Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study

Ishtiaq Rouf[a], Rob Miller[b], Hossen Mustafa[a], Travis Taylor[a], Sangho Oh[b]

Wenyuan Xu[a], Marco Gruteser[b], Wade Trappe[b], Ivan Seskar[b]

[a] Dept. of CSE, Univ. of South Carolina, Columbia, SC USA

{rouf, mustafah, taylort9, wyxu}@cse.sc.edu

[b] WINLAB, Rutgers Univ., Piscataway, NJ USA

{rdmiller, sangho, gruteser, trappe, seskar}@winlab.rutgers.edu

February 6, 2010

Abstract

Tire Pressure Monitoring Systems represent, to the best of our knowledge, the first in-car wireless network mandated for every new automobile. The security and privacy implications of such in-car wireless sensor networks are not fully understood— in particular, it is unclear whether the metal car body sufficiently shields these low-power transceivers from eavesdropping and outside interference. To address these questions, this paper presents a privacy and security evaluation of two tire pressure monitoring systems using both laboratory experiments with isolated tire pressure sensor modules and experiments with a complete vehicle system. We find that the sensor messages can be sniffed and decoded up to 40m from a passing vehicle with a basic low-noise amplifier and the openly available GNU radio platform. The proprietary protocols contain a static 32 bit identifier, and can be relatively easily reverse-engineered. In addition, a sensor message can be triggered through a wireless activation signal. This raises location privacy risks because vehicles could potentially be tracked through these identifiers and drivers do not have any option to disable the system. Furthermore, current protocols do not employ authentication mechanisms and vehicle implementation do not appear to perform basic input validation or filtering of messages. This allows straightforward spoofing of sensor messages. One of our experiments demonstrates this by triggering the tire pressure warning message in a moving vehicle through a spoofed message from another nearby vehicle. The paper concludes with a set of recommendations and a key management protocol that could significantly improve privacy and security of the tire pressure monitoring system and other forthcoming in-car wireless sensor networks.

1 Introduction

The quest for increased safety and efficiency of automotive transportation system is leading car makers to integrate wireless communication systems into automobiles. While vehicle-to-vehicle and vehicle-to-infrastructure communication systems [4] have received much attention, the first wireless network installed in every new vehicle is actually an in-vehicle sensor network: the tire pressure monitoring system (TPMS). The wide deployment of TPMSs in the United States is an outgrowth of the TREAD Act [32] resulting from the Ford-Firestone tire failure controversy [17]. Beyond preventing tire failure, however, alerting drivers about significantly underinflated tires promises to increase overall road safety and fuel economy because proper tire inflation improves traction, braking distances, and tire rolling resistance. These benefits have recently lead to similar legislation in the European Union [7] which mandates TPMSs on all new vehicles starting in 2012.

Tire Pressure Monitoring Systems continuously measure air pressure inside all tires of passenger cars, trucks, and multipurpose passenger vehicles, and alert drivers if any tire is significantly underinflated. While both direct and indirect measurement technologies exist, only direct measurement has the measurement sensitivity that is required by the TREAD Act and therefore, the only one in production. A *direct measurement* system uses battery-powered pressure sensors installed inside each tire to measure tire pressure and can typically detect any loss greater than 1.45 psi [37]. Since a wired connection from a rotating tire to the vehicle's electronic control unit is difficult to implement, the sensor module communicates its data via a wireless radio frequency transmitter. The receiving tire pressure control unit, in turn, analyzes the data and can send results or commands to the central car computer over the Controller-area Network (CAN) to trigger a warning message on the vehicle dashboard, for example. The sensor module is designed for a battery life of more than 10 years, to be able to outlive the tire. *Indirect measurement* systems infer pressure differences between tires from differences in the rotational speed, which can be measured using the anti-lock braking system (ABS) sensors. Since pressure affects the circumference of a tire, a lower-pressure tire has to rotate faster to travel the same distance as a higher-pressure tire. The disadvantages of this approach are that it is less accurate, requires calibration by the driver, and cannot detect the simultaneous loss of pressure from all tires (for example, due to temperature changes). While initial versions of the TREAD Act allowed indirect technology, updated rulings by the United States National Highway Transportation Safety Administration (NHTSA) have required all new cars sold or manufactured after 2008 in the United States to be equipped with direct TPMS [32] due to these disadvantages.

1.1 Security and Privacy Risks

Security and privacy aspects of vehicle-to-vehicle and vehicle-to-infrastructure communication have received significant consideration by both practitioners and researchers [3,33]. However, the already deployed in-car sensor communication systems have received little attention, because (i) the short communication range and metal vehicle body may render eavesdropping and spoofing attacks difficult and (ii) tire pressure information appears to be relatively innocuous. While we agree that the safety-critical application scenarios for vehicle-to-vehicle communications face higher security and privacy risks, we believe that even current tire pressure measurement systems present potential for misuse.

First, wireless devices are known to present tracking risks through explicit identifiers in protocols [20] or identifiable patterns in waveforms [10]. Since automobiles have become an essential element of our social fabric — they allow us to commute to and from work; they help us take care of errands like shopping and taking our children to day care — tracking automobiles presents substantial risks to location privacy. There is significant interest in wireless tracking of cars, at least for traffic monitoring purposes. Several entities are using mobile toll tag readers [5] to monitor traffic flows. Tracking through the TPMS system, if possible, would raise greater concerns because the use of TPMS is not voluntary and they are hard to deactivate.

Second, wireless devices make it easier to jam or spoof messages because no physical connection is necessary. While spoofing a low tire pressure readings does not appear to be critical at first, it will lead to a dashboard warning and will likely cause the driver to pull over and inspect the tire. This presents ample opportunities for mischief and criminal activities, if past experience is any indication. Drivers have been willing to tinker with traffic light timing to reduce their commute time [6]. It has also been reported that highway robbers make drivers pull over by puncturing the car tires [22] or by simply signaling a driver that a tire problem exists. If nothing else, repeated false alarms will undermine drivers' faith in the system and lead them to ignore subsequent TPMS-related warnings, thereby making the TMPS system ineffective.

To what extent these risks apply to TPMS and more generally to in-car sensor systems, however, remains unknown. A key question to judge these risks is whether the range at which messages can be overheard or spoofed is large enough to make such attacks feasible from outside the vehicle. While similar range questions have recently been investigated for RFID devices [24], the radio propagation environment in an

automobile setting with shielding from and strong reflections off metal car bodies is different enough to warrant further study. It is also unclear whether the TPMS message rate is high enough to make tracking of vehicles feasible. Thus, this paper aims to fill this void and conduct a security and privacy analysis of state-of-the art commercial tire pressure monitoring systems, as well as conducting detailed measurements of the communication range for in-car sensor transmissions.

1.2 Contributions

Our experimental analysis of two popular TPMSs used in a large fraction of vehicles in the United States yielded the following results.

Lack of security measures. The TPMS communications are based on standard modulation schemes and trivial protocols. Since the protocols do not rely on cryptographic mechanisms, the communication can be relatively easily reverse-engineered with GNU Radio [2] in conjunction with the Universal Software Radio Peripheral (USRP) [1], a software radio platform that is openly available to many engineers. Moreover, the implementation of the in-car system appears to fully trust all received messages. We found no evidence of basic security practices such as input validation being followed.

Significant communication range. While the vehicle's metal body does shield the signal, we found a larger than expected range. TPMS messages can be correctly received up to 10m from the car with a cheap antenna and up to 40m with a basic low noise amplifier. This means that an adversary can overhear or spoof transmission from roadside or possibly from a following vehicle. Therefore, the transmission powers being used are not low enough to justify the lack of other security measures through a short communication range.

Vehicle tracking. Each in-tire sensor module contains a 32-bit immutable identifier that is contained in every message. This length of the identifier field renders tire sensor module IDs sufficiently unique to be useful for tracking cars. Tracking vehicles is already possible through vision-based automatic license plate identification, or through toll tag or other wireless car components. Still, tracking through TPMS identifiers raises new concerns, because these transmitters are difficult to be deactivated by drivers, because they are available in all new cars, and because tracking with a wireless receiver is low-cost solution compared to the vision technology.

Defenses. We discuss security mechanisms that are applicable to this low-power in-car sensor scenario without taking away the ease of operation when installing a new tire. The mechanisms include relatively straightforward design changes that would significantly mitigate these risks and cryptographic protocols that can be implemented with the required energy budget.

We believe that the insights obtained can also benefit the design of other emerging wireless in-car sensing systems. Modern automobiles typically contain roughly three miles of wire [28], and this will only increase as we strive to make our motor vehicles more intelligent with the aid of an increasing number of on-board electronic components, ranging from navigation systems to entertainment systems to in-car sensors. Increasing the amount of wires directly affects car weight and wire complexity, which decreases fuel economy [13] and imposes difficulties on fault diagnosis [28]. For this reason, wireless technologies are increasingly being used in and around the car to collect control/status data of the car's electronics [16, 30], or to monitor the state of a car's tires. Thus, understanding and addressing the vulnerabilities associated with internal automotive communications, and TPMS in particular, is essential to ensuring that the new wave of intelligent automotive applications will be safely deployed within our cars.

1.3 Outline

We begin in Section 2 by presenting an overview of TPMS and raising related security and privacy concerns. Although the specifics of the TPMS communication protocols are proprietary, we present our reverse-

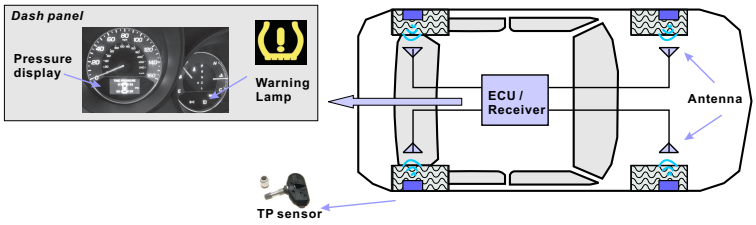


Figure 1: TPMS architecture with four antennas.

engineering effort that reveals the details of the protocols in Section 3. Then, we discuss our study on the susceptibility of TPMS to eavesdropping in Section 4 and message spoofing attacks in Section 5. After completing our security and privacy analysis, we recommend defense mechanisms to secure TPMS in Section 6. Finally, we wrap up our paper by presenting related work in Section 7 and conclusion in Section 8.

2 TPMS Overview and Goals

TPMS Architecture. A typical direct TPMS contains the following components: TPM sensors fitted into the back of the valve stem of each tire, a TPM electric control unit (ECU), a receiving unit (either integrated with the ECU or stand-alone), a dashboard TPM warning light, and one or four antennas connected to the receiving unit. The TPM sensors periodically broadcast the pressure and temperature measurements together with their identifiers. The TPM ECU/receiver receives the packets and performs the following operations before sending messages to the TPM warning light. First, since it can receive packets from sensors belonging to neighboring cars, it filters out those packets. Second, it performs temperature compensation, where it normalizes the pressure readings and evaluates tire pressure changes. The exact design of the system differs among suppliers, particularly in the antenna configuration and the wireless communication protocols. The four-antenna configuration is normally used in high-end car models, whereby each antenna is mounted in each wheel housing behind the wheel arch shell and connected to a receiving unit through high frequency antenna cables, as depicted in Figure 1. The four-antenna system has the advantage of prolonging the battery life of sensors, since the antennas are mounted close to the TPM sensors which reduces the required transmission power of the sensors. However, to reduce the cost of automobiles, the majority of car manufactories choose to use one antenna, whereby the antenna is typically mounted on the rear window [11, 36].

Communication protocols. The communications protocols used between sensors and TPM ECUs are proprietary. From supplier websites and marketing materials, however, one learns that TPMS data transmission commonly uses the 315 MHz or 433 MHz HF bands (UHF) and ASK (Amplitude Shift Keying) or FSK (Frequency Shift Keying) modulation schemes. Each tire pressure sensor carries an identifier (ID). Before the TPMS ECU can accept data reported by tire pressure sensors, IDs of the sensor and the position of the wheel that it is mounted on have to be entered to the TPMS ECU either manually in most cars or automatically in some high-end cars. This is typically done during tire installation. Afterwards, the ID of the sensor becomes the key information that assists the ECU in determining the origin of the data packet and filtering out packets transmitted by other vehicles.

To prolong the battery life, tire pressure sensors are designed to sleep most of the time and wake up in two scenarios: (1) when the car starts to travel at high speeds (over 40 km/h), the sensors are required to monitor tire pressures; (2) during diagnosis and the initial sensor ID binding phases, the sensors are required to transmit their IDs or other information to facilitate the procedures. Thus, the tire pressure sensors will wake up in response to two triggering mechanisms: a speed higher than 40 km/h detected by an on-board accelerometer or an RF activation signal.

The RF activation signals operates at 125 kHz in the low frequency (LF) radio frequency (RF) band and

can only wake up sensors within a short range, due to the generally poor characteristics of RF antennas at that low frequency. According to manuals from different tire sensor manufacturers, the activation signal can be either a tone or a modulated signal. In either case, the LF receiver on the tire sensor filters the incoming activation signal and wakes up the sensor only when a matching signal is recognized. Activation signals are mainly used by car dealers to install and diagnose tire sensors, and are manufacturer-specific.

2.1 Security and Privacy Analysis Goals

Our analysis will concentrate on tracking risks through eavesdropping on sensor identifiers and on message spoofing risks to insert forged data in the vehicle ECU. The presence of an identifier raises the specter of location privacy concerns. If the sensor IDs were captured at certain roadside tracking points and stored in databases, third parties could infer or proof that the driver has visited potentially sensitive locations such as medical clinics, political meetings, or nightclubs. An example of such risks are electronic toll records that are captured at highway entry and exit points but also by private entities for traffic monitoring purposes. In some states, these records are frequently subpoenaed for civil lawsuits. If tracking through the tire pressure monitoring system were possible, this would create additional concerns, particularly because the system will soon be present in all cars and cannot easily be deactivated by a driver. Besides these privacy risks, we will consider attacks where an adversary interferes with the normal operations of TPMS by trying to actively injecting forged messages. For instance, an adversary could attempt to send a low pressure packet to trigger the low pressure warning lights. Alternatively, the adversary could cycle through a few forged low pressure packets and a few normal pressure packets, causing the low pressure warning lights to be on and off. Such attacks, if possible, could undermine drivers faith in the system and potentially lead them to ignore TPMS-related warnings at all. To evaluate the privacy and security risks of such a system requires understanding the following issues:

Difficulty of reverse engineering. Many potential attackers are unlikely to have access to insider information and must therefore reconstruct the protocols, both to be able to extract IDs to track vehicles and to spoof messages. The level of information necessary differs among attacks, replays for example might only require knowledge of the frequency band but more sophisticated spoofing requires protocol details. For spoofing attacks we also consider whether there are available off-the-shelf radios that can generate and transmit packets at the required frequency.

Identifier characteristics. Tracking requires observing identifying characteristics from a message, so that multiple messages can be linked to the same vehicle. The success of tracking is closely tied to the answers of the following questions: (1) Are the sensor IDs used temporarily or used over long time intervals? (2) Does the length of the sensor ID suffice to uniquely identify a car? Since the sensor IDs are meant to primarily identify their positions in the car, it may not be globally unique and may render tracking difficult.

Transmission range and frequency. Tracking further depends on whether a road-side tracking unit will be likely to overhear a transmission from a car passing at high speed. This requires understanding the range and messaging frequency of packet transmissions. To avoid interference between cars and to prolong the battery life, the transmission powers of the sensors are deliberately chosen to be low. Is it possible to track vehicles with such low transmission power combined with low messaging frequency?

Security measures. The ease of message spoofing depends on the use of security measures in TPMS systems. The key questions to make message spoofing a practical threat include: (1) Are messages authenticated? Does the vehicle use consistency checks and filtering mechanisms to reject suspicious packets?

We will address these question in the following sections.

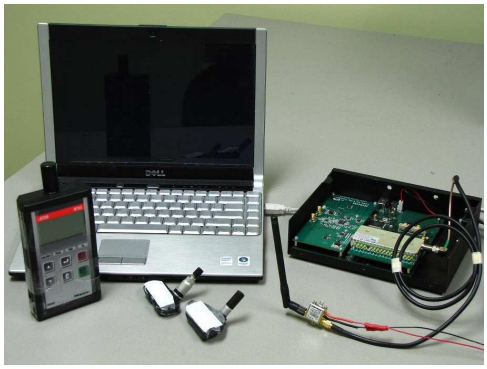


Figure 2: Equipment used for packet sniffing. At the bottom, from left to right are the ATEQ VT55 TPMS trigger tool, two tire pressure sensors (TPS-A and TPS-B), and a low noise amplifier (LNA). At the top is one laptop connected with a USRP with a TVRX daughterboard attached.

3 Reverse Engineering TPMS Communication Protocols

As the first step to analyze the security and privacy risks, a thorough comprehension of the protocols for *specific* sensor systems is necessary. To elaborate, one needs to know the modulation schemes, encoding schemes, and message formats, in addition to the activation and reporting methodologies to properly decode or spoof sensor messages. Apart from access to an insider, this information would most likely require reverse-engineering by an adversary. To convey the level of difficulty of this process for in-car sensor protocols, we provide a brief walk-through of our approach below, and we begin by presenting relevant hardware.

Tire pressure sensor equipment. We selected two representative tire pressure sensors that employ different modulation schemes. Both sensors are used in automobiles with high market share in the US. To prevent misuse of the information here, we refer to these sensors simply as *tire pressure sensor A (TPS-A)* and *tire pressure sensor B (TPS-B)*. To help our process, we also acquired a *TPMS trigger tool*, which is available for a few hundred dollars. Such tools are handheld devices that can activate and decode information from a variety of tire sensors implementations. These tools are commonly used by car technicians and mechanics for troubleshooting. For our experiments, we used a TPMS trigger tool from ATEQ [8] (ATEQ VT55).

Raw signal sniffer. Reverse engineering the TPMS protocols requires the capture and analysis of raw signal data. We selected GNU Radio [2] in conjunction with the Universal Software Radio Peripheral (USRP) [1]. GNU Radio is an open source, free software toolkit that provides a library of signal processing blocks that run on a host processing platform. Algorithms implemented using GNU Radio can receive data directly from the USRP, which is the hardware that provides RF access via an assortment of daughterboards. They include the TVRX daughterboards capable of receiving RF in the range of 50 Mhz to 870 MHz and LFRX daughterboards able to receive between DC to 30 Mhz RF band. For convenience, we also initially used an Agilent 89600 Vector Signal Analyzer for data capture, but such equipment is not necessary. The pressure sensor modules, trigger tool, and software radio platform are shown in Figure 2.

3.1 Reverse Engineering Walk Through

While our public domain search resulted in only high-level knowledge about TPM communication protocol specifics, anticipating sensor activity in the 315/433 MHz bands did provide us with a starting point for our reverse engineering analysis.

We began by collecting a few transmissions from each TPS-A or TPS-B sensor. The VSA was used to narrow down the frequency bandwidth necessary for capturing the full transmission and to estimate the

rough sample rate to avoid aliasing. We placed sensors close to the VSA receiving antenna while using ATEQ VT55 to trigger the transmission. Although initial data collections were obtained using the VSA, the research team switched to using the USRP to illustrate that our findings (and subsequently our attacks) can be achieved with low-cost hardware. An added benefit of using the USRP for the data collections is that it was capable of providing synchronized collects for the LF and HF frequency bands — thus allowing us to extract important timing information between the activation signals and the sensor responses. To perform these collects, the TVRX and LFRX daughterboards were used to provide access to the proper radio frequencies.

Once sensor bursts from TPS-A and TPS-B were collected, we began our signal analysis in Matlab to understand the modulation and encoding schemes, and then moved on to recover bits from the signal. The final step was to map out the message format.

Determine coarse physical layer characteristics. The first phase of characterizing the sensors involved measuring burst widths, bandwidth, and other physical layer properties. We noted that burst widths were on the order of 15 ms. During this initial analysis, we noted that each sensor transmitted multiple bursts in response to their respective activation signals. TPS-A used 4 bursts, while TPS-B responded with 5 bursts. Individual bursts in the series were determined to be exact copies of each other, thus each burst encapsulates a complete sensor report.

Identify the modulation scheme. Analysis of the baseband waveforms revealed two distinct modulation schemes for the sensors. TPS-A employed amplitude shift keying (ASK), while TPS-B employed a hybrid modulation scheme — simultaneous usage of ASK and frequency shift keying (FSK). We speculate that the hybrid scheme is used for two reasons: (1) to maximize operability with TPM readers and (2) to mitigate the effects of an adverse channel during normal operation. Figure 3 illustrates the differences between the sensors' transmission in both the time and frequency domains. The modulation schemes are also observable in these plots.

Resolve the encoding scheme. Despite the different modulation schemes, it was immediately apparent that both sensors were utilizing Manchester encoding (after distinct preamble sequences). The baud rate is directly observable under Manchester encoding and was on the order of 5 kBd. The next step was to determine the bit mappings from the Manchester encoded signal. In order to accomplish this goal, we leveraged knowledge of a known bit sequence in each message. We knew the sensor ID because it was printed on each sensor and assumed that this bit sequence must be contained in the message. We found that applying differential Manchester decoding generated a bit sequence containing the sensor ID.

Reconstructing the message format. While both sensors had used the same encoding scheme, the packet format differs among sensor brands. The next step was to determine the message mappings for the rest of the bits. To understand the size and meaning of each bitfield, we manipulated sensor transmissions by varying a single parameter and observed which bits changed in the message. For instance, we adjusted the temperature using hot guns and refrigerators, or adjusted the pressure. By simultaneously using the ATEQ VT55, we were also able to observe the exact transmitted values and correlate them with our decoded bits. Using this approach, we managed to determine the majority of message fields and their meanings for both TPS-A and TPS-B. These included temperature, pressure, and sensor ID, as illustrated in Figure 4. We also identified the use of a CRC checksum and determined the CRC polynomials through a brute force search.

At this point, we did not yet understand the meaning of a few bits in the message. We were later able to reconstruct these by generating messages with our software radio, changing these bits, and observing the output of the TPMS tool or a real car. It turned out that these were parameters like battery status, over which we had no direct control on the sensor module. More details on message spoofing are presented in Section 5.

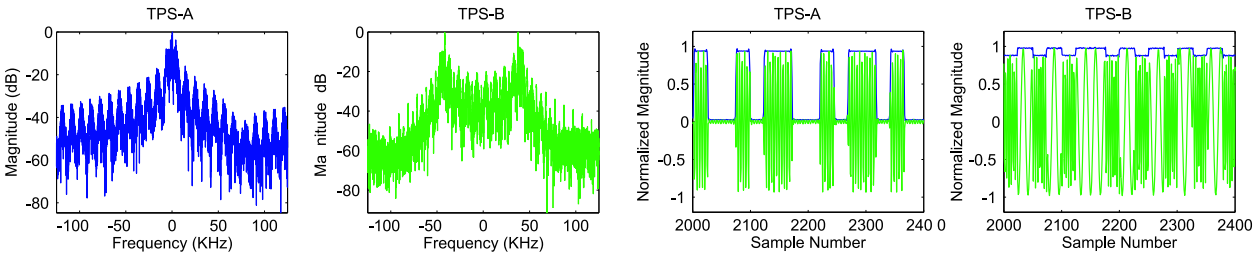


Figure 3: A comparison of FFT and signal strength time series between TSP-A and TSP-B sensors.

Figure 4: An illustration of a packet format. Note the size is not proportional to real packet fields.

3.2 Lessons Learned

The aforementioned reverse-engineering can be accomplished with a reasonable background in communications and computer engineering. It took a few days for a PhD-level engineer experienced with prior protocols and reverse engineering to build an initial system. It took several weeks for an MS-level students with no prior experience in reverse engineering and GNU Radio programming to understand and reproduce the attack. The equipment used (the VTEQ VT55 and USRP attached with TVRX) is openly available and costs $1500 at current market prices.

Perhaps one of the most difficult issues involved baud rate estimation. Since Manchester encoding was being used, our initial baud rate estimates involved averaging the gaps between the transition edges of the signal. However, the jitter (most likely associated with the local oscillators of the sensors) makes it almost impossible to estimate a baud rate accurate enough for simple software-based decoder to work correctly. To address this problem, we modified our decoders to be self-adjustable to compensate for the estimation errors throughout the burst.

The results of reverse engineering reveal the following observations. Firstly, it is evident that encryption has not been used—which makes the system vulnerable to various attacks. Secondly, each message contains a 28-bit or 32-bit sensor ID and the IDs do not change throughout the sensors' lifetimes. Given that there are 254.4 million registered passenger vehicles in United States [31], a 28-bit Sensor ID is enough to uniquely identify each registered car. Even in the future when the number of cars may exceed 256 million, the unique combination of four IDs each associated with one of the four car tires suffices for tracking purpose.

4 Experimental Evaluation of Eavesdropping

A critical question for evaluating privacy implications of in-car wireless networks is whether the transmissions can be easily overheard from outside the vehicle body. While tire pressure data does not require strong confidentiality, the TPMS protocols contain identifiers that can be used to track the locations of a device. In practice, the probability that a transmission can be observed by a stationary receiver depends not only on the communication range but also on the messaging frequency and speed of the vehicle under observation, because these factors affect whether a transmission occurs in communication range.

The transmission power of pressure sensors is relatively small to prolong sensor battery lifetime and reduce cross-interference. Additionally, the NHTSA requires tire pressure sensors to transmit data only once every 60 seconds to 90 seconds. The low transmission power, low data report rate, and high travel speeds of automobiles raise questions about the feasibility of eavesdropping.

In this section, we experimentally evaluate the range of TPMS communications and further evaluate the feasibility of tracking. This range study will use TPS-A sensors, since their TPMS uses a four-antenna

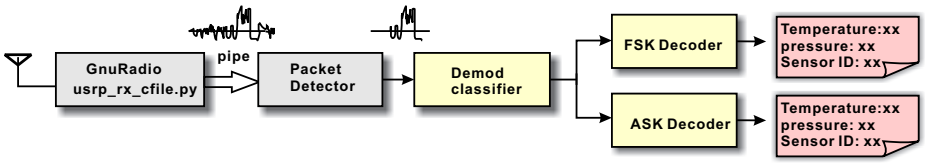


Figure 5: Block chart of the live decoder/eavesdropper.

structure and operates at a lower transmission power. It should therefore be more difficult to overhear.

4.1 Eavesdropping System

During the reverse engineering steps, we developed two Matlab decoders: one for decoding ASK modulated TPS-A and the other for decoding the FSK modulated TPS-B. In order to reuse our decoders yet be able to constantly monitor the channel and only record useful data using GNU radio together with the USRP, we created a live decoder/eavesdropper leveraging pipes. We used the GNU Radio standard python script `usrp_rx_cfile.py` to sample channels at a rate of 250 kHz, where the recorded data was then piped to a packet detector. Once the packet detector identifies high energy in the channel, it extracts the complete packet and passes the corresponding data to the decoder to extract the pressure, temperature, and the sensor ID. If decoding is successful, the sensor ID will be output to the screen and the raw packet signal along with the time stamp will be stored for later analysis. To be able to capture data from multiple different TPMS systems, the eavesdropping system would also need a modulation classifier to recognizes the modulation scheme and choose the corresponding decoder. For example, Liedtke's [26] algorithm could be used to differentiate ASK2 and FSK2. Such an eavesdropping system is depicted in Fig. 5.

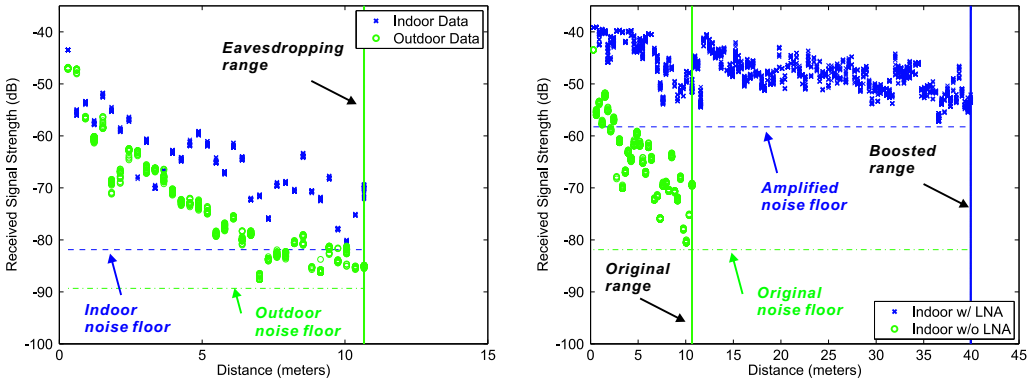
In early experiments, we observed that the decoding script generates much erratic data from interference and artifacts of the dynamic channel environment. To address this problem, we made the script more robust and added a filter to discard erroneous data. This filter will drop all signals that do not match TPS-A or TPS-B.

We have tested our live decoder on the interstate highway I-26 (Columbia, South Carolina) with two cars running in parallel at speeds exceeding 110 km/h.

4.2 Eavesdropping Range

We measured the eavesdropping range in both indoor and outdoor scenarios by having the ATEQ VT55 trigger the sensors. In both scenarios, we fixed the location of the USRP at the origin $(0, 0)$ in Figure 7 and moved the sensor along the y-axis. In the indoor environment, we studied the reception range of stand-alone sensors in a hallway. In the outdoor environment, we drove one of the authors' cars around to measure the reception range of the sensors mounted in its front left wheel while the car's body was parallel to the x-axis, as shown in Figure 7. In our experiment, we noticed that we were able to decode the packets when the received signal strength is larger than the ambient noise floor. The resulting signal strength over the area where packets could be decoded successfully and the ambient noise floors are depicted in Figure 6 (a). The results show that both the outdoor and indoor eavesdropping ranges are roughly 10.7 m, the vehicle body appears only to have a minor attenuation effect with regard to a receiver positioned at the side.

We next performed the same set of range experiments while using a low noise amplifier (LNA), shown in Figure 2, between the antenna and the USRP radio front end. As can be seen from Figure 6, the signal strength of the sensor transmissions still decreased with distance and the noise floor was raised because of the LNA, but the LNA amplified the received signal strength and improved the decoding range from 10.7 meters to 40 meters. This shows that with some inexpensive hardware a significant eavesdropping range can be achieved, a range that allows signals to be easily observed from the roadside.



(a) indoor vs. outdoor (w/o LNA) (b) with LNA vs. without LNA (indoor)

Figure 6: Comparison of eavesdropping range of TPS-A.

Note that other ways to boost receiving range exist. Examples include the use of directional antennas or more sensitive omnidirectional antennas. We refer readers to the antenna studies in [9, 15, 39] for further information.

4.3 Eavesdropping Angle Study

We now investigate whether the car body has a larger attenuation effect if the receiver is located at different angular positions. We also study whether one USRP is enough to sniff packets from all four sensors installed inside all wheels.

The effect of car body. In our first set of experiments, we studied the effect of the car's metallic body on signal attenuation to determine the number of required USRPs. We placed the USRP antenna at the origin of the coordinate, as shown in Figure 7, and position the car at several points on the line of $y = 0.5$ with its body parallel to the x-axis. Eavesdropping at these points revealed that it is very hard to receive packets from four tires simultaneously. A set of received signal strength (RSS) measurements when the front left wheel was located at $(0, 0.5)$ meters are summarized in Table 1. Results show that the USRP can receive packets transmitted by the front left, front right and rear left sensors, but not from the rear right sensor due to the signal degradation caused by the car's metallic body. Thus, to assure receiving packets from all four sensors, at least two observation spots are required with each located on either side of the car. For instance, two USRPs can be placed at different spots, or two antennas connected to the same USRP can be mounted meters away.

The eavesdropping angle at various distances. We studied the range associated with one USRP receiving packets transmitted by the front left wheel. Again, we placed the USRP antenna at the origin and recorded packets when the car moved along trajectories parallel to the x-axis, as shown in Figure 7. These trajectories were 1.5 meters apart. Along each trajectory, we recorded RSS at the locations from where USRP could decode packets. The colored region in Figure 8, therefore, denotes the eavesdropping range,

Location	RSS (dB)	Location	RSS (dB)
Front left	-41.8	Rear left	-55.0
Front right	-54.4	Rear right	N/A

Table 1: RSS when USPR is located 0.5 meters away from the front left wheel.

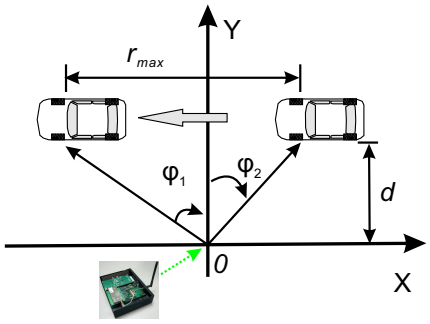


Figure 7: The experiment setup for the range study.

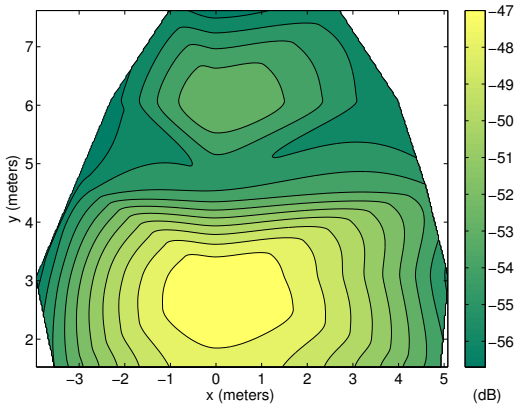


Figure 8: Study the angle of eavesdropping with LNA.

and the contours illustrate the RSS distribution of the received packets.

From Figure 8, we observe that the maximum horizontal eavesdropping range, r_{max}, changes as a function of the distance between the trajectory and the USRP antenna, d. Additionally, the eavesdropping ranges on both sides of the USRP antenna are asymmetric due to the car's metallic body. Without the reflection and impediment of the car body, the USRP is able to receive the packets at further distances when the car is approaching rather than leaving. The numerical results of r_{max}, φ_1, the maximum eavesdropping angle when the car is approaching the USRP, and φ_2, the maximum angle when the car is leaving the USRP, are listed in Figure 9. Since the widest range of 9.1 meters at the parallel trajectory was 3 meters away from the x-axis, an USRP should be placed 2.5 meters away from the lane marks to maximize the chance of packet reception, assuming cars travel 0.5 meter away from lane marks.

Messaging rate. According to the NHTSA regulations, TPMS sensors transmit pressure information every 60 to 90 seconds. Our measurements confirmed that both TPS-A and TPS-B sensors transmit one packet every 60 seconds or so. Interestingly, unlike documented on the manual, e.g., sensors starts to report data periodically after a speed higher than 40 km/h, both sensors periodically transmit packet even when cars are stationary. Furthermore, TPS-B transmits periodic packets even when the car is not ignited.

4.4 Lessons Learned: Feasibility of Tracking Automobiles

The surprising range of 40m makes it possible to capture a packet and its identifiers from the roadside, if the car is stationary (e.g., a traffic light or a parking lot). Given that a TPMS sensor only send one message per minute, tracking becomes difficult at higher speed. Consider, for example, a passive tracking system deployed along the roadside at highway entry and exit ramps. It seeks to extract the unique sensor ID for each car and link entry and exit locations as well as subsequent trips. To ensure capturing at least one packet, a row of sniffers would be required to cover the stretch of road that takes a car 60 seconds to travel. The number of required sniffers, $n_{passive} = ceil(v * T/r_{max})$, where v is the speed of the vehicle, T is the message report period, and r_{max} is the detection range of the sniffer. Using the sniffing system described in previous sections where $r_{max} = 9.1$ m, 110 sniffers are required to guarantee capturing one packet transmitted by a car traveling at 60 km/h. Deploying such a tracking system appears cost-prohibitive.

It is possible to track with fewer sniffers, however, by leveraging the activation signal. The tracking station can send the 125kHz activation signal to trigger a transmission by the sensor. To achieve this, the triggers and sniffers should be deployed in a way such that they meet the following requirements regardless of the cars' travel speeds: (1) the transmission range of the trigger should be large enough so that the passing car is able to receive the complete activation signal; (2) the sniffer should be placed at a distance from the activation sender so that the car is in the sniffers' eavesdropping range when it starts to transmit; and (3) the

d (m)	φ_1 (°)	φ_2(°)	r_{max} (m)
1.5	72.8	66.8	8.5
3.0	59.1	52.4	9.1
4.5	45.3	31.8	7.5
6.0	33.1	20.7	6.3
7.5	19.6	7.7	3.8

Figure 9: The eavesdropping angles and ranges when the car is traveling at various trajectories.

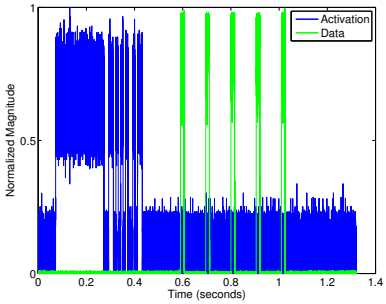


Figure 10: Time series of activation and data signals.

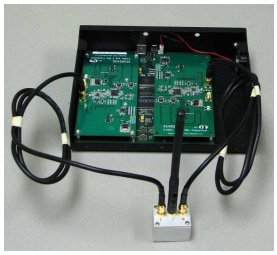


Figure 11: Frequency mixer and USRP with two daughterboards are used to transmit data packets at 315/433 MHz.

car should stay within the eavesdropping range before it finishes the transmission.

To determine the configuration of the sniffers and the triggers, we conducted an epitomical study using a USRP with two daughterboards attached, one recording at 125 kHz and the other recording at 315 MHz. Our results are depicted in Figure 10 and show that the activation signal of TPS-B takes approximately 359 ms to transmit. The sensors start to transmit 530 ms after the beginning of the activation signal, and the data takes 14 ms to transmit. This means, that to trigger a car traveling at 60 km/h, the trigger should have a transmission range of at least 6 meters. Since a sniffer can eavesdrop up to 9.1 meters, it suffices to place the sniffer right next to the trigger. Additional sniffers could be placed down the road to capture packets of cars traveling at higher speeds.

To determine feasibility of this approach, we have conducted a roadside experiment using the ATEQ VT55 which has a transmission range of 0.5 meters. We were able to activate and extract the ID of a targeted TPMS sensor moving at the speed of 35 km/h using one sniffer. We note that ATEQ VT55 was deliberately designed with short transmission range to avoid activating multiple cars in the dealership. With a different radio frontend, such as using a matching antenna, one can increase the transmission range of the trigger easily and enable capturing packets from cars at higher speeds.

5 Feasibility of Packet Spoofing

The ability to insert forged data into safety-critical in-vehicle system could present potentially even greater risks than the tracking risks discussed so far. While the TPMS is not yet a highly safety-critical system, we experiment with spoofing attacks to understand (1) whether the receiver sensitivity of an in-car radio is high enough to allow spoofing messages from outside the vehicle or neighboring vehicle, and (2) state-of-the-art security mechanisms and practices in such systems. In particular, we are curious whether the system uses authentication, input validation, or filtering mechanisms to reject suspicious packets.

The packet spoofing system. Our packet spoofing system takes input including the *sensor type* and the following sensor parameters: *sensor ID*, *temperature*, *pressure*, and *status flags*, and generates a properly formulated message. It then modulates the message at baseband (using ASK or FSK) while inserting the proper preamble. Finally, the rogue sensor packets are upconverted and transmitted (either continuously or just once) at the desired frequency (315/433 MHz).

Since we found no USRP daughterboard available for purchase (at the time of this writing) that can transmit in the 315MHz frequency band, we used a frequency mixing approach. To implement this, we leveraged two XCVR2450 daughterboards and a frequency mixer (mini-circuits ZLW11H) as depicted in Fig.11. By transmitting a tone out of one XCVR2450 into the LO port of the mixer, we were able to mix down the spoofed packet from the other XCVR2450 to the appropriate frequency. For 315 MHz, we used a tone at 5.0 GHz and the spoofed packet at 5.315 GHz.[1]

[1]For 433 MHz, the spoofed packet can be transmitted at 5.433 GHz. We have also successfully conducted the experiment using

To validate our system, we decoded spoofed packets with the TPMS trigger tool. Figure 12 shows a screen snapshot of the ATEQ VT55 after receiving a spoofed packet with a sensor ID of "DEADBEEF" and a tire pressure of 0 PSI. This testing also allowed us to understand the meaning of a some remaining status flags in the protocol.

5.1 Exploring Vehicle Security

We next used this setup to send various forged packets to a car using TPS-A sensors (belonging to one of the authors). We made the following observations.

No Authentication. The vehicle ECU ignores packets with a sensor ID that does not match one of the known IDs of its tires, but appears to accept all other packets. For example, we transmitted a forged packet with the ID of the left front tire and a pressure of 0 PSI and found 0 PSI immediately reflected on the dashboard tire pressure display. By transmitting a message with the alert bit set we were able to immediately illuminated the low-pressure warning light[2], and with about 2 seconds delay the vehicle's general-information warning light, as shown in Figure 13.

No filtering or input validation. We forget packets at a rate of 40 packets per second. Neither this increased rate, nor the occasional different reports by the real tire pressure sensor seemed to raise any suspicion in the ECU or any alert that something is wrong. The dashboard simply displayed the spoofed tire pressure. We next transmitted two packets with very different pressure values alternately at a rate of 40 packets per second. The dashboard display appeared to randomly alternate between these values. Similarly, when alternating between packet with and without the alert flat, we observed the warning lights switched on and off at non-deterministic time intervals. Occasionally, the display seemed to freeze on one value. While the tire pressure alert usually disappeared about 10 seconds after stopping our messages, we were once unable to reset it even by turning off the ignition. It did, however, reset after about 10min of driving.

Interestingly, the illumination of the low-pressure warning light depends only on the alert bit—the light turns on even if the rest of the message reports a normal tire pressure of 32 PSI! This further illustrates that the ECU does not appear to use any filtering or input validation.

5.2 Range Analysis of Packet Spoofing

We investigated the effectiveness of the attack when packets are injected packets at different angles and vehicles are traveled at different speeds.

Range and angle of attack. Similar to the decoding range shown previously in Figure 7, we measured the operating range of the packet spoofing attack by adjusting ϕ_2, e.g., the angle between the y-axis and the car body when the car is approaching the packet spoofing system. We observed a packet spoofing range of 14.1, 15.2, and 16.8 meters when ϕ_2 equals 0°, 45°, and 90°, respectively. For the purpose of proving the concept, we note that we only used low-cost antennas and radio devices in our experiments. We believe that the range of packet spoofing can be greatly expanded by applying amplifiers, high-gain antennas, or antenna arrays.

Feasibility of Inter-Vehicle Spoofing. We deployed the attacks against willing participants on inter-state highway I-26 to determine if they are viable at high speeds. Two cars owned by the authors were involved in the experiment, whereby the victim car has TPS-A sensors installed and the attacker's car was equipped with our packet spoofing system. Throughout our experiment, we transmitted alert packets using the front-left-tire ID of the target car, while the victim car was traveling to the right of the attacker's car.

two RFX-1800 daughterboards, whose operational frequencies are from 1.5 GHz to 2.1 GHz.

[2]To discover this bit we had to deflate one tire and observe the tire pressure sensors response. Simply setting a low pressure bit or reporting low pressure values did not trigger any alert in the vehicle.

Figure 12: The TPMS trigger tool displays the spoofed packet with the sensor ID "DEADBEEF". We crossed out the brand of TP sensors to avoid legal issues.

(a) (b)

Figure 13: Dash panel snapshots: (a) the tire pressure of left front tire displayed as 0 PSI and the low tire pressure warning light was illuminated immediately after sending a spoofed alert packet with 0 PSI; (b) the car computer turned on the general warning light around 2 seconds after keeping sending spoofed packets.

We observed that the attacker was able to trigger both the low-pressure warning light and the car's central-warning light on the victim's car when traveling at 55 km/h and 110 km/h, respectively. Additionally, the low-pressure-warning light illuminated immediately after the attacker enter the packet spoofing range.

5.3 Lessons Learned

The successful implementation of a series of spoofing attacks reveals that ECU relies on sensor IDs to filter packets, and no filter mechanisms are implemented to reject packets with conflicting information or abnormal packets transmit at extremely high rate. The absence of filter and authentication mechanisms opens many loopholes for adversaries to explore for their 'creative' attacks.

Furthermore, despite the unavailability of a radio frontend that can transmit at 315/433 Mhz, we managed to launch the spoofing attack using a frequency mixer. This result is both encouraging and alarming since it shows that an adversary can spoof packets even without easy access to transceivers that operate at the target frequency band.

6 Protecting TPMS Systems from Attacks

There are several steps that could be taken to improve the dependability and security of TPMS systems. In particular, some of the problems that we identified arise from poor system design, while other issues are tied to the lack of proper cryptographic mechanisms to secure the communications.

6.1 Reliable Software Design

The first recommendation that we would make is that software running on TPMS systems should follow basic reliable software design practices. In particular, one observation that we had during our investigation was that it was possible to convince the TPMS control unit to display readings that were clearly not possible. For example, the TPMS packet format included a field for tire pressure as well as a separate field for warning flags related to tire pressure. Unfortunately, the relationship between these fields were not checked by the TPMS control unit when processing communications from the sensors. As noted earlier, we were able to send a packet containing a legitimate tire pressure value while also containing a low tire pressure warning flag. The result was that the driver's display indicated that the tire had low pressure even though its pressure was normal. A straight forward fix for this problem (and other similar problems) would be to update the software on the TPMS control unit to perform consistency checks between the values in the data fields and the warning flags. Similarly, when launching message spoofing attacks, we continuously transmitted spoofed packets to 'over-power' the periodically transmitted legitimate packets so that the control unit neglected the legitimated packets. The control unit could have employed some detection mechanism

to, at least, raise an alarm when the incoming packet rate exceeds the expected one, or have enforced some simple signal processing operations to filter out continuous transmissions.

Another rule for secure system design is to restrict the supported system operations to their minimum. For instance, a sensor responses to activation signals regardless of its locations, which makes active tracking possible at much lower cost than passive tracking. Since the activation signal was designed to wake up sensors for regular measurements or diagnosis at dealerships, a quick fix to prevent active tracking is to prohibit sensors from responding to activations when it is at a speed higher than 40 km/h.

6.2 Improving Data Packet Format

One fundamental reason that eavesdropping and spoofing attacks are feasible in TPMS systems is that packets are transmitted in plaintext. To prevent these attacks, a first line of defense is to encrypt TPM packets. The basic packet format in a TPMS system included a sensor ID field, fields for temperature and tire pressure, fields for various warning flags, and a checksum. Clearly, a first step to ensuring that eavesdropping is not possible would be to encrypt the packets before they are transmitted. Unfortunately, the current packet format used is ill-suited for proper encryption, and the packet format should be changed. We note that naively encrypting the current packet format would still support dictionary-based cryptanalysis, as well as replay attacks against the system. For this reason, we recommend that an additional sequence number field be added to the packet to ensure freshness of a packet. Further, requiring that the sequence number field be incremented during each transmission would ensure that subsequent encrypted packets from the same source become indistinguishable, thereby making eavesdropping and cryptanalysis significantly harder. Lastly, although the ideal position for CRC checksumming is outside of the encrypted payload, we recommend that an additional cryptographic checksum (e.g. a message authentication code) be placed prior to the CRC checksum if a Vernam-style encipherment (e.g. a block cipher in CFB/OFB/CTR mode) is being performed.

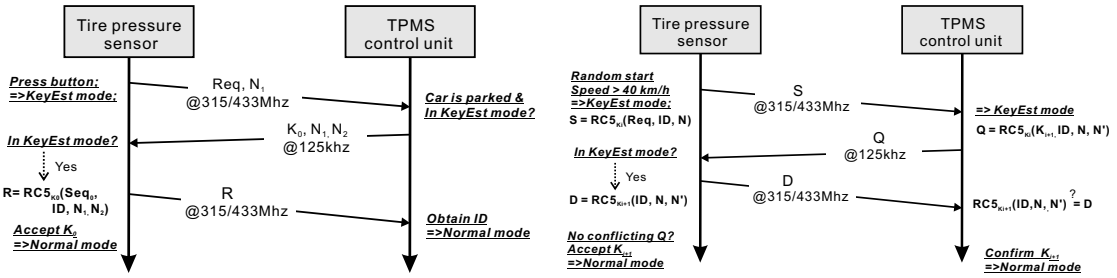
6.3 Maintaining Confidentiality

Beyond requiring that the sensors and the control unit be able to perform encryption, we must also maintain the keys shared between the TPMS control unit and the TPMS sensors. Key management schemes for TPMS should satisfy the following basic requirements: (1) The scheme should be efficient in terms of computation and communication requirements. For this reason, public key schemes like Diffie-Hellman should be avoided. (2) The key establishment protocol should not be initiated easily in a public location. Cars are routinely left in parking lots unattended and thus it is desirable to ensure that no adversary can tamper or change keys purely from outside the car. (3) The key establishment procedure should be easy to perform as tires may need to be replaced and a complicated procedure would increase the difficulty of tire maintenance. (4) Instead of using one key throughout the lifetime of a tire, the key should be updated to ensure freshness.

To meet these requirements, we present a key establishment protocol involving two steps: (1) establishing an initial key K_0 between a sensor and the control unit (Section 6.3.1), and (2) updating the current key K_i to K_{i+1} as needed (Section 6.3.2).

6.3.1 Establish the Initial Key, K_0

We expect that new tires are installed either in the factory, dealership or properly certified garage and hence, we assume these are trustworthy locations (we will discuss untrustworthy dealerships shortly). Hence, for the sake of initialization, we can allow the initial key K_0 to be transmitted in plaintext using short-range wireless communication. Tire pressure sensors can receive activation signals in the 125kHz frequency band,



(a) Initial Key Establishment (b) Key Update

Figure 14: The key establishment and key update protocol between a tire pressure sensor and the TPMS control unit.

where the signals have limited range due to the generally poor characteristics of RF antennas at that low frequency. Thus, we can leverage the activation signals from the control unit to transmit K_0 at 125 kHz. The step-by-step key establishment procedure is illustrated in Figure 14 (a).

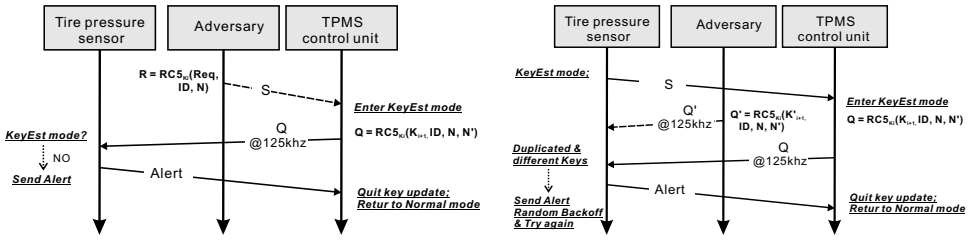
We envision that the key initialization procedure starts when a mechanic manually sets sensors and the car computer into a *key establishment* mode. For example, this may occur by pressing a key-initialization button on the sensor and adjust the car computer accordingly. Responding to the button pressing, the sensor sends a key initialization request (Req, N_1) at the 315/433 Mhz band, where N_i is a random nonce used to indicate the freshness of the request. Once the control unit receives the request, it will send out a randomly generated key, K_0, at 125 kHz if it is in parking status and in key establishment mode. The sensor receives and accepts K_0 only if it is in the key establishment mode. Additionally, it confirms K_0 by sending a message encrypted with K_0 using an appropriate cipher (for the rest of this paper we use RC5 for the block cipher) at 315/433 Mhz, e.g., $RC5_{K_0}(Seq, ID, N_1, N_2)$, and enters the *normal* mode. Once receiving the confirmation message, the control unit records the sensor ID and enters *normal* mode. Thereafter, both entities start to communicate using K_0.

Security Analysis. First, using the 125 kHz frequency band to transmit K_0 makes eavesdropping from a distance extremely difficult. Second, requiring that a button on the sensor be pressed as well as initializing key establishment mode on the car's computer effectively makes it difficult for an adversary to trigger the key establishment procedure. Hence, when a car is left in a public lot unattended, altering and acquiring keys becomes difficult. We acknowledge that, in unlikely cases when an adversary breaks into the car to adjust the car mode and removes rubber layer of tires to activate the button on the sensors, he/she can change and obtain the new shared key. However, we take the viewpoint such a case is best dealt with by a car's anti-theft system. Finally, we have chosen RC5 as the cipher because of its implementation simplicity and low computational costs.

6.3.2 Update the Shared Secret

Updating the shared secret occasionally reduces the risk of an adversary obtaining a key through cryptanalysis. A detailed key update protocol is depicted in Figure 14 (b). At random times when the sensor travels at a speed faster than 40 km/h, it will initiate the key update procedure and transmit a request with a nonce N and its ID. Upon receiving the request, the control unit enters the key establishment mode and transmits the new key K_{i+1} using the short-range 125 kHz band, with the message format as $RC5_{K_i}(K_{i+1}, ID, N, N')$. Upon receiving the new key K_{i+1}, the sensor acknowledges the K_{i+1} by sending a message encrypted with K_{i+1}, e.g., $RC5_{K_{i+1}}(ID, N, N')$. After this message exchange, both the sensor and the control unit would wait for a brief time before they return to the normal mode and use the new key K_{i+1}.

Security Analysis One threat that may undermine key update protocols between two parties, Alice and Bob, is an impersonation attack, whereby an adversary pretends to be Alice and triggers Bob to update the



(a) Impersonating the sensor (b)Impersonating the control unit

Figure 15: An illustration of key-update protocol in malicious scenarios.

key, causing Alice and Bob to no longer agree on the same key. Our key update protocol is immune to such attacks as keys can be updated only if both the sensor and the control unit are active. In particular, we let the sensor initiate the procedure only after it travels at a high speed. As such, any response to the adversary will be heard by the other entity and trigger an "alert" message to end the key update.

Figure 15 illustrates two adversarial scenarios: impersonating the sensor and impersonating the control unit. When an adversary tries to trigger the control unit to update its key (by transmitting a request message), it makes the control unit transmit a key-update message at 125 kHz. A proper implementation of the sensor would then require that receiving a key-update message without having initiated the request would necessitate that the sensor sends an alert message, causing the control unit to quit the key update. Similarly, after a sensor initiates the key update procedure, the adversary may transmit a new key message shortly thereafter. However, the control unit would still transmit a new key in response to the key-update request, and thus a proper implementation would have the sensor generate an alert message when it receive two different key-update messages.

7 Related Work

As wireless technologies have been applied to a wide range of applications, many wireless devices have emerged and become an inseparable part of our social fabric. As such, much effort has been dedicated to analyze the privacy and security issues associated with them. Devices being studied include RFID systems [24, 27, 38], mass-market UbiComp devices [35], household robots [14], and implantable medical devices [21]. Although our work falls in the same category and complements those work, TPMS in automobiles exhibits distinctive features with regard to the radio propagation environment (strong reflection off metal car bodies), the ease of access by adversaries (left unattended in public), the span of usage, the tight linkage to the owners, etc. All these characteristics warrant further studies on the security and privacy of TPMS.

Location privacy in wireless networks has attracted much attention since wireless devices are known to present tracking risks through explicit identifiers in protocols or identifiable patterns in waveforms. In the area of WLAN, Brik *et al.* have shown the possibility to identify users by monitoring radiometric signatures [10]. Gruteser *et al.* [19] demonstrated that one can identify a user's location through link- and application-layer information. A common countermeasure against breaching location privacy is to frequently dispose user identity. For instance, Jiang *et al.* [23] proposed a pseudonym scheme where users change MAC address each session. Similarly, Grenstein *et al.* [18] have suggested an identifier-free mechanism to protect user identities, whereby users can change addresses for each packet.

In the area of centralized cellular systems, Lee *et al.* have shown that the location information of roaming users can be released to third parties [25], and proposed to use the temporary mobile subscriber identifier to cope with the location privacy concern. IPv6 also has privacy concerns caused by the fixed portion of the

address [29]. Therefore, the use of periodically varying pseudo-random addresses has been recommended.

The use of pseudonym is not sufficient to prevent automobile tracking since the sensors report similar tire pressure and temperature which can be used to build a signature of the car. Furthermore, pseudonym cannot defend against the packet spoofing attacks.

Security and privacy in wireless sensor networks have been studied extensively. Perrig *et al.* [34] have proposed a suite of security protocols to provide data confidentiality and authentication for resource-constrained sensors. Random key predistribution schemes [12] have been proposed to establish pairwise keys between sensors on demand. Those key management schemes cannot work well with TPMS, since sensor networks are concerned with establishing keys among a large number of sensors while the TPMS focuses on establishing keys between four sensors and the ECU only.

8 Concluding Remarks

Tire Pressure Monitoring Systems (TPMS) are the first in-car wireless networks that are integrated into all new cars in the US and will soon be deployed in the EU. This paper has evaluating the privacy and security implications of such TPMS, by experimentally evaluating two representative tire pressure monitoring systems.

Our study revealed several concerns. First, we reverse engineered the protocols using the GNU Radio in conjunction with the Universal Software Radio Peripheral (USRP) and found that (i) the TPMS does not employ any cryptographic mechanisms and (ii) transmits a fixed 32bit sensor ID in each packet, which raises the possibility of tracking vehicles through these identifiers. Sensor transmissions can be triggered from roadside stations through an activation signal. We further found out that neither the heavy shielding from the metallic car body nor the low-power transmission has reduced the range of eavesdropping sufficiently to reduce the eavesdropping concerns. In fact, the TP packets can be intercepted up to 40 meters from a passing car using the GNU Radio platform with a basic low-noise amplifier. We note that the eavesdropping range could be further increased with directional antennas, for example.

We also found that current implementations do not appear to follow basic security practices. Messages are not authenticated, we were able to inject spoofed messages and illuminated the low tire pressure warning lights on a car traveling at highway speeds from another nearby car. The vehicle ECU also does not appear to use input validation or message filtering mechanisms, it relies on information from packets that might be inconsistent.

Finally, we have recommended security mechanisms that can alleviate the security and privacy concerns presented without unduly complicating the installation of new tires. The recommendations include standard reliable software design practices, cryptographic protocols that can provide data confidentiality, and a key management protocol customized for TPMS. We believe that our analysis and recommendations on TPMS can provide guidance towards designing more secure in-car wireless networks.

References

[1] Ettus Research LLC. http://www.ettus.com/.

[2] GNU radio. http://gnuradio.org.

[3] IEEE 1609: Family of Standards for Wireless Access in Vehicular Environments (WAVE). http://www.standards.its.dot.gov/fact_sheet.asp?f=80.

[4] IEEE P802.11p: IEEE draft standard for information technology -telecommunications and information exchange between systems. http://www.ieee802.org/11/.

[5] Portable, solar-powered tag readers could improve traffic management. Available at http://news.rpi.edu/update.do?artcenterkey=1828.

[6] Traffic hackers hit red light. Available at http://www.wired.com/science/discoveries/news/2005/08/68507.

[7] Improving the safety and environmental performance of vehicles. *EUROPA-Press Releases*, 23rd May 2008.

[8] ATEQ VT55. http://www.tpms-tool.com/tpms-tool-ateqvt55.php.

[9] C. Balanis and P. Ioannides. Introduction to smart antennas. *Synthesis Lectures on Antennas*, 2(1):1–175, 2007.

[10] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 116–127. ACM, 2008.

[11] M. Brzeska and B. Chakam. RF modelling and characterization of a tyre pressure monitoring system. In *EuCAP 2007: The Second European Conference on Antennas and Propagation*, pages 1 – 6, 2007.

[12] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 197, Washington, DC, USA, 2003. IEEE Computer Society.

[13] G. Cole and A. Sherman. Lightweight materials for automotive applications. *Materials Characterization*, 35:3–9, 1995.

[14] T. Denning, C. Matuszek, K. Koscher, J. R. Smith, and T. Kohno. A spotlight on security and privacy risks with future household robots: attacks and lessons. In *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*, pages 105–114, New York, NY, USA, 2009. ACM.

[15] A. Feresidis and J. Vardaxoglou. High gain planar antenna using optimised partially reflective surfaces. In *IEEE Proceedings on Microwaves, Antennas and Propagation*, volume 148, pages 345 – 350, 2001.

[16] L. Fredriksson and K. AB. Bluetooth in automotive applications. *http://www.kvaser.com/can/info/files/bluetooth-in-automotive-appl.pdf*.

[17] S. Govindjee. Firestone tire failure analysis, 2001.

[18] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceeding of Mobile systems, applications, and services (MobiSys)*, pages 40–53, New York, NY, USA, 2008. ACM.

[19] M. Gruteser and D. Grunwald. A methodological assessment of location privacy risks in wireless hotspot networks. In *Security in Pervasive Computing, First International Conference*, pages 10–24, 2003.

[20] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. *ACM Mobile Networks and Applications (MONET)*, 10(3):315–325, 2005.

[21] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 129–142, Washington, DC, USA, 2008. IEEE Computer Society.

[22] Italy. *http://aglobalworld.com/international-countries/Europe/Italy.php*.

[23] T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless lans. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 246–257, New York, NY, USA, 2007. ACM.

[24] K. Koscher, A. Juels, V. Brajkovic, and T. Kohno. EPC RFID tag security weaknesses and defenses: passport cards, enhanced drivers licenses, and beyond. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 33–42, 2009.

[25] C.-H. Lee, M.-S. Hwang, and W.-P. Yang. Enhanced privacy and authentication for the global system for mobile communications. *Wireless Networks*, 5(4):231–243, 1999.

[26] F. Liedtke. Computer simulation of an automatic classification procedure for digitally modulated communication signals with unknown parameters. *Signal Processing*, 6:311–323, 1984.

[27] D. Molnar and D. Wagner. Privacy and security in library RFID: issues, practices, and architectures. In *Proceedings of Computer and communications security*, pages 210–219, New York, NY, USA, 2004. ACM Press.

[28] N. Murphy. A short trip on the can bus. *Embedded System Programming*, 2003.

[29] T. Narten and R. Draves. RFC 3041 - privacy extensions for stateless address autoconfiguration in IPv6, Jan 2001.

[30] R. Nusser and R. Pelz. Bluetooth-based wireless connectivity in an automotive environment. *Vehicular Technology Conference*, 4:1935 – 1942, 2000.

[31] B. of Transportation. Number of vehicles and vehicle classification, 2007.

[32] D. of Transportation National Highway and T. S. Administration. 49 cfr parts 571 and 585 federal motor vehicle safety standards; tire pressure monitoring systems; controls and displays; final rule. *http://www.tireindustry.org/pdf/TPMS_FinalRule_v3.pdf*.

[33] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure Vehicular Communication Systems: Design and Architecture. *IEEE Communcations Magazine*, 46(11):100–109, November 2008.

[34] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. Spins: security protocols for sensor networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 189–199, New York, NY, USA, 2001. ACM.

[35] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: privacy trends in consumer ubiquitous computing. In *Proceedings of USENIX Security Symposium*, pages 1–16, Berkeley, CA, USA, 2007. USENIX Association.

[36] H. Song, J. Colburn, H. Hsu, and R. Wiese. Development of reduced order model for modeling performance of tire pressure monitoring system. In *IEEE 64th Vehicular Technology Conference*, pages 1 – 5, 2006.

[37] S. Velupillai and L. Guvenc. Tire pressure monitoring. *IEEE Control Systems Magazine*, 27:22–25, 2007.

[38] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212, 2004.

[39] P. Yeh, W. Stark, and S. Zummo. Performance analysis of wireless networks with directional antennas. *IEEE Transactions on Vehicular Technology*, 57(5):3187–3199, 2008.