CommanderGPS: Practical Hidden Command Dissemination Through Pedal Operations

Edwin Yang, Guanchong Huang, and Song Fang
University of Oklahoma, Norman, OK, 73019, {edwiny, guanchong.huang, songf}@ou.edu

2024 IEEE Conference on Communications and Network Security (CNS) | 979-8-3503-7596-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/CNS62487.2024.10735572

***Abstract*—It is critical to enable a driver to control in-vehicle devices (e.g., a smartphone) during driving, especially when the driver is in danger. Existing in-vehicle communication often relies on speech communication or hand operation. However, such an approach is not always available (e.g., if the phone is out of reach or the driver is monitored and forced not to use it), and may also bring safety concerns by diverting attention from driving. In this work, we investigate the possibility of disseminating commands for a driver through pedal operation and develop *CommanderGPS*, a system that utilizes pedal operation to control vehicle movement as a transmitter, while an in-vehicle GPS-equipped device, which can capture vehicle movement, as a receiver. The discovered communication modality is inconspicuous and safe because it does not require the driver to use their voice, take their hands off the wheel, or remove their foot from the pedals. It still allows the driver to engage in communication while driving. We implement *CommanderGPS* on top of commercial off-the-shelf (COTS) GPS-equipped mobile devices and conduct a suite of experiments to validate its feasibility, efficiency, and security. Extensive experimental results demonstrate that *CommanderGPS* could achieve a success rate of above 96% for transmitting Morse code and a comparable text input speed to other Morse code based input systems. Meanwhile, an attacker is unable to recover the disseminated commands by stalking the target vehicle.**

***Index Terms*—Pedal control, Command dissemination, GPS**

I. Introduction

All modern motor vehicles are equipped with emergency flashers, which, when activated, can indicate that the vehicle temporarily meets a hazard through the simultaneous operation of a vehicle's all direction-indicator lamps [40]. However, the driver must physically push the hazard flasher button to initiate the warning signals, and the lights typically only warn drivers in neighboring vehicles. Thus, the automotive hazard warning system is often insufficient for a driver to send real-time emergency messages, such as calling the police.

It is well known that taxi or rideshare drivers are exposed to high risks of crimes or accidents [1], [35]. To help drivers in an emergency, ridehails such as Uber and Lyft provide emergency call features (e.g., direct 911 button) in their smartphone apps [17], [39]. However, they still require the driver to trigger the alert with either a keypress or voice command, which may not always be available to the driver. For example, it is reported that in 2017, an off-duty Uber driver was threatened by two armed criminals to drive them [44]; in 2020, a Uber driver was forced to drive to an ATM in a robbery [29]. In both cases, the victims were unable to call 911 while driving as the criminals were pointing a gun at them. Such incidents (e.g., carjacking and forcing the driver to drive) happen quite frequently (e.g., [20], [36]) and pose serious threats to the safety and security of the vehicle, the driver, and the passenger. Thus, developing a practical way of enabling a driver to secretly send out emergency messages while driving is critical and urgent.

Meanwhile, traditional in-vehicle message transmission methods via hand operation or voice may cause distracted driving, which diverts the driver's attention from driving [22]. During driving, a driver may fiddle with the in-vehicle infotainment system and operate a mobile phone. According to the AAA Foundation for Traffic Safety, drivers using in-vehicle voice-based or touch screen features may be mentally and physically distracted, potentially causing dangerous consequences [10]. Also, the National Safety Council estimates that at least 27% of all traffic crashes each year are caused by drivers using cell phones and texting [23]. One out of four accidents in the United States are caused by texting and talking on the phone while driving [16]. Even when using the autopilot system that can control the vehicle without constant manual control, the driver should keep both hands on the steering wheel at all times [34].

Besides, speech communication between the driver and a passenger within a moving vehicle is often difficult owing to the high noise level. Drivers may turn their heads to communicate with passengers, which is unsafe and may bring discomfort. Numerous research efforts have been proposed to suppress the noise signal and thus improve in-vehicle communication performance (e.g., [7], [32], [38]). It is desirable that there is a reliable way that enables a driver to safely interact with any other entity (e.g., phone, passenger, or vehicle).

The aforementioned bottlenecks in existing schemes (hand operation or voice) for a driver to send out messages during driving motivate us to think: *is there another communication modality available?* This paper takes the first step towards positively answering this question and explores the possibility of using vehicle movement as a new communication modality. Our basic idea relies on the following key observation: a driver utilizes a foot to control the vehicle movement, which any in-vehicle GPS-equipped devices can capture. Specifically, a driver can map a command (e.g., "turn on the music" or "call the police") into a sequence of pedal operation and perform it under safe driving circumstances (e.g., light traffic). At the same time, an in-vehicle device can decode them with collected vehicle movement data.

Currently, more than a billion devices (e.g., vehicles, smartphones, or tablets) equip GPS, providing estimates of velocity, position, and time instantaneously, continuously, and inexpensively [19]. The proposed system is named *CommanderGPS*, which enables a driver to send out commands to a GPS-equipped device without diverting attention from driving.

In practice, a driver may alternatively step on or release the accelerator and brake pedals with varying force to adapt to dynamic road traffic. Thus, an important technical challenge is how to convert a command into pedal operation, which would not be confused with the normal pedal operation for accommodating traffic. We analyze the driving patterns derived from the transitions of vehicular behavior (e.g., acceleration, deceleration, and uniform motion) during a certain period. The driving patterns are controlled by the driver and can be immediately observed by in-vehicle GPS-equipped devices. In particular, the switch between the accelerator and brake pedals shows through the structure of vehicular behavior transitions. For example, assume that a driver steps on pedals with appropriate force and in order of "A→B→A" (where 'A' and 'B' denote accelerator and brake pedals, respectively) during driving. Accordingly, with velocities collected by GPS, a receiver may observe three transitions of vehicular behavior, i.e., acceleration, deceleration, and acceleration. A driver can perform a specified sequence of pedal operation to generate corresponding driving patterns. Thus, if these driving patterns are used to encode commands by the driver, the receiver (an in-vehicle GPS-equipped device) can accordingly decode them by extracting driving patterns in the observed velocity stream.

However, if a driving pattern is frequently observed in everyday driving, many commands will be demodulated while the driver does not intend to send them. On the other hand, if the converted pedal operation is unsafe or difficult to perform, such as swiftly and frequently switching the vehicular behavior, the driver should not perform them for safety considerations. To overcome both restrictions, we propose a customized modulation method to translate chosen commands into *artificial driving patterns*, which are rarely observed in everyday driving and easily manipulable by the driver via pedal operation.

We point out that the discovered communication modality can be used without sacrificing safety, as the driver still focuses on driving when *CommanderGPS* is launched. Also, this communication channel is more flexible as it does not require the driver and the device to be in close physical proximity. Instead, the device can capture the commands anywhere in the vehicle. *CommanderGPS* is positioned as complementary to existing communication techniques using hand operation or voice and plays a critical role, especially when other methods are unavailable.

Nevertheless, to make *CommanderGPS* practical, non-trivial effort should be made to address the following issues:

- **Transmission errors**: To ensure that any in-vehicle GPS-equipped device can accurately extract the driving patterns specified by a driver, *CommanderGPS* should be robust against GPS errors and device heterogeneity.
- **Communication efficiency**: *CommanderGPS* should enable a driver to deliver a command within a reasonable time frame. Thus, the achievable data rates for *CommanderGPS* under varying traffic situations deserve discussion.
- **Message confidentiality**: The sent commands may contain private information, which should be kept secret from unauthorized users. An adversary may closely follow the target vehicle, aiming to infer the sent commands. It is a critical problem to prevent such stalking attacks.

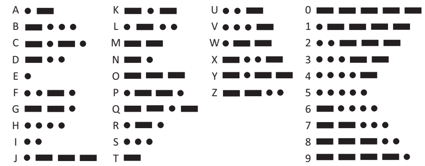

Fig. 1: The International Morse Code

To the best of our knowledge, we are the first to explore the possibility of using pedal operation to disseminate commands by solving the above three essential concerns. We implement and evaluate a prototype system using off-the-shelf GPS-equipped devices. Experimental results demonstrate that CommanderGPS can achieve a success rate of above 96% for command dissemination, with comparable input speed to other Morse code input systems (e.g., [25]). We also demonstrate that when an attacker follows a target vehicle, such a stalking attack is ineffective at recovering the disseminated commands.

II. Background on Morse Code

Morse code is a widely and flexibly used character encoding scheme for sending messages [3], [9], [25], [47]. Most people can memorize it in a few hours [6]. Therefore, we choose to transmit Morse code using pedal operations.

The International Morse Code [15] encodes each letter or number as a sequence of two signals named *dot* ($\cdot$) and *dash* ($-$), which are distinguishable by their durations. The dot duration is the basic unit of time measurement in Morse code transmission, and the duration of a dash is three times that of a dot. Each signal within a character is followed by a dot period of silence, and a dash period of silence separates the two characters. Figure 1 shows the Morse code chart for the 26 English letters (without distinguishing between upper case and lower case) and the 10 Arabic numerals.

In this research, we borrow the idea of using a combination of the dash and dot from the Morse code. We define three symbols - a dash, a dot, and an inter-signal spacing, into three different driving patterns. A command, composed of characters, can be thus represented with a sequence of these three symbols.

III. Driving Characteristics

A. Pedal Control Basics

A driver usually changes vehicle velocity by stepping on the accelerator or brake pedal, and each acts as a mechanical lever amplifying the force that the driver applies to the pedal. We begin with real-world experiments to show how a driver can change the vehicle's behavior by applying force on the pedals. An in-vehicle GPS-equipped device continuously captures the vehicle velocity. For comparison, we attach a force sensor on

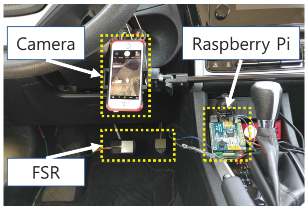

Fig. 2: Setup of the camera and the force sensor.

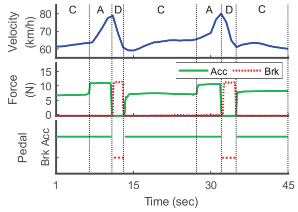

Fig. 3: Pedal control in different modalities.

each pedal to observe the force variation and record the whole process of pedal operation with a video camera.

Figure 2 shows the corresponding setup. On each pedal, we attach a force sensing resistor (FSR), i.e., Interlink Electronics FSR 406 [14], which is low cost and can translate applied force to the resistor (i.e., pedal) into a voltage output. Both FSRs are connected to a Raspberry Pi, which collects the generated output voltage and converts it into the force value. Besides, we mount a phone camera to monitor the foot position directly.

Figure 3 depicts the vehicle velocity, the applied force, and the pedal that the foot is stepping on, for a 45-second driving performed by an adult on an ordinary road. As the two pedals are close (around 30 cm apart), the time for moving the foot from one pedal to the other becomes negligible when the driver swiftly switches the pedal. We observe two major tendencies. First, the vehicle engine responds instantly with corresponding velocity variation when the driver presses down on the accelerator or brake pedal. Such quick responsiveness enables a driver to control the vehicle movement flexibly via foot force. Second, by alternatively applying varying force to either pedal, the driver generates a sequence of vehicle movement states, each of which may be the state of acceleration (A), deceleration (D), or constant velocity (C).

Let $a(t)$ denote the acceleration of the driving vehicle at time t. If $a(t)$ is too small (i.e., $|a(t)| \leq a_0$, where a_0 is a predetermined threshold), the vehicle state is regarded as C. The vehicle state then becomes A if $a(t) > a_0$ while turns to D if $a(t) < -a_0$. A driver can adjust the force applied to different pedals to generate desired vehicle movement states.

Vehicle Acceleration Model: Let M denote the load mass (i.e., the tare mass plus the load mass). According to the principle that the net force equals the product of the mass and the acceleration, the vehicle's acceleration $a(t)$ can be calculated as $\frac{F(t)-f(t)}{M}$ [28], where $F(t)$ and $f(t)$ are the tractive force and the environment-dependent resistive force (i.e., the sum of aerodynamic drag and tire rolling resistance [4]), respectively.

The tractive force depends on which pedal is stepped on and the applied force. Specifically, when the accelerator pedal is pressed, we have $F(t) = F_m * w_a(t)$, where F_m is the generated maximum tractive force with the engine at peak power and $w_a(t)$ denotes the degree of pushing the accelerator pedal; when the brake pedal is pressed, we have $F(t) = -B_m * w_b(t)$, where B_m is the maximum resistance force that the brake pedal induces, and $w_b(t)$ denotes the degree of pushing the brake pedal. Both of the pushing degrees, i.e., $w_a(t)$ and $w_b(t)$, lie in the range of [0, 1], with 0 denoting that no pedal force

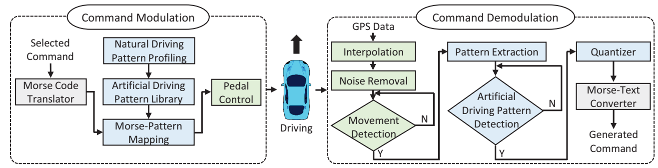

Fig. 4: *CommanderGPS* scheme flow chart.

is applied while 1 representing that the maximum force is enforced. Therefore, we have

$$a(t) = \frac{F_m * w_a(t) - B_m * w_b(t) - f(t)}{M}. \tag{1}$$

By manipulating $w_a(t)$ or $w_b(t)$, a driver can control the vehicle's acceleration.

B. Driving Characteristics Injection

A driver normally steps on the accelerator and brake pedals based on the traffic situation, which may be affected by various factors, including weather, light/dark, road surface construction, traffic signals, etc. Such driving activities based on the road and traffic situations are called *natural driving*. Vehicle movement state may vary among the states of A, D, and C. A driver can perceive vehicular behavior transitions if both consecutive vehicle movement states last at least a certain time Δt_0, and Δt_0 is referred to as the *minimum resolvable interval*. Without loss of generality, we have $\Delta t_0 = 1$ s. Those transitions reflect the characteristics of the pedal operation during driving.

On the other hand, we observe that a driver may actively perform the extra pedal operation safely. For example, on a clear straight road, a driver can alternately step on the accelerator and brake pedals with minor force for a certain period without interfering with the traffic. Such minute changes in vehicular behavior can be captured by in-vehicle GPS-equipped devices. If a driver can generate specified driving characteristics different from those in normal driving, the driver can thus use them as carriers to modulate commands, and any in-vehicle device can then decode the disseminated commands.

IV. The Design of CommanderGPS

A. System Overview

CommanderGPS provides a method for a driver to turn the accelerator and brake pedals into an input interface, as a GPS-equipped device can continuously sense the vehicle movement state variation occurring when the driver steps on the pedals during driving. Based on a selected message encoding scheme, a command that the driver intends to send will be first converted into specified driving patterns that are distinct compared with those exhibited during normal driving (i.e., when no command is sent). The driver can then inject the pedal operation, corresponding to those artificial driving characteristics, into the driving process.

On the other hand, pedal operation controls vehicle movement, which can be tracked by any GPS-equipped device (i.e., receiver) anywhere in the vehicle. With the collected velocities from GPS, the receiver first removes noise from the data stream.

As pedal operation cannot change the vehicle movement state when the vehicle stalls (i.e., not in driving mode), the receiver then separates the driving data from non-driving data (observed when the vehicle stalls). Next, the receiver extracts all driving patterns in the obtained driving data. It detects the artificial driving patterns, which will be finally converted into a command with the corresponding message decoding scheme. Figure 4 shows the flow chart of *CommanderGPS* when Morse code is applied to encode a command.

B. Command Modulation

Command modulation is the process of mapping commands into artificial driving patterns, and converting the mapped patterns into corresponding pedal operations.

1) Message Encoding: Different message encoding schemes can be utilized to disseminate commands. Intuitively, we can convert each command into a single artificial driving pattern. Let N denote the maximum number of available artificial driving patterns that can represent messages. With this scheme, however, the system only works for transmitting N' (up to N) commands, and meanwhile it requires the driver to remember all utilized N' artificial driving patterns, which would be a great burden to the driver when N' is not small (e.g., $N' \geq 5$). On the other hand, binary code, which uses a two-symbol system to represent messages, seems to solve the problem, i.e., only two different artificial driving patterns are required to denote each character, but it causes a usability problem – the driver may have difficulty to remember the conversions from each character to corresponding binary codes.

As Morse code is easy to master and only utilizes three symbols (e.g., dash, dot, and inter-signal space) to denote each character, we focus on transmitting Morse code.

2) Artificial Driving Pattern Determination: In general, a qualified artificial driving pattern for message modulation should satisfy the following two requirements: (1) *uniqueness:* it differs from natural driving patterns so that normal driving will not initiate *CommanderGPS*; (2) *maneuverability:* the pedal operation that can generate such patterns is not complex so that they can be smoothly performed by a driver.

Feature Selection: Ideally, a feature extracted from a sequence of GPS velocities would enable us to uniquely determine the corresponding injected message. Let T_s denote the feature perception time, i.e., a time window for injecting a transmission unit. Since the driver may not accurately estimate the time, T_s is set flexibly, i.e., $T_l \leq T_s \leq T_u$, where T_l and T_u are the lower and upper bounds, respectively.

A perfect feature would then classify natural driving and artificial driving into different categories, and further divide artificial driving into distinguishable groups so that the driver can send different messages by generating and delivering different artificial driving features. Our strategy is thus to find a feature that can separate artificial driving from natural driving, and also divide artificial driving into at least 3 groups for transmitting Morse code. With an N-sample velocity sequence captured during T_s, we then identify a feature from its self-contained relationship among the samples of the sequence.

Every two successive velocity samples generate a vehicle movement state ("A", "D" or "C"). The N-sample velocity sequence can be thus translated into a sequence of $N-1$ vehicle movement states, denoted with $\mathbf{S}_T = [s_1, \cdots, s_{N-1}]$, where s_i ($i \in \{1, \cdots, N-1\}$) represents the inter-sample movement state (i.e., state of the $i+1$ sample compared with the i-th sample). It is still a difficult task for a driver to memorize all vehicle movement states for every two successive velocity samples. We thus cannot utilize an inter-sample vehicle movement state to craft a feature. Instead, we combine successive same inter-sample vehicle movement states into one state, and therefore $\mathbf{S}_T$ will be translated into a sequence of alternative vehicle movement states $\mathbf{S} = [s_1, \cdots, s_M]$ ($M \leq N-1$), where $s_j \neq s_{j+1}$ ($j \in \{1, \cdots, M-1\}$). Based on $\mathbf{S}$, we build the following two features:

Length M: We empirically find that a driver does not frequently change the vehicle movement state within a short period in a normal driving situation, while more frequent vehicle movement state transitions can be injected with pedal operation when the road situation is allowable. Consequently, a smaller M usually denotes natural driving while a larger M denotes artificial driving. However, using M as a feature raises a dilemma. Let $\mathcal{U}_A$ be a set consisting of all possible values of M in artificial driving. On one hand, to enable message (Morse code) dissemination in artificial driving, the size of $\mathcal{U}_A$ should be no less than 3, driving the value of M larger. On the other hand, a larger M means that the driver needs to perform pedal operation more frequently. Such a requirement exerts a greater burden on the driver and is also prone to error.

Transition Sequence S: The length feature only provides the number of state variations in a sequence, and does not consider the exact state information for each variation. The transition sequence S provides both information and can be a better feature. In fact, except for the number of state variations, a driver is also able to identify the experienced vehicle movement states (e.g., via feeling the force that the acceleration/deceleration causes because of the body's inertia). Theoretically, according to the feature of transition sequence, if the vehicle maintains the same state during the entire period of T_s, there are 3 groups, i.e., A, D, and C; if the vehicle switches the state for just once, there are $3 * 2 = 6$ possibilities of the transition sequence, i.e., $A \to D$, $A \to C$, $C \to A$, $C \to D$, $D \to A$, and $D \to C$. Note that this period has at most $T_s - 1$ state transitions. Recursively, the maximum number of driving patterns can be calculated by $N_s = 3 + 3 \times 2^1 + \cdots + 3 \times 2^{T_s-2} = 3 \times (2^{T_s-1} - 1)$.

In natural driving, however, only a partial of these N_s driving patterns may appear, as it is usually unnecessary for the driver to change the vehicle movement state so quickly. Let N_n denote the number of driving patterns that appear in normal driving, and thus we have $N_n < N_s$. The rest $(N_s - N_n)$ patterns can be exploited for command injection.

3) Symbol-Pattern Mapping: Let S denote the set of all driving patterns extracted from an N-sample velocity stream, and $|S|$ denote the size of the set S. When a driver simply adjusts the pedal operation based on the traffic situation during

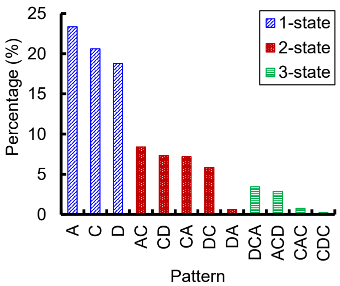

Fig. 5: Frequencies of different driving patterns in normal driving.

Demodulated Pattern \ Disseminated Pattern	X	Y	Z	ω
X	1.0	0.0	0.0	0.0
Y	0.0	0.96	0.02	0.0
Z	0.0	0.04	0.98	0.0
ω	0.0	0.0	0.0	1.0

Fig. 6: Confusion matrix for driving pattern demodulation.

regular driving, all detectable natural driving patterns construct a set S_n. Also, all driving patterns that a vehicle rarely experiences in regular driving but a driver can intentionally make the vehicle exhibit form another set S_i. It can be easily obtained that $S = S_n \cup S_i$. Ideally, if $S_n \cap S_i = \emptyset$ satisfies, natural and artificial driving patterns would not be confused with each other (i.e., the false positive of recognizing an artificial driving pattern would be 0). Empirically, with an appropriate T, the set S_i often has driving patterns from which a driver can select one or more for transmitting commands.

Intuitively, we can translate a Morse code symbol into an element in S_i. Afterward, the driver can securely perform pedal operation to enable the vehicle to exhibit this selected driving pattern. Figure 5 shows the frequency of detected driving patterns with a driver's 16-hour regular driving data when $T = 5$. The theoretical total number N of driving patterns equals $3 \times (2^4 - 1) = 45$. It is observed that there are only 12 driving patterns performed in regular driving, where 1-state and 2-state driving patterns compose more than 90% of all. Thus, there exists room for the driver to determine certain artificial driving patterns for communication with GPS-equipped devices. For example, the driver may choose the driving pattern 'CAC' as it only has a 0.45% probability of appearance in normal driving situations (this is because a driver usually does not accelerate just for a very short time in natural driving), and can be generated with ease. Within a 5-second interval, the driver can perform the following operations: maintaining a roughly constant speed, pressing the gas/accelerator pedal to accelerate, and soon releasing it to maintain the increased speed.

By selecting three artificial driving patterns representing the symbols of dot, dash, and inter-signal spacing, the driver can send morse codes of any comment by injecting the corresponding pedal operation. We conduct a preliminary experiment to show that the driver can disseminate the selected patterns without confusion, and the receiver experiences low pattern recognition errors. We let a driver choose three different driving patterns (i.e., X = 'CAD', Y = 'ADA', and Z = 'CDA') where they do not appear during natural driving. Let ω indicate a driving pattern different from X, Y, and Z. The driver injects each of the selected patterns and ω 50 times separately, while an in-vehicle GPS-equipped device records the vehicle movement states.

Figure 6 shows the derived confusion matrix, where the cells lie on a diagonal from (X, X) to (ω, ω) indicate probabilities of the correct recognition, while off-diagonal show probabilities of incorrect recognition. We see that at least 96% of the selected patterns are correctly recognized. Also, none of the artificial driving patterns that are not selected are recognized as the chosen driving patterns. These results imply that the driver and receiver can clearly distinguish the artificial driving patterns, while the receiver can experience minor recognition errors between similar artificial driving patterns sent by the driver.

C. Command Demodulation

Command demodulation converts the collected GPS velocities to corresponding commands.

1) Data Pre-processing: The received GPS time series is often noisy and may contain intermittent gaps (or missing data) due to various interferences (e.g., hardware design, obstructions by buildings [21]) to the satellite signals.

Interpolation: We occasionally observe GPS signal loss (i.e., null values) for one or multiple samples. Considering that the vehicle velocity usually cannot be changed drastically within a few seconds, to obtain those missing samples, we thus utilize linear interpolation [18], which estimates a new value by connecting two adjacent known values with a straight line.

Noise Reduction: We then perform discrete Fourier transform (DFT) to the GPS velocities and observe that the frequencies of GPS velocity variations due to pedal operations primarily lie at low frequencies. We thus adopt a Butterworth low-pass filter [27] to minimize the impact of the high-frequency noises.

Movement Detection: When a vehicle stalls (in park or neutral), its velocity should be 0. However, due to noises and GPS signal delay, this value sometimes may slightly deviate from 0. Meanwhile, we observe that GPS coordinates almost remain constant when a vehicle stops. Therefore, when the variations of both the vehicle velocity and the observed GPS coordinates are quite small (within pre-determined thresholds), the vehicle will be regarded as stopped. Correspondingly, the vehicle movement can be distinguished.

2) Pattern Extraction: Let $[v_1, \ldots, v_N]$ denote the collected N-sample velocity sequence. With two adjacent velocities v_i, and v_{i+1} ($i \in \{1, 2, \ldots, N-1\}$), the vehicle driving state $s_{i,i+1}$ for these two samples (referred to as sample-level driving state) can be calculated as

$$s_{i,i+1} = \begin{cases} \text{“}A\text{”}, & \text{if } v_{i+1} - v_i > \delta \\ \text{“}C\text{”}, & \text{if } v_{i+1} - v_i \leq \delta \\ \text{“}D\text{”}, & \text{if } v_i - v_{i+1} > \delta \end{cases}, \quad (2)$$

where "A", "C", and "D" denote acceleration, cruising (i.e., uniform motion), and deceleration, where δ serves as the reference level for tolerating velocity measurement errors. Based on Equation (2), we can calculate the corresponding sample-level driving state sequence $\mathbf{s} = [s_{1,2}, s_{2,3}, \cdots, s_{N-1,N}]$.

Since we focus on state transitions that are utilized by the driver to construct artificial driving patterns, the consecutive same sample-level driving states will be concatenated and regarded as one state of an interval (i.e., interval-level driving state). For example, a sample-level driving state sequence

$[A, D, D, C, C, A, A]$ will be translated into corresponding interval-level driving state sequence $[A, D, C, A]$. The process from a velocity stream into an interval-level driving state sequence can be denoted with a function below:

$$f: \mathbf{v} \mapsto P = \left[p_1, p_2, \cdots p_j\right],$$

where P is the obtained driving pattern and p_j denotes the j-th interval-level driving state.

The pattern extraction phase utilizes a dynamic sliding window T to translate velocity time series into driving patterns and search for artificial driving patterns that can be further decoded into commands. Let S_{P_A} denote the set of selected artificial driving patterns. Specifically, the receiver then executes the following steps (initially, we set $T = T_l$ (i.e., the lower bound of the input period, as discussed in Section IV-B2), the index $i = 1$, and a flag $flag = 0$):

(1) Take T consecutive velocities $\mathbf{v} = [v_i, \cdots, v_{i+T}]$;
(2) Run the function f to obtain the driving pattern P;
(3) When $P \in S_{P_A}$, if $T < T_u$ (i.e., the upper bound of the input period), we set $flag = 1$, $T = T_l + 1$; if $T = T_u$, an artificial driving pattern is detected, and we set $i = i + T$ and reset $flag = 0$ and $T = T_l$;
(4) When $P \notin S_{P_A}$, if $flag = 0$, i.e., no artificial driving pattern is detected, we then enable $i = i + 1$; if $flag = 1$, i.e., an artificial driving pattern P is detected, we then reset $flag = 0$ and $T = T_l$, and enable $i = i + T$;
(5) Jump to step (1).

3) Command Decoding: With the obtained artificial driving pattern sequence, the receiver runs quantization to generate dash, dot, and inter-signal spacing (or character spacing), which is an inverse process of Morse-pattern mapping. Next, all symbols within two successive inter-signal spacings will be converted into a character using the public Morse code encoding scheme. Lastly, the command is delivered with a grammar check, which parses the character sequence into words. The receiver can thus execute the command correspondingly.

D. Security Analysis

1) Co-located Attack: An adversary has a GPS-equipped device physically co-located with the driver in the vehicle, so that she can launch a co-located attack, where she attempts to monitor messages sent by the driver to the outside world by observing the vehicle movement data caused by pedal operation. This attack can be avoided via physical security. In the following, we discuss that the proposed scheme can still defend against such attacks by well-regulating parameters.

On one hand, the value range of the flexible period T, during which the driver generates artificial driving patterns, can be set dynamically. On the other hand, we observe that the number of candidate artificial driving patterns is usually larger than the number of those actually utilized, as discussed in Section IV-B. An attacker, thus, may not know which artificial driving patterns are used to represent meaningful information. Therefore, it imposes a strong requirement for the adversary to obtain T as well as the Morse-pattern mapping. Meanwhile, the driver may also insert meaningless artificial driving states through pedal operation to further confuse the adversary.

2) Stalking Attack: We relax the above requirements for an adversary outside of the target vehicle and discuss a more suspicious attack, i.e., a stalking attack, where a sneaky and lucky adversary successfully guesses the parameters the target driver takes, and follows the trajectory of the target vehicle by driving another vehicle. It is normally impossible for such an adversary (or stalker) to directly measure the driving velocity of the preceding vehicle all the time due to the requirement of dedicated speed measurement systems and the line-of-sight obstruction of various traffic factors such as roadside objects or other running vehicles on the road. Instead, the adversary may take another strategy, i.e., following the preceding vehicle on the same route and attempting to make her own vehicle movement similar to the preceding vehicle, so that she can demodulate the commands that the driver disseminates with the collected GPS data on her vehicle.

However, the stalker and the target driver cannot synchronize the pedal operations for two reasons: (1) they may have varying distance perception accuracy and reaction time; and (2) the two vehicles in fact experience similar but different traffic situations due to uncontrollable traffic flow. As a result, there is an inevitable mismatch between the observed vehicle movements of the target driver and the stalker. Therefore, even if a stalker can figure out the parameters that *CommanderGPS* uses, she is still unable to identify the disseminated commands. We verify this via real-world experiments in Section V-B.

3) GPS Jamming and Spoofing Attacks: An active attacker may send a strong radio signal over the air to jam GPS satellite signals from reaching GPS receivers or even inject falsified (fake) GPS signals to mislead GPS receivers [24], [37], [48]. However, to successfully launch such attacks, an adversary needs to place a malicious jammer or spoofer in either the target vehicle or another vehicle (or a drone) that always closely follows the target vehicle [48]. This imposes a strong requirement for an adversary.

Moreover, a GPS jamming attack simply disables a receiver's ability to decode commands via the observed vehicle movement while a GPS spoofing attack is more suspicious as an adversary may inject a fake trajectory into fake GPS signals. However, an adversary must also carefully select spoofed routes to make them consistent with the physical road network for GPS spoofing; otherwise, there often exists a noticeable contradiction between the vehicle's actual routing on physical roads and the corresponding GPS trajectory [48]. To detect GPS jamming and spoofing attacks, a GPS receiver may cross-check the collected GPS data with dead reckoning results based on inertial measurement unit (IMU) sensors (e.g., accelerometers) [48] or with the messages received from the mobile cellular network [26].

V. Experimental Evaluation

We implement the prototype of *CommanderGPS* and evaluate it in real-world scenarios. The receiver device can be an off-the-shelf GPS device. The driver aims to disseminate a

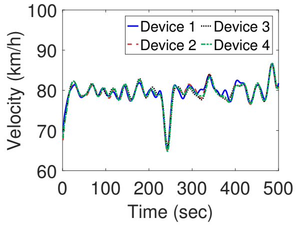

Fig. 7: Velocity data from devices in the same vehicle.

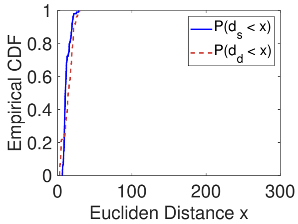

Fig. 8: The empirical CDFs of measurement similarities.

command (consisting of multiple Morse code symbols) during driving. The following metrics are utilized for evaluation:

- *Symbol Success Rate*: the number of successfully demodulated symbols over the total number of disseminated symbols (i.e., demodulation accuracy).
- *Symbols per Minute (SPM)*: the number of Morse-code symbols transmitted in a minute.
- *Characters per Minute (CPM):* the number of characters transmitted in a minute.
- *False Positive Rate*: the number of unintended symbols demodulated over the total count of obtained symbols.
- *False Negative Rate*: the number of symbols not demodulated over the total count of obtained symbols.

A. *Impact of Device Type and Proximity*

Varying GPS-equipped devices may be used and can be placed at different locations. We explore whether device type and proximity cause observed vehicle movement data differences. For each device in each case, we conduct 100 rounds of experiments, each collecting 500 GPS samples.

Device Type: We put four devices (i.e., Apple iPhone 11 Pro Max, LG Nexus 5X, Motorola MotoTab, and Raspberry Pi 3 connected with a u-blox NEO-6M [41] GPS module) together within a running vehicle, and refer to each as Device 1 to 4, respectively. Figure 7 shows an example of their obtained velocity streams. We see that the streams are highly similar.

Proximity: We collect GPS data for two scenarios: (1) place the devices at the same location (i.e., passenger seat), and (2) place the devices at different in-vehicle locations (i.e., armrest storage box, dashboard, back seat, and floor). For the above two configurations, we derive the respective Euclidean distances, d_s and d_d, between obtained GPS data from each pair of devices, respectively. Figure 8 plots the empirical cumulative distribution functions (CDFs) of d_s and d_d scenarios. We see that 98.1% of d_s and 97.2% of d_d are less than 25. The results demonstrate that device type and proximity yield a trivial impact on the accuracy of the observed velocity data.

B. *Modulation Efficiency and Security*

1) Efficiency: In the command modulation phase, each Morse code symbol is mapped into an artificial driving pattern characterized by two important parameters, i.e., the duration T and the number L_s of vehicle movement states in an artificial driving pattern. We select four common commands: *SOS*, *mute*, *radio*, and *call*. 'SOS' is a standard distress signal in an

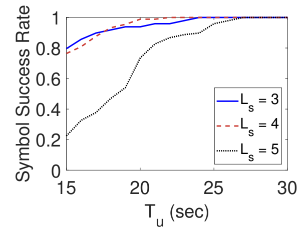

Fig. 9: Symbol success rate.

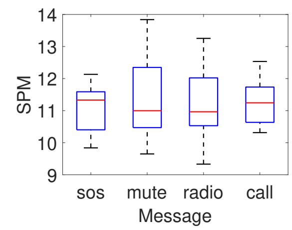

Fig. 10: Symbol input speed.

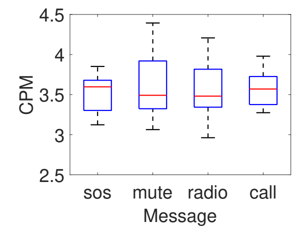

Fig. 11: Character input speed.

emergency [13]; others aim to turn off the sound, play the radio, and make a phone call, respectively. We enable the driver to send each command 50 times. Figure 9 shows the symbol success rate under different input duration T_u (i.e., maximum T) and pattern length L_s. For all L_s, with T_u increasing, the symbol success rate increases and approaches 100%. This shows that the driver requires enough time to inject the artificial driving patterns successfully. Also, with a larger L_s, a longer T_u is required to achieve 100% symbol success rate. Specifically, when T_u is above 18, 22, and 27 seconds for injecting the artificial driving patterns with 3, 4, and 5 movement states, respectively, we observe 100% symbol success rates.

Figures 10 and 11 demonstrate the achieved SPM and CPM. The top and bottom of each whisker denote the maximum and the minimum values. The top and bottom of the box are 25^{th} and 75^{th} percentile, and the horizontal line in the box represents the median. We can see that the three commands (i.e., 'mute', 'radio', and 'call') introduce similar ranges of SPM (9.3 - 13.8) and CPM (2.9 - 4.4). The command 'mute' presents the highest possible SPM and CPM (13.8 and 4.4), as it requires the driver to input the shortest Morse code sequence. In Figure 12, low average symbol FPR ($< 1\%$) and FNR ($< 0.71\%$) are observed.

2) Security: We assume a strong attacker who knows the mapping between the selected artificial driving patterns and the Morse code symbols. We allow a driver to disseminate the four commands to a receiver device. The attacker aims to mimic the movement states of the preceding vehicle, and the attacker's GPS-equipped device attempts to demodulate the commands.

Figure 13 illustrates the symbol success rates for the legitimate user and the attacker. We see that the attacker obtains up to 25% of the disseminated Morse code symbols. However, as the attacker always gets intermittent Morse code symbols, which do not provide sufficient clues for the correct commands, she fails to recover all disseminated messages. Meanwhile, the legitimate user successfully recognizes at least 96% of the symbols.

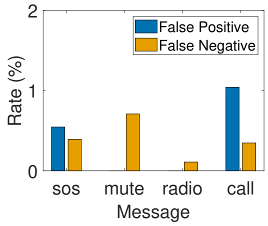

Fig. 12: FPR and FNR over different commands.

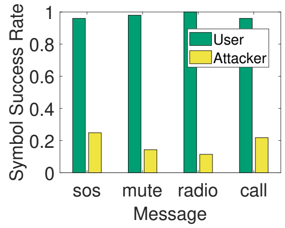

Fig. 13: Symbol success rate for the user/attacker.

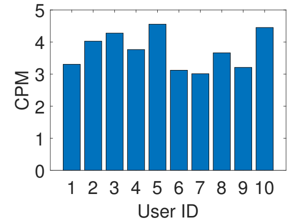

Fig. 14: Message input speed of 10 different users.

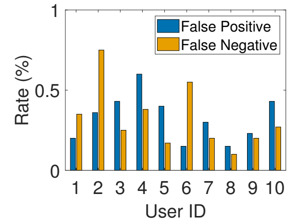

Fig. 15: FPR and FNR over different users.

TABLE I: Performance under different traffic conditions.

Traffic	SPM/CPM			Symbol Success Rate
	Minimum	Maximum	Average	
Heavy	7.9/2.5	9.5/3.0	8.6/2.8	97.6%
Moderate	9.5/3.1	10.2/3.2	9.8/3.1	98.9%
Light	9.5/3.0	13.0/4.1	12.3/3.9	100%

C. Impact of Traffic Flow

We consider three levels of traffic flow, i.e., heavy, moderate, and light, according to a common traffic classification method [5]. With the weekday state traffic statistics, we accordingly select a commuting route and three different time slots, i.e., 10-11 pm, 4-5 pm, and 8-9 pm, for light, moderate, and heavy traffic, respectively. We let the driver inject a command 'SOS' 50 times separately, during their daily driving.

Table I shows the corresponding performance. We see that the driver can achieve the smallest SPM/CPM (i.e., 8.6/2.8) on average under heavy traffic flow, while light traffic yields the best mean SPM/CPM (12.3/3.9). This is because the driver has limited opportunities to generate the artificial driving pattern under the heavier traffic flow (e.g., frequent slow down). The symbol success rate for all traffic conditions is consistently high (above 97.5%), and it reaches 100% for light traffic.

D. User Study

We recruited 10 participants (U1-U10; aged 21-32 years old; 3 self-identified as females and 7 as males) and asked each to disseminate messages during daily driving. Our study has been approved by our institutional IRB. We allow each user to disseminate the message 50 times. After the experiment, each participant compares the demodulated and sent messages to quantify the dissemination accuracy. Figure 14 shows the achieved average CPMs. We see that the CPM ranges from 3.01 to 4.56. For all users, the average CPM is 3.74, with a standard deviation of 0.57. Users 5 and 10 present higher average CPMs (4.56 and 4.45) than the others. This is because they manipulate the accelerator and brake pedals more frequently, thus disseminating more symbols over the same driving period. Figure 15 shows the corresponding average FPR and FNR, which do not exceed 0.60% and 0.75%, respectively. The results verify the robustness of *CommanderGPS* across different users.

VI. Related Work

Communication Over Covert Channel: Various covert channels have been employed to hide message transmission. For example, light from light-emitting-diode (LED) [12], audio [43], ultrasonic signal [11], vibration [30], finger gesture [31], WiFi channel state information (CSI) [25], [33], or human breath monitoring [2] are utilized to deliver information.

Morse Code Based Input Schemes: Morse code can be used to disseminate messages [2], [25], [31]. For example, [25] allows people with Motor Neuron Disease (MND) to send Morse codes by moving their finger on a table, generating interferences in WiFi CSI, which can be captured and in turn translated into sent Morse codes (i.e., messages). *CommanderGPS* also disseminates messages via Morse code, while it relies on popular GPS receivers to demodulate them.

GPS Applications in a Vehicle: GPS receivers are widely utilized in navigation systems to help drivers find a way to their destinations. Also, GPS data may improve the physical security of a vehicle. It is well known that Remote Keyless Entry Systems on modern vehicles are vulnerable to relay attack [8]. To resist relay attacks, the user's current GPS location is utilized as context information for authentication [42]. Moreover, GPS data obtained by the devices in the same vehicle can be utilized for secret key establishment [45], [46]. In contrast, *CommanderGPS* is the first study to utilize GPS as a communication modality.

VII. Conclusion

We propose *CommanderGPS*, a novel in-vehicle message dissemination scheme where a driver can transmit an encoded message by changing the movement state of the vehicle, and a GPS-enabled device in the same vehicle can decode the driver's message. The key idea is that vehicle movement can be utilized as a secure communication channel, as the driver can naturally change the vehicle movement state while driving. We develop a Morse code-based message modulation scheme that utilizes selected artificial driving patterns, and allows the receiver device to demodulate the observed vehicle movement pattern in GPS data. Our real-world experiment results show that the receiver achieves at least 96% demodulation success rate. Meanwhile, an attacker who follows the target vehicle is unable to intercept the messages sent by the driver.

Acknowledgements

We would like to thank all anonymous reviewers for their insightful comments. This work was supported in part by the National Science Foundation under Grant No.2155181.

References

[1] Bureau of Labor Statictics, "Fatal occupational injuries to independent workers," August 2017. [Online]. Available: https://www.bls.gov/opub/btn/volume-8/fatal-occupational-injuries-to-independent-workers.htm

[2] Cao, Qiang, Yu, Chuang, Cheng, Xue-Feng, Sun, Wu-Ji, He, Jing-Hui, Li, Na-Jun, Li, Hua, Chen, Dong-Yun, Xu, Qing-Feng, and Lu, Jian-Mei, "Polysquaramides: Rapid and stable humidity sensing for breath monitoring and morse code communication," *Sensors and Actuators B: Chemical*, vol. 320, p. 128390, 2020.

[3] Cheng-Hong Yang, Li-Yeh Chuang, Cheng-Huei Yang, and Ching-Hsing Luo, "Morse code application for wireless environmental control systems for severely disabled individuals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 4, pp. 463–469, Dec 2003.

[4] M. Ehsani, K. M. Rahman, and H. A. Toliyat, "Propulsion system design of electric and hybrid vehicles," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 1, pp. 19–27, Feb 1997.

[5] Federal Highway Administration, "Human factors design guidelines for advanced traveler information systems (atis) and commercial vehicle operations (cvo)," https://www.fhwa.dot.gov/publications/research/safety/98057/, 2016.

[6] D. Finley, *Morse Code: Breaking the Barrier*. M F J Enterprises, Incorporated, 1998. [Online]. Available: https://books.google.com/books?id=WAFuAAAACAAJ

[7] J. Franzen and T. Fingscheidt, "A delay-flexible stereo acoustic echo cancellation for dft-based in-car communication (ICC) systems," in *INTERSPEECH*, 2017, pp. 181–185.

[8] 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 16*, 2016.

[9] Google, "Use morse code," 2024. [Online]. Available: https://support.google.com/accessibility/android/answer/9011881

[10] A. Gross, "New vehicle infotainment systems create increased distractions behind the wheel," 2017. [Online]. Available: https://newsroom.aaa.com/2017/10/new-vehicle-infotainment-systems-create-increased-distractions-behind-wheel/

[11] M. Guri, Y. Solewicz, and Y. Elovici, "Mosquito: Covert ultrasonic transmissions between two air-gapped computers using speaker-to-speaker communication," in *IEEE DSC*, 2018, pp. 1–8.

[12] M. Guri, B. Zadov, and Y. Elovici, "Led-it-go: Leaking (a lot of) data from air-gapped computers via the (small) hard drive led," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2017, pp. 161–184.

[13] L. S. Howeth, *History of communications electronics in the United States Navy*. For sale by the Superintendent of Documents, US Govt. Print. Off., 1963.

[14] Interlink Electronics, "FSR model 406," 2024. [Online]. Available: https://buyinterlinkelectronics.com/collections/new-standard-force-sensors/products/fsr-model-406

[15] International Telecommunication Union, "Recommendation itu-r m.1677-1," October 2009. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1677-1-200910-I%21%21PDF-E.pdf

[16] G. Kratsas, "Cellphone use causes over 1 in 4 car accidents," Marchr 2014. [Online]. Available: https://www.usatoday.com/story/money/cars/2014/03/28/cellphone-use-1-in-4-car-crashes/7018505/

[17] Lyft, "Safety," 2024. [Online]. Available: https://www.lyft.com/safety

[18] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, March 2002.

[19] Misra, P. and Enge, P., *Global Positioning System: Signals, Measurements, and Performance*, Revised 2nd edition ed. Ganga-Jamuna Press, Lincoln MA, 2010.

[20] S. Mross, "Alleged carjacker faces multiple felony charges," February 2021. [Online]. Available: https://www.hotsr.com/news/2021/feb/01/alleged-carjacker-faces-multiple-felony-charges/

[21] National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS Accuracy," March 2022. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/

[22] National Highway Traffic Safety Administration, "Distracted driving," 2023. [Online]. Available: https://www.nhtsa.gov/risky-driving/distracted-driving

[23] National Safety Council, "Cell phone policy kit," 2023. [Online]. Available: https://injuryfacts.nsc.org/motor-vehicle/motor-vehicle-safety-issues/distracted-driving/

[24] T. Nighswander, B. Ledvina, J. Diamond, R. Brumley, and D. Brumley, "GPS software attacks," in *Proceedings of the ACM CCS*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 450–461.

[25] K. Niu, F. Zhang, Y. Jiang, J. Xiong, Q. Lv, Y. Zeng, and D. Zhang, "Wimorse: A contactless morse code text input system using ambient wifi signals," *IEEE Internet of Things Journal*, 2019.

[26] G. Oligeri, S. Sciancalepore, O. A. Ibrahim, and R. Di Pietro, "Drive me not: GPS spoofing detection via cellular network: (architectures, models, and experiments)," in *Proceedings of ACM WiSec*, 2019, pp. 12–22.

[27] Oppenheim, Alan V., Willsky, Alan S., and Nawab, S. Hamid, "Signals & systems (2nd ed.)," Upper Saddle River, NJ, USA, 1996.

[28] H. Rakha, M. Snare, and F. Dion, "Vehicle dynamics model for estimating maximum light-duty vehicle acceleration levels," *Transportation Research Record*, vol. 1883, no. 1, pp. 40–49, 2004.

[29] Rappahannock Media LLC, "Uber driver held at gunpoint, forced to drive to atm in robbery," 2020. [Online]. Available: https://www.insidenova.com/news/crime_police/prince_william/uber-driver-held-at-gunpoint-forced-to-drive-to-atm-in-robbery/article_b48ed312-89ac-11ea-9936-ff9589a7acc4.html

[30] Roy, Nirupam, Gowda, Mahanth, and Choudhury, Romit Roy, "Ripple: Communicating through physical vibration," in *USENIX Symposium on Networked Systems Design and Implementation*, 2015, pp. 265–278.

[31] H. Sachdeva, A. Misra, K. Chauhan, and M. Dave, "Morse-comm: Mobile application environment for visually-impaired and neuro-muscular disabled people using morse code conversion," in *Proceedings of IEEE SPIN*, 2020, pp. 809–813.

[32] G. Schmidt and T. Haulick, "Signal processing for in-car communication systems," *Signal Processing*, vol. 86, no. 6, pp. 1307 – 1326, 2006, applied Speech and Audio Processing.

[33] M. Schulz, J. Link, F. Gringoli, and M. Hollick, "Shadow Wi-Fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over Wi-Fi," in *Proceedings of ACM MobiSys*, 2018, pp. 256–268.

[34] Tesla, "Autopilot and full self-driving," 2024. [Online]. Available: https://www.tesla.com/support/autopilot

[35] Texas Department of Insurance, "Taxi, ride-hailing, and chauffeur drivers safety fact sheet," 2024. [Online]. Available: https://www.tdi.texas.gov/pubs/videoresource/fstaxisafe.pdf

[36] A. Thornhill, "Uber driver robbed and abducted at gun point," May 2020. [Online]. Available: https://potomaclocal.com/2020/05/02/uber-driver-robbed-abducted-at-gun-point/

[37] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proceedings of ACM CCS*, 2011, pp. 75–86.

[38] V. Tourbabin, I. Malka, and E. Tzirkel-Hancock, "Performance of fixed in-car microphone array beamformer under variations in car noise," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, March 2017, pp. 21–25.

[39] Uber, "Uber's emergency button," 2019. [Online]. Available: https://www.uber.com/newsroom/emergencybutton/

[40] United Nations, "Uniform provisions concerning the approval of vehicles with regard to the installation of lighting and light-signalling devices," October 2014. [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/2015/R048r12e.pdf

[41] U-blox, "NEO-6 series," 2024. [Online]. Available: https://www.u-blox.com/en/product/neo-6-series

[42] J. Wang, K. Lounis, and M. Zulkernine, "Cskes: A context-based secure keyless entry system," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2019, pp. 817–822.

[43] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Messages behind the sound: Real-time hidden acoustic signal capture with smartphones," in *Proceedings of ACM MobiCom*, 2016, pp. 29–41.

[44] M. Woodrow, "Off-duty uber driver forced to drive to richmond at gunpoint," May 2017. [Online]. Available: https://abc7news.com/news/off-duty-uber-driver-forced-to-drive-to-richmond-at-gunpoint/1964703/

[45] E. Yang and S. Fang, "GPSKey: GPS based secret key establishment for intra-vehicle environment," in *Fourth International Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2022, pp. 1–8.

[46] E. Yang, S. Fang, and D. Shen, "DASK: Driving-assisted secret key establishment," in *IEEE CNS*, 2022, pp. 73–81.

[47] Z. Yin, W. Jiang, S. M. Kim, and T. He, "C-morse: Cross-technology communication with transparent morse coding," in *IEEE INFOCOM 2017*, May 2017, pp. 1–9.

[48] 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, pp. 1527–1544.