by showing high success rates on 3 different KF-based MSFs (§6.4), only one (BA-MSF) of them is production-grade implementation for AD systems. Ideally it is better to evaluate on other production-grade ones, but very unfortunately BA-MSF is the only one that is publicly available so far and it is unlikely for other AV companies to publicly release their implementations in the near future. Thus, due to the lack of information, it is unclear whether other leading AV companies, *e.g.*, Waymo and GM, are vulnerable to our attack. Nevertheless, since BA-MSF is representative both at the design and implementation levels (§2.1) and our attack is general to KF-based MSF by design (§4.2), if other

AV companies also adopt such a representative design, at least at design level they are also susceptible to the discovered take-over vulnerability. Thus, as the first study, we believe our current discovery and evaluation results can already most generally benefit the understanding of the security property of MSF-based AD localization today.

Attack practicality. We evaluate FusionRipper on real-world traces and under various practical factors such as spoofing inaccuracies and AD control taking effect (§7). To further improve the attack practicality, we design an offline attack parameter profiling method that can achieve 84.2% and 80.7% success rates for off-road and wrong-way attacks, with the profiling cost of at most half a day. Nevertheless, due to the cost and legal regulation for GPS spoofing, we did not conduct attack experiments on real-world AVs, which thus can be a valuable future work. Note that GPS spoofing has been proven practical on various end systems [16–23], including cars such as Tesla cars [22] (§2.2). Moreover, in this work, we model GPS spoofing based on attack capabilities shown in prior work [18, 19, 23] to minimize any unrealistic assumptions.

As mentioned in §3.2, we assume the attacker owns an AV and can leverage AD perception algorithms to track the physical position of the victim. Although accurate position-tracking of surrounding obstacles is a basic task for AVs, we did not conduct physical-world experiments to confirm this, which is thus left as a valuable future work.

9.2 Defense Discussions

In this section, we discuss the potential defense directions against FusionRipper.

Defend against GPS spoofing. Our attack depends on GPS spoofing, so one direct defense direction is to leverage existing GPS spoofing detection or prevention techniques. Unfortunately, neither GPS spoofing detection nor prevention are fully-solve problems today. On the detection side, numerous techniques have been proposed leveraging signal power monitoring [100–102], multi-antenna based signal arrival angle detection [101, 103], or crowdsourcing based cross-validation [104]. However, they either can be circumvented by more advanced spoofers [21, 101] or are only applicable to limited domains such as airborne GPS receivers [104]. On the prevention side, cryptographic authentication based civilian GPS infrastructure can fundamentally prevent direct fabrications of GPS signals [101]. However, it requires significant modifications to the existing satellite infrastructure and GPS receivers, and is still vulnerable to replay attacks [105]. Thus, one interesting future work direction is to more concretely understand how effective the latest GPS spoofing defenses can be against the current or adapted versions of FusionRipper.

Improve confidence of MSF state and LiDAR locator. Another fundamental defense direction is to improve the positioning confidence of MSF state and LiDAR locator, the two most important factors to the take-over vulnerability in real-world trace (§4). Fundamentally, such lacks of confidence in practice result from algorithm inaccuracies and sensor noises

(§4), and as shown in our analysis, even for the high-end sensors and production-grade LiDAR locator used in AVs today, these inaccuracies and noises are unfortunately large and frequent enough for FusionRipper to exploit. To improve on this, substantial technology breakthrough in sensing and LiDAR-based localization needs to take place. Unfortunately, it is unclear when such breakthrough can take place.

Leverage independent positioning sources (*e.g.*, camera-based lane detection) as fail-safe features for high-level AD localization. Since fundamental defense directions above are not immediately deployable, it is highly desired to discuss the possibility of short-term mitigation solutions. One promising direction is to leverage independent positioning sources to cross-check the localization results and thus serve as *fail-safe* features for AD localization. For example, since both off-road and wrong-way attacks will cause the victim AV to deviate from the current lane, they should be detectable by *camera-based lane detection* [106], a mature technology available in many vehicle models today [107]. However, we find that in the high-level AD system design today, such a technology has not been generally considered for fail-safe purposes. For example, the latest release of Baidu Apollo (version 5.5) uses it only for camera calibration [10], while Autoware does not use it at all [11]. This might be because the lane detection output is local positioning within the current lane boundaries, and thus cannot be directly used for comparison against global positioning from MSF. However, the vulnerability discovered in this paper strongly motivates the need for considering adding such kind of fail-safe features in future AD localization, at least for *anomaly detection*. Note that more investigations are needed to understand how effective and robust such kind of fail-safe features can be in the defense. For example, when camera-based lane detection is applied for anomaly detection, the precision/recall rates need to be further explored since it needs to carefully consider (1) AVs legitimately deviating from current lane due to routing requirements, and (2) lane line scratches or incompleteness. Moreover, camera-based lane detection itself is vulnerable to physical-world attacks [108, 109].

Note that even if such fail-safe features can perform perfect attack detection, our attack still causes denial-of-service of the victim's global localization function, which can render the victim in unsafe scenarios, *e.g.*, stopping in the middle of highway lanes, since the victim can neither correctly reach the destination nor safely locate the road shoulder to pull over. Thus, a more useful defense direction is to *correct* the attacked localization results. However, so far the global positioning accuracy of cameras is unsatisfying for high-level AD localization, especially along the longitudinal direction (forward/backward) since only the stop lines can be used as features [32, 110]. This is why LiDAR locator is used more predominantly in high-level AD localization (§2.1). Moreover, such correction is yet another multi-sensor fusion problem

and thus is still fundamentally vulnerable to the take-over vulnerability discovered in this paper (§4). Thus, how to leverage other independent positioning sources to effectively perform such correction under our attack is still an open research challenge, which can be a valuable future work direction.

10 Related Work

GPS spoofing on navigation systems. Recently, Zeng *et al.* [18] find that GPS spoofing can be used to stealthily deviate a victim car to an attacker-controlled destination. Later Narain *et al.* [19] further find that such attack also exists for a GPS/INS (Inertial Navigation System) navigation system. Compared to our work on MSF-based localization, these prior works target single-source localization systems without fusion from other position sources, such as a LiDAR locator.

Theoretical work on KF security. Existing theoretical works [73–76] from the control systems domain have studied the security of KF under sensor spoofing. Compared to our work, they only study single-source KFs without any sensor fusion. Also, they focus on the theoretical aspect of the KF and assume the attacker has full access to the KF internals, *e.g.*, KF state and uncertainties. In comparison, our work does not make such assumptions and hence is much more realistic.

AV-related attacks and defenses. Various previous works studied security problems on traditional vehicle systems [111–113], but not AD systems. Closer to this work, prior works discovered various sensor attack vectors on sensors related to AD systems, such as camera, LiDAR, IMU, radar, and ultrasonic sensors [15, 114–118]. However, none of them considers how to leverage these attack vectors to attack AD localization. On the defense side, recently Choi *et al.* [119] and Quinonez *et al.* [120] propose to use control or physical invariants to detect sensor attacks to small robotics vehicles such as drones and ground rovers. However, it is unclear how these methods can be effectively applied to AD systems, since AVs operate in highly complex and dynamic road conditions where the baseline/normal behaviors can be much harder to accurately model or predict.

11 Conclusion

In this paper, we perform the first security study on MSF-based localization in high-level AV settings under GPS spoofing. We discover a take-over vulnerability that can fundamentally defeat the MSF design principle, and design FusionRipper, a novel and general attack that opportunistically captures and exploits it. Our evaluation on real-world traces shows that FusionRipper can achieve over 97% and 91.3% success rates in all traces for off-road and wrong-way attacks. Such high effectiveness is also found highly robust to various practical factors. We also design an offline method that can identify effective attack parameters within at most half a day. We also discuss both long-term and short-term defenses directions, and identify that a promising mitigation is to use camera-based lane detection as a fail-safe feature, which has not been generally considered for such purpose today. As the first study

on AD localization security, we hope that our findings and insights can bring immediate attention and inspire the development of effective defenses considering the critical role of localization for safe and correct AV driving.

References

[1] "40+ Corporations Working On Autonomous Vehicles." https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list.

[2] SAE On-Road Automated Vehicle Standards Committee and others, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," *SAE International: Warrendale, PA, USA*, 2018.

[3] "Waymo has launched its commercial self-driving service in Phoenix - and it's called 'Waymo One'." https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12.

[4] "UPS joins race for future of delivery services by investing in self-driving trucks." https://abcnews.go.com/Business/ups-joins-race-future-delivery-services-investing-driving/story?id=65014414.

[5] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *Robotics: science and systems*, vol. 4, p. 1, Citeseer, 2007.

[6] T. G. Reid, S. E. Houts, R. Cammarata, G. Mills, S. Agarwal, A. Vora, and G. Pandey, "Localization Requirements for Autonomous Vehicles," *arXiv preprint arXiv:1906.01061*, 2019.

[7] "Self-Driving Car Engineer Nanodegree." https://www.udacity.com/course/self-driving-car-engineer-nanodegree--nd013.

[8] "Self-Driving Fundamentals: Featuring Apollo." https://www.udacity.com/course/self-driving-car-fundamentals-featuring-apollo--ud0419.

[9] "State Estimation and Localization for Self-Driving Cars." https://www.coursera.org/learn/state-estimation-localization-self-driving-cars.

[10] "Baidu Apollo." https://github.com/ApolloAuto/apollo.

[11] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware On Board: Enabling Autonomous Vehicles with Embedded Systems," in *ICCPS'18*, pp. 287–296, IEEE Press, 2018.

[12] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in *CVPR*, 2018.

[13] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, T. Kohno, and D. Song, "Physical Adversarial Examples for Object Detectors," in *WOOT*, 2018.

[14] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors," in *CCS*, 2019.

[15] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving," in *CCS*, 2019.

[16] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the Requirements for Successful GPS Spoofing Attacks," in *CCS*, 2011.

[17] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," in *ION GNSS'08*, 2008.

[18] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems," in *USENIX Security*, 2018.

[19] S. Narain, A. Ranganathan, and G. Noubir, "Security of GPS/INS based On-Road Location Tracking Systems," in *IEEE Symposium on Security and Privacy (SP)*, 2019.

[20] L. Franceschi-Bicchierai, "Drone Hijacking? That's Just the Start of GPS Troubles," *Retrieved April*, vol. 27, p. 2013, 2012.

[21] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned Aircraft Capture and Control via GPS Spoofing," *Journal of Field Robotics*, 2014.

[22] "Tesla Model S and Model 3 Vulnerable to GNSS Spoofing Attacks." https://www.gpsworld.com/tesla-model-s-and-model-3-vulnerable-to-gnss-spoofing-attacks/.

[23] J. Bhatti and T. E. Humphreys, "Hostile Control of Ships via False GPS Signals: Demonstration and Detection," *NAVIGATION: Journal of the Institute of Navigation*, 2017.

[24] J. Noh, Y. Kwon, Y. Son, H. Shin, D. Kim, J. Choi, and Y. Kim, "Tractor Beam: Safe-hijacking of Consumer Drones with Adaptive GPS Spoofing," *ACM Transactions on Privacy and Security (TOPS)*, vol. 22, no. 2, pp. 1–26, 2019.

[25] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and Precise Vehicle Localization based on Multi-Sensor Fusion in Diverse City Scenes," in *ICRA*, pp. 4670–4677, IEEE, 2018.

[26] Y. Gao, S. Liu, M. Atia, and A. Noureldin, "INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm," *Sensors*, vol. 15, no. 9, 2015.

[27] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor Fusion-based Low-Cost Vehicle Localization System for Complex Urban Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1078–1086, 2016.

[28] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Mapping and Localization Using GPS, Lane Markings and Proprioceptive Sensors," in *IROS*, IEEE, 2013.

[29] M. Schreiber, H. Königshof, A.-M. Hellmund, and C. Stiller, "Vehicle Localization with Tightly Coupled GNSS and Visual Odometry," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2016.

[30] F. de Ponte Müller, "Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles," *Sensors*, vol. 17, no. 2, p. 271, 2017.

[31] A. Soloviev, "Tight Coupling of GPS, Laser Scanner, and Inertial Measurements for Navigation in Urban Environments," in *IEEE/ION Position, Location and Navigation Symposium*, IEEE, 2008.

[32] B.-H. Lee, J.-H. Song, J.-H. Im, S.-H. Im, M.-B. Heo, and G.-I. Jee, "GPS/DR Error Estimation for Autonomous Vehicle Localization," *Sensors*, vol. 15, no. 8, pp. 20779–20798, 2015.

[33] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-Calibration," *IJRR*, 2011.

[34] S. Lee, Y. Cho, and B.-C. Min, "Attack-Aware Multi-Sensor Integration Algorithm for Autonomous Vehicle Navigation Systems," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3739–3744, IEEE, 2017.

[35] D. Davidson, H. Wu, R. Jellinek, V. Singh, and T. Ristenpart, "Controlling UAVs with Sensor Input Spoofing Attacks," in *WOOT*, 2016.

[36] S. M. Albrektsen, T. H. Bryne, and T. A. Johansen, "Robust and Secure UAV Navigation Using GNSS, Phased-Array Radio System and Inertial Sensor Fusion," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1338–1345, IEEE, 2018.

[37] "Report On Road User Needs And Requirements," tech. rep., European GNSS Agency, 2019.

[38] NovAtel, "NovAtel SPAN on ProPak6 Datasheet." https://www.novatel.com.

[39] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and More*. Springer Science & Business Media, 2007.

[40] J. Levinson and S. Thrun, "Robust Vehicle Localization in Urban Environments Using Probabilistic Maps," in *2010 IEEE International Conference on Robotics and Automation*, pp. 4372–4378, IEEE, 2010.

[41] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 110–124, 2015.

[42] P. Biber and W. Straßer, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in *IROS*, IEEE, 2003.

[43] "HD Maps: New Age Maps Powering Autonomous Vehicles." https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/.

[44] E. Berger, "CSRankings." http://csrankings.org/.

[45] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, "Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing (Extended Version)," *arXiv preprint arXiv:2006.10318*, 2020.

[46] S. Piperakis, D. Kanoulas, N. G. Tsagarakis, and P. Trahanias, "Outlier-Robust State Estimation for Humanoid Robots," in *IROS*, IEEE, 2019.

[47] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: LiDAR-Inertial-Camera Odometry," *arXiv preprint arXiv:1909.04102*, 2019.

[48] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang, "Visual-Inertial Localization With Prior LiDAR Map Constraints," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3394–3401, 2019.

[49] M. Miiller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, and W. Stürzl, "Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision," in *IROS*, pp. 3701–3708, IEEE, 2018.

[50] K. Eckenhoff, P. Geneva, J. Bloecker, and G. Huang, "Multi-Camera Visual-Inertial Navigation with Online Intrinsic and Extrinsic Calibration," in *ICRA*, pp. 3158–3164, IEEE, 2019.

[51] G. D. Arana, M. Joerger, and M. Spenko, "Efficient Integrity Monitoring for KF-based Localization," in *ICRA*, IEEE, 2019.

[52] E. Allak, R. Jung, and S. Weiss, "Covariance Pre-Integration for Delayed Measurements in Multi-Sensor Fusion," in *IROS*, IEEE, 2019.

[53] "Learning Wheel Odometry and IMU Errors for Localization, author=Brossard, Martin and Bonnabel, Silvere," in *ICRA*, IEEE, 2019.

[54] N. Gosala, A. Bühler, M. Prajapat, C. Ehmke, M. Gupta, R. Sivanesan, A. Gawel, M. Pfeiffer, M. Bürki, I. Sa, *et al.*, "Redundant Perception and State Estimation for Reliable Autonomous Racing," in *ICRA*, pp. 6561–6567, IEEE, 2019.

[55] Z. Zhang, S. Liu, G. Tsai, H. Hu, C.-C. Chu, and F. Zheng, "Pirvs: An Advanced Visual-Inertial SLAM System with Flexible Sensor Fusion and Hardware Co-design," in *ICRA*, pp. 1–7, IEEE, 2018.

[56] M. Brossard, S. Bonnabel, and A. Barrau, "Unscented Kalman Filter on Lie Groups for Visual Inertial Odometry," in *IROS*, IEEE, 2018.

[57] F. Poggenhans, N. O. Salscheider, and C. Stiller, "Precise Localization in High-definition Road Maps for Urban Regions," in *IROS*, IEEE, 2018.

[58] S. Arnold and L. Medagoda, "Robust Model-Aided Inertial Localization for Autonomous Underwater Vehicles," in *ICRA*, IEEE, 2018.

[59] D. Zhang, J. Gabaldon, L. Lauderdale, M. Johnson-Roberson, L. J. Miller, K. Barton, and K. A. Shorter, "Localization and Tracking of Uncontrollable Underwater Agents: Particle Filter Based Fusion of On-Body IMUs and Stationary Cameras," in *ICRA*, IEEE, 2019.

[60] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "GOMSF: Graph-Optimization based Multi-Sensor Fusion for Robust UAV Pose Estimation," in *ICRA*, pp. 1421–1428, IEEE, 2018.

[61] P. Geneva, K. Eckenhoff, and G. Huang, "Asynchronous Multi-Sensor Fusion for 3D Mapping and Localization," in *ICRA*, IEEE, 2018.

[62] H. F. Chame, M. M. Dos Santos, and S. S. da Costa Botelho, "Reliable Fusion of Black-box Estimates of Underwater Localization," in *IROS*, pp. 1900–1905, IEEE, 2018.

[63] R. Piché, "Online Tests of Kalman Filter Consistency," *International Journal of Adaptive Control and Signal Processing*, vol. 30, no. 1, pp. 115–124, 2016.

[64] C. Croarkin, P. Tobias, J. Filliben, B. Hembree, W. Guthrie, *et al.*, "NIST/SEMATECH e-Handbook of Statistical Methods," *NIST/SEMATECH*, 2006.

[65] "Baidu debuts Robotaxi ride hailing service in China, using self-driving electric taxis." https://www.marketwatch.com/story/baidu-debuts-robotaxi-ride-hailing-service-in-china-using-self-driving-electric-taxis-2019-09-26.

[66] P. D. Groves, "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, [Book review]," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 2, pp. 26–27, 2015.

[67] C4ADS, "Above Us Only Stars - Exposing GPS Spoofing in Russia and Syria." https://www.c4reports.org/aboveusonlystars.

[68] T. Nighswander, B. Ledvina, J. Diamond, R. Brumley, and D. Brumley, "GPS Software Attacks," in *CCS*, 2012.

[69] S. of California Department of Motor Vehicles, *California Commercial Driver Handbook: Section 2 – Driving Safely*. 2019. Available at https://www.dmv.ca.gov/portal/dmv/detail/pubs/cdl_htm/sec2.

[70] "California Vehicle Code 21663." https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=21663.

[71] "California Vehicle Code 21460." https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=21460.

[72] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[73] J. Su, J. He, P. Cheng, and J. Chen, "A Stealthy GPS Spoofing Strategy for Manipulating the Trajectory of an Unmanned Aerial Vehicle," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 291–296, 2016.

[74] W. Liu, C. Kwon, I. Aljanabi, and I. Hwang, "Cyber Security Analysis for State Estimators in Air Traffic Control Systems," in *AIAA Guidance, Navigation, and Control Conference*, p. 4929, 2012.

[75] Y. Mo and B. Sinopoli, "False Data Injection Attacks in Control Systems," in *Preprints of the 1st workshop on Secure Control Systems*, 2010.

[76] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False Data Injection Attacks Against State Estimation in Wireless Sensor Networks," in *49th IEEE Conference on Decision and Control (CDC)*, IEEE, 2010.

[77] "Apollo Data Open Platform." http://apollo.auto/index.html.

[78] N. Medeiros, N. Ivaki, P. Costa, and M. Vieira, "Software Metrics as Indicators of Security Vulnerabilities," in *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 216–227, IEEE, 2017.

[79] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K.-K. R. Choo, "A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks," *IEEE Transactions on Emerging Topics in Computing*, 2016.

[80] M. Brown, J. Crawford, S. Nordstrom, F. Scholl, and F. Mhlanga, "Understanding the Presence of Experiential Learning Opportunity Programs in the Information Security Field," in *Proceedings of the 2013 on InfoSecCD'13: Information Security Curriculum Development Conference*, p. 53, ACM, 2013.

[81] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 2013.

[82] D. Frossard and R. Urtasun, "End-to-End Learning of Multi-Sensor 3D Tracking by Detection," in *ICRA*, pp. 635–642, IEEE, 2018.

[83] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "VectorNet: Encoding HD Maps and Agent Dynamics from Vectorized Representation," in *CVPR*, 2020.

[84] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex Urban Dataset with Multi-Level Sensors from Highly Diverse Urban Environments," *IJRR*, vol. 38, no. 6, pp. 642–657, 2019.

[85] J. Solà, "Quaternion Kinematics for the Error-State Kalman Filter," *arXiv preprint arXiv:1711.02508*, 2017.

[86] ETH Zürich, "Ethzasl MSF Framework." https://github.com/ethz-asl/ethzasl_msf.

[87] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation," in *IROS*, 2013.

[88] Velodyne, "Velodyne HDL-32E Datasheet." https://velodynelidar.com.

[89] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[90] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst," *arXiv preprint arXiv:1812.03079*, 2018.

[91] B. Friedland, *Control System Design: An Introduction to State-Space Methods*. Courier Corporation, 2012.

[92] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[93] LG, "LGSVL Simulator: An Autonomous Vehicle Simulator." https://github.com/lgsvl/simulator.

[94] "Video demo for the FusionRipper attack in the paper." https://sites.google.com/view/cav-sec/fusionripper.

[95] "Waymo's Self-Driving Cars Are Near: Meet the Teen Who Rides One Every Day." https://www.bloomberg.com/news/features/2018-07-31/inside-the-life-of-waymo-s-driverless-test-family.

[96] "Uber is Bringing its Self-Driving Cars to Dallas." https://www.theverge.com/2019/9/17/20870969/uber-self-driving-car-testing-dallas.

[97] "Lyft and Aptiv Have Completed 50,000 Self-Driving Car Rides in Las Vegas." https://www.cnet.com/roadshow/news/lyft-aptiv-self-driving-car-50k-rides/.

[98] "Waymo's next-generation self-driving system can 'see' a stop sign 500 meters away." https://www.theverge.com/2020/3/4/21165014/waymo-fifth-generation-self-driving-radar-camera-lidar-jaguar-ipace.

[99] J.-A. Ting, E. Theodorou, and S. Schaal, "A Kalman Filter for Robust Outlier Detection," in *IROS*, IEEE, 2007.

[100] D. M. Akos, "Who's Afraid of the Spoofer? GPS/GNSS Spoofing Detection via Automatic Gain Control (AGC)," *NAVIGATION: Journal of the Institute of Navigation*, vol. 59, no. 4, pp. 281–290, 2012.

[101] M. L. Psiaki and T. E. Humphreys, "GNSS Spoofing and Detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[102] A. Ranganathan, H. Ólafsdóttir, and S. Capkun, "SPREE: A Spoofing Resistant GPS Receiver," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016.

[103] J. Magiera and R. Katulski, "Detection and Mitigation of GPS Spoofing Based on Antenna Array Processing," *Journal of applied research and technology*, vol. 13, no. 1, pp. 45–57, 2015.

[104] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, "Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 1018–1031, IEEE, 2018.

[105] P. Papadimitratos and A. Jovanovic, "GNSS-based Positioning: Attacks and Countermeasures," in *MILCOM 2008-2008 IEEE Military Communications Conference*, pp. 1–7, IEEE, 2008.

[106] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent Progress in Road and Lane Detection: A Survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[107] "Guide to Lane Departure Warning & Lane Keeping Assist." https://www.consumerreports.org/car-safety/lane-departure-warning-lane-keeping-assist-guide/.

[108] "Experimental Security Research of Tesla Autopilot." https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf, 2019.

[109] T. Sato, J. Shen, N. Wang, Y. J. Jia, X. Lin, and Q. A. Chen, "Security of Deep Learning based Lane Keeping System under Physical-World Adversarial Attack," *arXiv preprint arXiv:2003.01782*, 2020.

[110] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Synthetic 2D LIDAR for Precise Vehicle Localization in 3D Urban Environment," in *ICRA*, pp. 1554–1559, IEEE, 2013.

[111] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in *USENIX Security*, 2011.

[112] R. Baker and I. Martinovic, "Losing the Car Keys: Wireless PHY-Layer Insecurity in EV Charging," in *USENIX Security*, 2019.

[113] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock It and Still Lose It —on the (In)Security of Automotive Remote Keyless Entry Systems," in *USENIX Security*, 2016.

[114] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and Lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[115] C. Yan, W. Xu, and J. Liu, "Can You Trust Autonomous Vehicles: Contactless Attacks Against Sensors of Self-Driving Vehicle," *DEF CON*, vol. 24, 2016.

[116] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and Delivered: Fabricating Implicit Control over Actuation Systems by Spoofing Inertial Sensors," in *USENIX Security*, 2018.

[117] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors," in *USENIX Security*, 2015.

[118] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks," in *EuroS&P*, pp. 3–18, IEEE, 2017.

[119] H. Choi, W.-C. Lee, Y. Aafer, F. Fei, Z. Tu, X. Zhang, D. Xu, and X. Deng, "Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach," in *CCS*, 2018.

[120] R. Quinonez, J. Giraldo, L. Salazar, E. Bauman, A. Cardenas, and Z. Lin, "SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants," in *USENIX Security*, 2020.

[121] "2019 MKZ." https://www.lincoln.com/services/assets/Brochure?make=Lincoln&model=MKZ&year=2019.

[122] W. J. Stein and T. R. Neuman, "Mitigation Strategies for Design Exceptions," tech. rep., United States. Federal Highway Administration. Office of Safety, 2007.

A Calculation of Required Deviations in Attack Goals and Distances to Lane Line

The required deviations under off-road and wrong-way attacks are calculated based on common widths of the AV, lane, and road shoulder. These values differ in local and highway settings. Fig. 12 shows the measurements we used in the calculation. For the AV width, we use the width (including mirrors) of the Baidu Apollo's reference car, Lincoln MKZ [121]. For the lane widths and shoulder widths, we refer to the design guidelines [122] published by the US Department of Transportation Federal Highway Administration. For off-road attack, we use the deviation when the AV goes *beyond* the road shoulder from the center of the lane as the required deviation, which is

calculated using $\frac{L-C}{2} + S = 0.895m$ (local) and $1.945m$ (highway), where L is the lane width, C is the car width, and S is the road shoulder width. For wrong-way attack, we define the required deviation as the AV completely invades the neighbor lane, and it is calculated with $\frac{L}{2} + \frac{C}{2} = 2.405m$ (local) and $2.855m$ (highway). We calculate the deviation of *touching the lane line* using $\frac{L-C}{2}$, which is $0.295m$ on local roads and $0.745m$ on the highway.
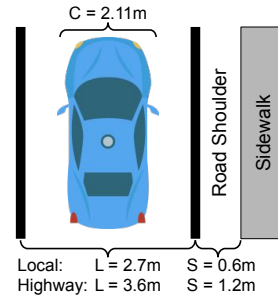

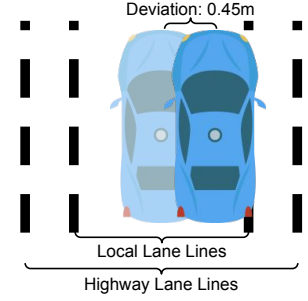Figure 12: Common AV, traffic lane, and road shoulder widths used in this paper.

Figure 13: Visualization of the lateral deviation 0.45 meters on local and highway roads.

B Convert Steering to Lateral Position and Heading Rate Changes

Fig. 14 shows the mathematical conversion from the steering angle to physical world lateral position change. The position change can be calculated as $\delta_{pos} = vt \sin(\frac{\theta}{\phi})$, where v is the velocity, t is the cycle time of the controller, θ is the steering angle, and ϕ is the steering ratio, which is a constant describing the ratio of the turning angle of the steering wheel to that of the vehicle wheel. The steering angle can be directly converted to heading rate change using $\delta_{\omega} = \theta/\phi t$, where δ_{ω} is the yaw (*i.e.*, heading) rate change.
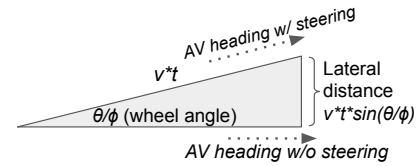

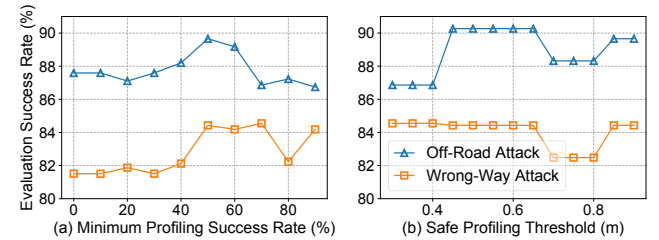Figure 14: Conversion from the steering wheel angle to lateral position change.

Figure 15: Profiling results when using different (a) minimum profiling success rates, and (b) safe profiling thresholds.