3-22-2019

Preserving Privacy in Automotive Tire Pressure Monitoring Systems

Kenneth L. Hacker

PRESERVING PRIVACY IN AUTOMOTIVE TIRE PRESSURE MONITORING SYSTEMS

THESIS

Kenneth L. Hacker, 2d Lt, USAF

AFIT-ENG-MS-19-M-031

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-MS-19-M-031

PRESERVING PRIVACY IN AUTOMOTIVE TIRE PRESSURE MONITORING SYSTEMS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Kenneth L. Hacker, BSEE

2d Lt, USAF

March 2019

AFIT-ENG-MS-19-M-031

PRESERVING PRIVACY IN AUTOMOTIVE TIRE PRESSURE MONITORING SYSTEMS

THESIS

Kenneth L. Hacker, BSEE
2d Lt, USAF

Committee Membership:

Scott R. Graham, Ph.D.
Chair

Lt Col Patrick J. Sweeney, Ph.D.
Member

Stephen J. Dunlap, M.S.
Member

AFIT-ENG-MS-19-M-031

Abstract

The automotive industry is moving towards a more connected ecosystem, with connectivity achieved through multiple wireless systems. However, in the pursuit of these technological advances and to quickly satisfy requirements imposed on manufacturers, the security of these systems is often an afterthought. It has been shown that systems in a standard new automobile that one would not expect to be vulnerable can be exploited for a variety of harmful effects.

This thesis considers a seemingly benign, but government mandated, safety feature of modern vehicles; the Tire Pressure Monitoring System (TPMS). Typical implementations have no security-oriented features, leaking data that can be used for reliable tracking by a determined attacker, and being completely open to spoofing attacks.

This research investigates potential privacy concerns of TPMS, first by demonstrating the feasibility of both identifying vehicles and reconstructing their routes without prohibitive cost or expertise. Then, an ID obfuscating scheme is proposed, called TPMS Obfuscation through Rolling ID (TORI), to mitigate these privacy threats while remaining true to the design requirements of TPMS. Various conditions are tested using a modified traffic simulator, which validate the ability to reconstruct the identities of vehicles even from sparse detections.

Acknowledgements

Dark clouds bring waters, when the bright bring none.

-John Bunyan, The Pilgrim's Progress

I would like to thank AFIT and the USAF for an opportunity such as this. Many contributed to my success in this pursuit. Faculty such as Dr. Scott Graham provided countless instances of personal and professional guidance, and exemplified many virtues in their conduct. The students that accompanied me on this journey provided support and companionship that allowed us to lighten each others' burdens and form lasting bonds. Lastly, I am grateful to my family, who have always supported my goals, and enabled me to reach where I am today.

Kenneth L. Hacker

Table of Contents

List of Figures

Figure Page

List of Tables

List of Abbreviations

ANPR	Automated Number Plate Reader
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
DRBG	Deterministic Random Bit Generator
ECU	Electronic Control Unit
FCC	Federal Communications Commission
IoT	Internet of Things
MAC	Message Authentication Code
NHTSA	National Highway Traffic Safety Administration
OEM	Original Equipment Manufacturer
PRNG	Pseudo-Random Number Generator
RANSAC	Random Sample Concensus
SDR	Software Defined Radio
SUMO	Simulator for Urban MObility
TORI	Tire Obfuscation through Rolling ID
TPMS	Tire Pressure Monitoring System
TREAD	Transportation Recall Enhancement, Accountability and Documentation
VANET	Vehicular Ad-Hoc Network

PRESERVING PRIVACY IN AUTOMOTIVE
TIRE PRESSURE MONITORING SYSTEMS

I. Introduction

1.1 Motivation

For over ten years, Tire Pressure Monitoring Systems(TPMS) have been a required feature on nearly all consumer vehicles. TPMS has been very helpful in terms of safety and convenience, warning drivers when a tire is outside of a safe range or has a slow leak. However, security features are absent from current systems, allowing any observer with a radio and basic signal processing knowledge to pick up the packets broadcast by the sensors. The packets themselves hold little inherent value; even if an attacker were given thousands of them, it would reveal little about a vehicle or its driver. However, by aggregating these packets and their timestamped location data, this information can become collectively valuable. The ability to accomplish such an attack is just one symptom of a system without any security measures. The integrity and availability of the system can also be compromised with basic equipment, which can be leveraged for physical effects, such as convincing a driver to pull the car over to check the tire. While the impacts of some of these attacks may seem minor, they can presently be used in a larger scheme, and should not be ignored given the ubiquity of these systems and potential safety implications. This research focuses primarily on the privacy concerns in TPMS, but the impact of the solution presented here called TPMS Obfuscation through Rolling ID (TORI), will be evaluated in light of attacks on the integrity or availability of the system as well.

1.2 Problem Statement

This research examines the security concerns of Tire Pressure Monitoring Systems. Because of the lack of security features, and in the absence of sufficient research into potential exploits, there are several avenues of investigation that could be pursued. This document focuses on privacy concerns that result from the lack of obfuscation leading to easily obtainable tire data, specifically the unique ID associated with a given tire. It experimentally shows the feasibility of identifying and tracking vehicles using TPMS data, and investigates a solution to greatly increase the difficulty of attacking these systems while remaining within the constraints of the operating environment. Additionally, the question of how much information is required to track (or alternatively, what information needs to be obfuscated) is explored to examine the impact of potential security features. Transportation is one of the United States' 16 critical infrastructure sectors [1], and with four of these devices on most consumer vehicles, the sheer number of these devices on the road presents a significant attack surface. A widespread exploit could have dire consequences.

Attack Scenarios.

The feasibility of tracking using Tire Pressure Monitoring Systems could present opportunities for attackers to gain valuable information about drivers. Detectors are relatively small and concealable, so their presence is unlikely to be noticed. Consider the following scenarios, gradually increasing in impact, depending on the determination of an eavesdropper:

- Binary Detection - An adversary places the minimum number of detectors necessary to ensure that they can determine whether or not a target is in a specific location. This could be one detector at the end of a driveway, or perhaps at every entrance and exit to a suburban neighborhood to cast a wider net. It is very reasonable to then leverage this information to establish pattern-of-life data about the group of victims and enable general burglary or targeted crime.

- Targeted Tracking - An adversary places detectors at a controlled entry point for a location of interest, for example the parking area for a technology firm. This detector could be offline, and picked up later to harvest the gathered data. After receiving the desired amount of data, the attacker then drives through residential areas where employees are likely to live with a mobile detection unit. If a tire ID match occurs, an employee is identified, and that address is flagged as being associated with the location of interest. This information could be used for later corporate espionage or for a type of spearphishing attack.
- Full-Scale Tracking - Depending on the resources available, complete tracking of a large number of vehicles could be accomplished by deploying detectors at intersections. Results presented in Chapter 4 demonstrate that monitoring as little as 10% of intersections can lead to reliable vehicle identification, and increasing coverage can lead to higher tracking accuracy between observations. This could be used with good intentions, such as keeping a database of tire IDs and monitoring major highways to find vehicles used in crimes. It could also enable the abuse of such information, as can be imagined when you know the whereabouts of nearly every vehicle in a designated area.

1.3 Research Questions

The goal of this research is to test the feasibility of using TPMS data to identify and track vehicles. Within that question, the research tested how traffic and sensor conditions affect the results. Additionally, the research explored what solutions are effective and feasible to combat attacks that have been demonstrated or hypothesized.

The following supporting questions guided this research.

- What are the safety implications of TPMS vulnerabilities?
- What risks to privacy does TPMS present?
    - Can TPMS data be used to track vehicles?
- Can the privacy and integrity vulnerabilities of TPMS be quantified?
    - What evaluation metrics are appropriate?
- What is the effectiveness and cost of TORI?

1.4 Organization

Chapter II provides the background necessary to establish the problem and develop an experiment to test the research goals. Chapter III describes the methodology and design decisions that are used to test the association and tracking algorithms. The simulation conditions and data flow are presented, along with the details of the programs used to track vehicles. Chapter IV analyzes the results of the experiments. It explores the trends and impact created by the different simulation conditions, along with potential explanations and improvements for future simulation. Chapter V summarizes the document and discusses avenues and objectives for future research in this area.

II. Literature Review

2.1 Tire Pressure Monitoring Systems

The introduction of Tire Pressure Monitoring System, hereafter referred to as TPMS, has increased safety by reducing the number of vehicles with one severely under-inflated tire by 55.6% [2]. Tires under-inflated by 25% or more, the threshold for a TPMS warning light, "...are 3 times as likely to be cited as critical events in the pre-crash phase"[3]. When equipped with a monitoring system, drivers are rapidly warned if there are issues with the tire pressure in any of their tires, allowing them to take timely measures to prevent further damage. The operating environment, a rotating wheel, necessitates wireless devices to measure and transmit data, resulting in a system with an attack surface that does not require physical contact with the vehicle, and shortcuts that expose vulnerabilities. The dependence on a battery, and the requirement for a long service life, impose significant constraint on the design and implementation of TPMS systems. Despite being a requirement on most new vehicles in the United States for over a decade, there has been little literature examining TPMS from a cybersecurity standpoint. One goal of this document is to shed light on the potential malicious uses of these devices, and spur research and security improvements before any serious exploits are found in the wild.

History of Legislation.

In the United States, steps toward mandating TPMS on new vehicles occurred after a series of fatalities related to defective tires. The Transportation Recall Enhancement, Accountability and Documentation (TREAD) Act was rapidly passed by the U.S. Congress (2000), and called for a mandated system to warn drivers when a tire is significantly under-inflated. The National Highway Traffic Safety Administration drafted the more detailed requirements, requiring compliance on new vehicles beginning September 1, 2007 [4]. These requirements included reporting to the driver if one or more tire was 25% below minimum pressure, within 20 minutes of the pressure dropping. The European Commission likewise mandated TPMS on new vehicles after 2012 as part of a safety and emission-reduction program [5]. As a result, there are millions of these sensors on the roadways, with a growing percentage of TPMS-equipped vehicles as older cars are removed from the roadway.

Methods of Measurement.

There are two primary ways for the TPMS mandate to be fulfilled; indirect and direct measurement. The key difference between these approaches is that indirect measurement uses wheel speed and known wheel size to calculate pressure, while direct measurement embeds sensors in the tire to gather true measurements. Direct measurement necessitates wireless communication to achieve these results. Indirect measurement can be achieved with wired communication, but with a loss of accuracy.

Indirect Measurement.

Indirect measurement leverages input from sensors that are already in use by other systems in the vehicle. Specialized software can estimate pressure changes by monitoring minor changes in wheel speed that are a result of small changes in wheel diameter, which in turn are a result of a tire pressure differential. These systems have the advantage of using existing hardware, but have several drawbacks that make it poorly-fitted to meet legal requirements. Indirect TPMS is less accurate, doesn't have truth data for the actual tire pressure, and can give false readings when non-standard tire sizes are used or if tires lose pressure simultaneously [6]. As a result, the vast majority of vehicles incorporate direct measurement.

Direct Measurement.

Direct TPMS is the standard used in most vehicles today. It is composed of a pressure sensor, a battery, and transmitting hardware placed inside the tire, typically integrated with the valve stem which is pushed through the rim to allow for inflation. A picture of an example device used during this research is shown in Figure 1. Because the sensors must last the life of the tire, battery life is a critical consideration. To meet these constraints, the sensors must not use any energy when the car is parked. When the vehicle stops, typically the accelerometer on the devices takes note and stops transmitting after a period of being stationary, such as 10 minutes. When a car is turned on, a low frequency wake up signal is sent to the sensors which awakens them from a standby state. For further energy savings, the sensors typically only transmit every 1-2 minutes during normal operation, or more frequently if a problem is detected or a wake-up signal is sent. The vehicle is equipped with either four individual antennae, or a single centralized receiver. This approach is much more accurate than an indirect method, with accompanying higher costs, but also introduces wireless vulnerabilities, which are presented in Section 2.2.

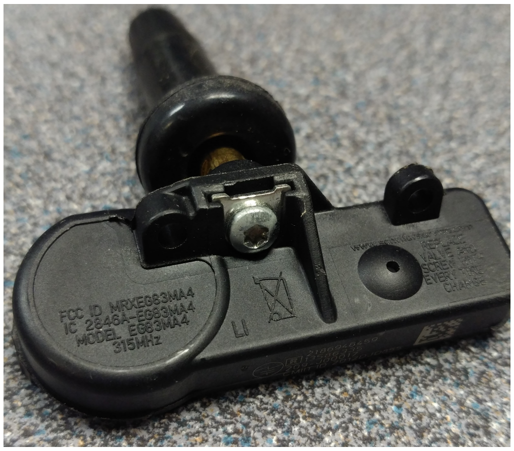


Figure 1. OEM TPMS Direct Measurement Sensor

Typical Implementation.

For direct measurement, the wireless protocol is straightforward. Notably, information is only transmitted one way, from sensor to main vehicle, thereby relieving the sensor of the cost of powering a receiver. The in-tire module reads the pressure and temperature sensors, constructs a data packet, encodes using a method such as Manchester Encoding, and finally transmits it using Amplitude Shift Keying or Frequency Shift Keying. An example packet might include 32 bits of ID, 8 bits of pressure data, 8 bits of temperature data, 4 bits of status flags, and 12 bits of a Cyclic Redundancy Check (CRC), as shown in Figure 2. The ID assigned to a sensor and broadcast with every packet does not change for the lifetime of the device. This leads to the naming convention used in the remainder of this thesis, Static TPMS, referring to the static IDs belonging to each sensor. None of the data in a packet is encrypted or obfuscated for security, allowing any listener in proximity to read this data. One reason for the lack of security is the energy cost that encryption or two way communication would require. Existing sensors must be completely replaced when the battery is depleted, with a typical lifespan being five to ten years. The batteries are usually set in epoxy and therefore not replaceable, while size and weight requirements prevent the use of a larger battery.

$$\underbrace{ID}_{\text{32 bits}}\ \underbrace{PressureReading}_{\text{8 bits}}\underbrace{TemperatureReading}_{\text{8 bits}}\underbrace{Flags}_{\text{Variable}}\ \underbrace{CRC}_{\text{Remaining bits}}$$

Figure 2. Allocation of bits in an example TPMS packet

2.2 Attack Scenarios

Introductory work on attack scenarios against TPMS was done by [7] as a case study for in-car wireless networks.

Spoofing.

The researchers demonstrated that spoofing packets is trivial once the packet and transmission characteristics are determined [7]. The first proof of concept shown was to spoof a packet with the ID “DEADBEEF” and pressure value of 0 PSI on a TPMS trigger tool. These trigger tools are handheld receivers that transmit TPMS wakeup signals and can be configured to read the packets, used to check the status of a vehicle’s sensors. They then spoofed low-pressure packets to a real vehicle, where they were able to illuminate the TPMS warning light falsely, even when spoofing from another vehicle while the victim and attacker were traveling at high speeds.

Denial of Service.

The low transmission power of TPMS sensors is easily overpowered by either narrow or broadband frequency jamming. However a noisy attack such as this is fairly easy to detect, and the driver would also be warned that the sensor is unresponsive. Focused methods would be more difficult to trace. For example, [7] were able to deny the functionality of the service in two ways: first as a part of a spoofing attack, resulting in unreliable readings, and secondly, and apparently inadvertently, by disabling the TPMS Electronic Control Unit(ECU). In the process of testing packet transmission frequency, the researchers disabled the ECU, illuminating a ‘Check TPMS System’ warning, and causing the car to display no tire pressure information. Attempts to restart or recover the ECU were unsuccessful, and the only solution was to replace the ECU at a dealership, demonstrating that packet spoofing can lead to permanent cyber-physical effects.

Demonstrated Attacks on Vehicles.

As the infusion of technology into vehicles continues, interest into the exploitation of that technology has grown. One of the most notable are the exploits described in [8], in which the researchers penetrate several vehicles including a 2014 Jeep Cherokee, through multiple attack surfaces. Features such as the entertainment system and cellular access allowed them to inject payloads into the vehicle, controlling systems such as braking, locks, headlights, and the instrument cluster. Exploits that stall the engine, disable the airbags, and disable power steering are described in [9, 10], and require fairly minimal resources once a connection to the CAN bus is established. The ECU that communicates with the TPMS may control other vehicular systems, raising concern that the TPMS receiver ECU may become the next attack surface to launch these demonstrated payloads.

2.3 Tracking

Tracking Algorithms.

There are many algorithms available for tracking objects, often tailored to the detectable data and the specific problem space. Several have been tested against a real-world vehicular data set, including those based on particle filters, Markov Chains, and RANSAC [11]. From the input data, these algorithms typically create many possible paths which get progressively refined. Researchers have been able to constrain particle filters to road networks to reduce computational effort and uncertainty in results [12].

Tracking using these algorithms has been shown to be effective, but this research explores the possibility of modelling the road network as a graph data structure, rather than as particles. This would automatically incorporate assumptions about where a vehicle could travel and change routes, without having to modify other algorithms. Additionally, using these structures could leverage available Graph Theory algorithms and optimizations. Even when a tracking algorithm performs well, it can have difficulty associating tracks to identities, which is the primary contribution of this research. Future work could therefore use TPMS as a detection and association tool, to feed into a more typical tracking algorithm.

Tracking Via TPMS.

Rouf et al. also explored the feasibility of tracking a vehicle via TPMS transmissions [7]. Given the ID's of the four tires associated with a vehicle, which do not presently change, it is simple to associate this with the identity of a vehicle. There would need to be over one billion vehicles on the road to even approach a 1% chance of misidentifying a vehicle, given the data from all four tires, without considering any other factors such as geographic location. Admittedly, there are some formidable challenges to creating an eavesdropping infrastructure, particularly with passive data collection. The low power transmissions from the sensors limit the range of receivers, so a large number of eavesdroppers must be placed to be guaranteed to capture the infrequently transmitted packets. A more effective solution explored by the same researchers would be to stimulate a transmission with the low frequency activation signal, and have a high frequency receiver co-located. While noisier and slightly more complex, this could be used to guarantee a reading at points of interest for those who wish to track using TPMS. They concluded their section on tracking with a comparison to a system in use, Automatic Number Plate Reading (ANPR). Tracking via TPMS would have the advantage of a higher read rate (99% vs 90%) and not requiring Line of Sight, but would require changes to the legal system to be effective for law enforcement.

2.4 Pseudo-random Number Generators

A Pseudo-Random Number Generator(PRNG) is a type of Deterministic Random Bit Generator(DRBG) which produces a pseudorandom sequence of bits using an initial seed and potentially other inputs [13]. True random number generation is difficult to achieve, and is actually not desirable for the application in this document. PRNGs rely on a cryptographically secure one-way function to operate. Given the same initial conditions, a PRNG will always generate the same sequence of numbers, however an outsider should not be able to increase their odds of determining the next number that will be output. A block cipher with a secret key is considered a suitable one-way function to operate as a PRNG [14].

There are many valid block ciphers which could be considered sufficiently secure for the requirements of the rolling ID scheme proposed in Chapter 3. Therefore, the options must be weighed using additional criteria. The TPMS environment currently necessitates low-power operation. Many of the challenges of the TPMS environment are shared by Internet of Things (IoT) devices, creating a wider body of research from which to draw. Certain block ciphers have been designed specifically for low construction and power costs, and considered in this section. This section does not seek to prove that a specific implementation is optimal, but will examine a select few to demonstrate what should be considered in a future device and pick one to use as an example in later investigations.

SIMON.

The SIMON and SPECK family of block ciphers proposed in [15] were designed to be flexible and secure block ciphers that could be tuned for a wide variety of platforms. SPECK is tuned for optimal performance in software, while SIMON is tuned for hardware performance. The current size of a typical tire ID is 32 bits, so for compatibility and minimizing complexity this is an ideal block size. These protocols support a 32 bit block size with a 64 bit key. In practical implementation, this key could be printed in a format such as hex, base64, or QR code on the sensor to be read before installation. The implementation of this specification in the paper is referred to as SIMON32/64. It can be implemented in a Gate Equivalent (GE) area of 523 gates. Not all block ciphers can be configured to operate with blocks and keys this small; for size comparison, Table 1 shows the GE, flash memory, and SRAM required for different ciphers to be implemented on 64 bit block sizes

Table 1. Gate Equivalent Area for Block Ciphers (64 bit block size)

Cipher Name	Gate Equivalent	flash (bytes)	SRAM (bytes)
SIMON[15]	838	274	0
SPECK[15]	984	186	0
TWINE[16]	1011	1304	414
PRESENT[17]	1030	487	0
PICCOLO[18]	1043	unavailable	unavailable
KATAN[19]	1054	272	18
KLEIN [20]	1478	766	18

PRESENT was shown to have feasible power and size requirements in [21], however the goal of that paper was to show that PRESENT can be used, not to determine how ideal it was. Based on the hardware requirements of SIMON in relation to PRESENT, and the fact that PRESENT is not implemented for 32 bit block size, SIMON can be expected to operate within the constraints as well.

2.5 Proposed TPMS Security Improvements

Some researchers have proposed solutions to increase the difficulty of TPMS tracking and spoofing attacks by obfuscating the packet ID. [22] proposed a system which incorporates pseudo-ID's, sequence numbers, message authentication codes, and session keys, solving many of the privacy and integrity issues present. However, this system requires a 3-way handshake to establish the key, and current TPMS sensors are not equipped to receive data. [21] examined the cost of encryption on the sensors and demonstrated that an implementation of the PRESENT protocol could be used within the typical constraints. [23] proposed a system using a Linear Feedback Shift Register to generate a new ID, using the initial state and a polynomial as the shared secret between sender and receiver. This achieved the goal of rolling, or hopping IDs, but did not explore issues of desynchronization between the parties. [24] proposed a system with rolling ID's and optional encryption, and tested their system on development hardware to demonstrate feasible power consumption. Their system extended packet size and required a 3-way handshake to establish a session ID, which is not possible with one-way communication.

2.6 Summary

This chapter presented the background information and research surrounding TPMS. The history and operation of TPMS devices was described, along with attack scenarios that it presently faces. Expanding on the tracking attack scenario, simulation conditions necessary to demonstrate feasibility were examined, as were tracking algorithms that could be used by an adversary. Several PRNGs were compared to determine one that would fit the constraints of TPMS to enable an updated version to mitigate a tracking attack. Other solutions in literature were briefly reviewed, but were shown to be missing elements that are desirable in the operating environment.

Information from this chapter was used to construct a simulation environment and set of experiments, propose a solution, and test the feasibility of tracking and the potential gains of an updated TPMS.

III. Methodology

3.1 Introduction

This chapter describes experiments to analyze TPMS susceptibility to tracking and proposes a solution, referred to as Tire Obfuscation through Rolling ID (TORI). TORI solves many of the known vulnerabilities of Static TPMS, and increases the difficulty of exploiting those that remain. To understand the privacy risks of Static TPMS and the proposed TORI solution, the remainder of this chapter details the experimental design of an appropriate simulation.

The first task is to simulate the flow of traffic, including creating a road network, generating traffic, and placing detectors to gather TPMS data. This chapter then introduces the tasks of tire association and route reconstruction that an attacker would have to accomplish in order to track a vehicle using TPMS data. The algorithms for the Associator and Route Reconstructor are presented, along with a scoring metric to compare the results between experimental conditions. Details of applying this experiment to TORI are discussed, including the potential for collisions that may be introduced.

3.2 Recreation of Past Work

Proof-of-Concept spoofing experiments were recreated early in the research for this document. First, a USRP N200 Software Defined Radio (SDR), with a GNURadio program was used to detect the packets. Figure 3 is a picture of the TPMS trigger tool used to induce a packet broadcast for a sensor from a 2012 Toyota Avalon. The SDR detected the packet, which was then decoded in MATLAB. After determining the packet encoding and structure, the SDR was configured to send spoofed packets, which could be set to any ID, Temperature, Pressure, and Battery Level. This activity demonstrated that successful eavesdropping or spoofing is trivial to accomplish, as the frequency and keying scheme are available from the FCC, and no security obstacles were present in that device. An attacker with enough time to determine the exact structure of all the TPMS sensors on the road, or insider knowledge, could detect and decode packets from any vehicle. The same information can also be used to successfully spoof packets and convince a TPMS receiver to forward erroneous information to other onboard systems such as the instrument cluster.

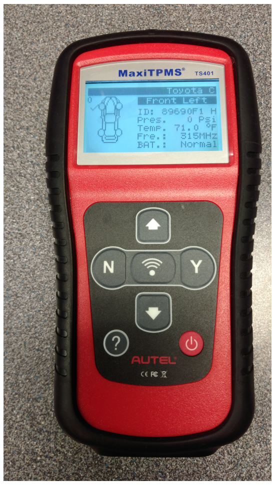


Figure 3. Handheld TPMS Reader

3.3 TORI

This section describes the implementation of Tire Obfuscation through Rolling ID (TORI). The goal of TORI is to render the IDs in TPMS packets unusable to an attacker. Accomplishing this goal mitigates the ability to track using TPMS and hardens the system against spoofing. The ID hopping scheme is described here, which requires a PRNG and secret key to generate Pseudo-IDs. The requirements for a receiver to understand sensors equipped with TORI is then described.

ID Hopping.

An alternative system that could enhance privacy would ideally also match the current structure when possible, to maximize backwards compatibility. Therefore, in the proposed TORI system, the basic packet structure of an ID, pressure, temperature, flags, and CRC would remain. The proposed change simply allows for a hopping ID. Currently, the sensor IDs are transmitted in the clear and do not change, so with enough reception points it would be simple to associate these ID's with a single vehicle. In TORI, the sensor would utilize a pseudo-random number generator (PRNG) which takes the last ID and a unique secret key associated with that sensor as input to generate the next ID. Such a system would have seemingly random hopping ID's, defeating tracking by preventing an eavesdropper from easily associating ID's with a specific vehicle. An attacker seeking to spoof packets would not be able to guess the next ID without the secret key, defeating simplistic attacks based on spoofing. (Note that a sophisticated attacker, after collecting a significant number of messages, may be able to determine the secret key. However, the cost to the attacker is significantly higher in this case.) The key elements required to implement the ID hopping scheme are described in greater detail below.

Pseudo-random number generator.

A number of lightweight block cipher implementations exist and could be used as pseudo-random number generators. These are designed for low power applications and require a low Gate Equivalent value, roughly denoting the number of transistors required to implement it. These include PRESENT, Simon, XTEA, KTANTAN, and Piccolo, among others. For the remainder of this paper, the Simon protocol will be used as an example of a suitable cipher, though the purpose is not to provide in-depth analysis of the optimal choice. Simon is a lightweight protocol tuned for hardware implementation proposed in [15], and published by the National Security Agency (NSA). It can be implemented in many different key and block sizes, allowing it to be extended for longer keys or packet sizes in the future. These are discussed in greater detail in Section 2.4

Pre-shared key.

The solution proposed here would require a secret shared key between the sensor and receiver. Currently, the 28 or 32 bit ID for a sensor is usually printed in hex directly on the sensor casing, which is inaccessible once it is installed in the wheel and a tire has been mounted. This ID would be replaced with a secret key of longer length, which could still be printed in hex or in another format, such as a QR code, directly on the sensor. This key would be registered with the vehicle by a dealer or tire shop upon installation, to allow the vehicle to recognize its own tires. (Incidentally, this sets the upper effort threshold for determining the key at removing the tire and reading the sensor key or being present at installation, but that would require such targeted effort that it is not considered a scenario to be addressed by TORI. If such threats were realistic, then the key could be printed on a separate card and stored somewhere safely or destroyed after registration.) Note that these secret keys are intended to last the lifetime of the sensor, which may remain with the wheel even as tires wear out and are replaced, but is presently limited by battery life.

Pseudo-IDs.

In the table below, the Simon protocol is used to generate a hopping sequence of 32 bit pseudo-IDs as a function of their previous ID and a 64 bit secret key. Simon is implemented in 64 bit key mode, with a 32 bit block size, in Electronic Code Book (ECB) mode. Each tire is assigned a randomly generated secret key, and the initial

ID is set at 0x00000001.

Table 2. Example ID hopping based on Simon protocol

Tire:	Left Front	Right Front	Left Rear	Right Rear
Secret Key:	0x1e69817e96552645	0x0d5642034fc6feac	0x743e4bc52d349a51	0x564c4ac4314e3c5a
Round 0	0x00000001	0x00000001	0x00000001	0x00000001
Round 1	0xf21ae8fa	0x6ba0c094	0x0ce07b83	0xb9f00520
Round 2	0x672bbfa5	0x5890f0cc	0x7589fc2e	0x7e7c1540
Round 3	0x9cb82f3b	0x39edf661	0x95d9e232	0x29da9a76
Round 4	0x44518033	0x5808aa81	0xcff832cc	0x06daf377
Round 5	0xc2883e72	0xfea4c40d	0x8e489cbf	0x9f929b62

The ID from the last round, along with the secret key, is used as the input to generate the next ID. In the presence of many moving vehicles, an eavesdropper would no longer be able to easily associate tires (pseudo-IDs) with a vehicle, as the ID changes with every transmission.

Sensor-Receiver Synchronization.

One issue that can arise from hopping pseudo-IDs is the handling of tire association from an untrusted state. The system proposed here solves this by implementing a simple detection algorithm to receive and check the authenticity of packets without requiring the TPMS sensors to receive any data from the vehicle. At startup, the receiver would have an empty packet buffer and not yet trust any sensors. As TPMS data packets are received, either due to a low frequency activation signal from the vehicle or as a part of normal operation, these packets are buffered. The receiver, knowing the secret keys for each of its tires, can calculate the next N expected pseudo-IDs as a function of the packet's pseudo-ID and the four registered secret keys. After receiving a predetermined number of packets matching the expected sequence, that tire would be considered synchronized with the receiver. Figure 4 demonstrates an example where an untrusted packet arrives, and the receiver calculates the next 5 expected ID's for each tire. The sequences are calculated with the Simon protocol as described in section 3.1.3. The central buffer shows received packets, which arrive from multiple tires that may be registered to the vehicle. A comparison of the buffer and the expected IDs shows that the Right Front tire is transmitting an expected sequence of ID's, and could be considered synchronized.

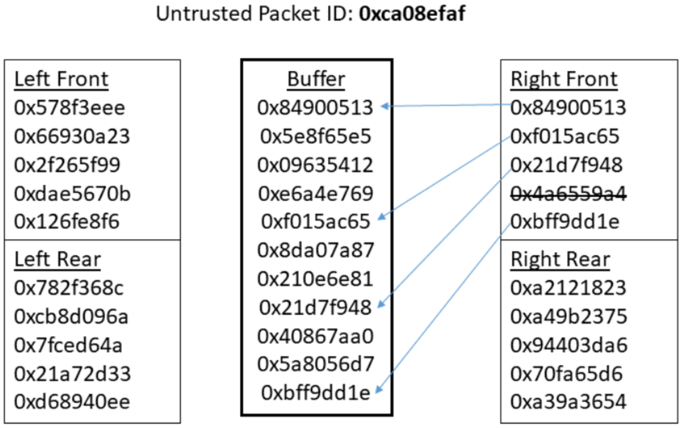


Figure 4. Example of Tire Resynchronization Based on Expected ID Sequence

Note that in this figure, the IDs listed under each tire are the expected sequence based on the untrusted packet shown at the top. Because the receiver does not know which tire the packet belongs to, it generates a potential sequence using every tire's secret key, knowing that most of those sequences will be discarded. Only one (or none, if the packet is from an external source) of the sequences will match, in this case the Right Front tire. The unmatched packets in the buffer represent real received packets, that belong to the other tires, and would have their own synchronization process. This process would be followed for every untrusted packet that arrives, which is not frequent in normal operation. If a matching sequence does not arrive in time, the vehicle could send activation signals to stimulate a "quick sync", causing the sensor to send multiple packets within seconds, rather than the minutes it would normally take. If this is unsuccessful, the receiver could enter a failure state and alert the driver via the dashboard warning light.

The resilience of this system to packet loss and spoofing attacks is a trade space based on the receiver memory and antenna placement. The existing system has no concept of packet sequence, so as long as packets arrive frequently enough for the sensor to be considered responsive, there are no issues with lost packets. If the synchronization algorithm only calculated the next expected ID, a packet loss could cause the receiver to lose trust in a sensor. Calculating the next N expected ID's adds resilience by allowing N-1 sequential packets to be lost without desynchronizing.

The number and location of antennae also affect the difficulty of tire association and signal interference. If the vehicle has one receiving antenna per tire, it can be calibrated such that interference from other tires on the vehicle and other vehicles on the road is unlikely, also saving power on the sensor transmissions. A central receiving antenna would save on material cost but suffer from potential packet collisions, and the control unit would have to distinguish between packets coming from all 4 tires into a single buffer.

Under normal operation, packets are infrequent such that even a small buffer could hold enough packets to maintain synchronization and handle packet loss. For example, if the receiver calculates the next 5 expected packet ID's, for each incoming packet it compares the ID to the list of expected ID's for each tire. If it matches, then that packet is parsed for the matched tire. A packet that does not align with any expected sequence could be thrown out until a tire is considered out-of-sync, at which point untrusted packets are used to calculate expected sequences to attempt to resynchronize. This creates a new vulnerability arising when a large number of packets are spoofed to fill this buffer and prevent synchronization, to be discussed in later sections.

A simplified version of this TORI receiver was implemented in software to demonstrate viability. Four tires and a central receiver were modelled. Each tire was given a unique secret key, known to the receiver as well. The tires generated packets every 90 seconds, rolling their ID according to the SIMON cipher with each transmission. Every second of simulation has a pre-determined chance of packet injection (with a random ID), and a pre-determined chance of starting an emergency condition on a tire. When in an emergency state, the tire transmits every five seconds for one minute, then clears the emergency (unless an emergency was reissued, which resets the 60 second timer). These timestamped packets are fed into a receiver program, which maintains separate buffers for each tire. When a packet arrives, it is immediately compared to the Expected ID buffer for each tire. If there is a match, the packet is accepted by that tire. If there is no match and all tires are considered in sync, the packet is discarded. If any tires are not in sync, it is placed in that tire's buffer and marked as suspicious. The next five expected IDs for that suspicious packet are stored. If a later packet arrives that matches that sequence, that suspicious packet is accepted and the new packet, and the tire is considered in sync. Sync is lost when no new packets have arrived in two cycles, 180 seconds.

The implementation described here is not considered complete or ideal. It does demonstrate feasibility, and provided initial insight into how one may design a receiver in the future. The most difficult problem for the tires was achieving first sync. Lengthy time between packets and high packet loss conditions made receiving two consecutive packets difficult, so achieving first sync and recovering from a desync are problems to consider. Once synced, the rejection of unexpected packets presented strong resilience to packet injection. If a tire is not synced, however, these erroneous packets will be kept in the suspicious buffer. This presents a memory-performance tradespace, where keeping more suspicious packets allows for more resilience to packet injection while desynchronized. The number of IDs calculated ahead (five in the proof-of-concept implementation) is related to resilience to packet loss. Under normal conditions a tire receiver will time out before missing five packets, but if the transmission frequency were increased (as in an emergency) and packet loss is high, it is possible that the sequence will roll ahead past the Expected IDs buffer. The memory requirements still remain low, so a relatively large amount of memory could be allocated to these buffers to make such a case very unlikely.

3.4 Simulation Construction

Traffic Simulation.

Demonstrating the feasibility of tracking via TPMS data requires a large sample set of realistic vehicles and driving patterns. Physical deployment would require a large investment of resources to develop and deploy detectors, and would require the collaboration of several parties over a lengthy period of time to complete. Employing a simulator shifts the power into the hands of the researcher, and allows for control over nearly every detail, along with the ability to rapidly change conditions and run many trials. There are two general types of simulators considered for this. The first are microscopic simulators, where the level of simulation goes down to individual vehicles and lanes, acting on their own and responding in a realistic manner. In contrast, macroscopic simulators abstract the individual vehicles into a general traffic flow in a section of a map. To meet the goals of this research, a simulator was needed that was intuitive, realistic, had a variety of maps available, and could be used to generate TPMS data.

Simulator for Urban MObility (SUMO).

SUMO is an open source traffic simulation suite that provides several tools for map and traffic generation, manipulation, and simulation. First released in 2002, it continues to be actively developed, providing a platform to test routing protocols, run traffic congestion models, and generate realistic traffic data that can be used for further research [25]. Maps are modeled as nodes and edges mapped to a Cartesian grid, and can be constructed, randomly generated, or imported from other sources such as OpenStreetMap [26]. Traffic conditions such as the number of lanes, traffic light timings, speed restrictions, and more can all be specified or imported. Vehicles can belong to standard classes such as car, truck, or bus, or customized to meet the needs of the simulation. Simulations are defined by the map and route files. The map establishes the places a vehicle may travel, along with road conditions and restrictions, while the route defines the points at which a vehicle enters and exits the roadway, which roads it travels on, and how it behaves during the trip. The simulation can be modified while online using the Traffic Control Interface (TraCI) to observe how changing conditions such as traffic lights or a collision may affect the simulation.

Map.

The map used for simulation is a section of the downtown Dayton, OH area. SUMO includes a tool to use OpenStreetMap data to download real data for an area, such that it accurately represents the traffic lights, speed limits, one-ways, and other elements of traffic flow. The size of the map is a simulation parameter that may be adjusted to serve different purposes; the chosen size was approximately 600 nodes and 1200 edges. This map was selected for the sake of familiarity and as a representational map, with sufficiently diverse traffic conditions to demonstrate feasibility. Urban deployment with a relatively high density of intersections was of particular interest in this study. The selected map of Dayton is shown in Figure 5

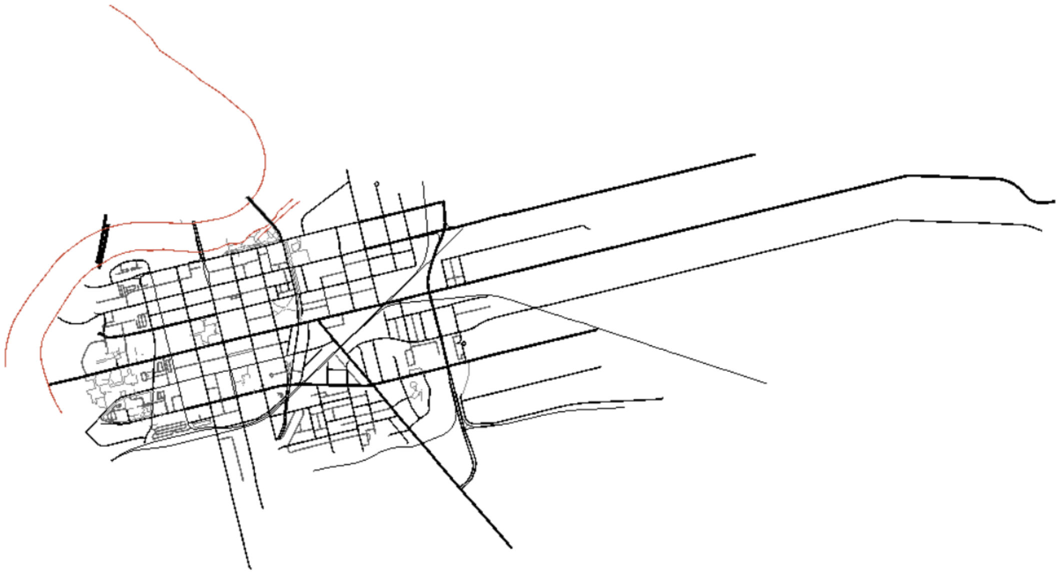


Figure 5. Map of Downtown Dayton in SUMO

Traffic Generation.

To generate traffic, the randomTrips SUMO Python script utilizes the network description, simulation time, optional seed, and traffic density to generate an XML trip file describing every vehicle created and its source node and destination node. The SUMO tool DUAROUTER then converts these source/destination pairs into actual routes that describe what roads the vehicle will attempt to take during simulation.

Wireless Detection.

SUMO includes a package for wireless communication which can be manipulated to model various technologies such as Bluetooth and Vehicular Ad-Hoc Networks (VANET) [10]. Vehicles can be given receivers and transmitters independently, and assignment can be done explicitly or randomly with a provided percentage. For these experiments, 100% of vehicles had transmitters, representing the vehicles being equipped with tire pressure sensors. TPMS detectors are modeled as vehicles, equipped with receivers, which park alongside the road at every intersection, as shown in Figure 6. Edge cases in which multiple intersections were very close to each other were handled by manually removing overlapping detectors. When a vehicle enters the detector's range, it is recorded in an XML file at the end of simulation that describes a large amount of detail about the vehicle's identity and travel conditions, which are used later to compare the algorithm results to the baseline truth provided by the simulation. At this stage, the density of detector deployment can be altered by optionally eliminating a percentage of detectors.

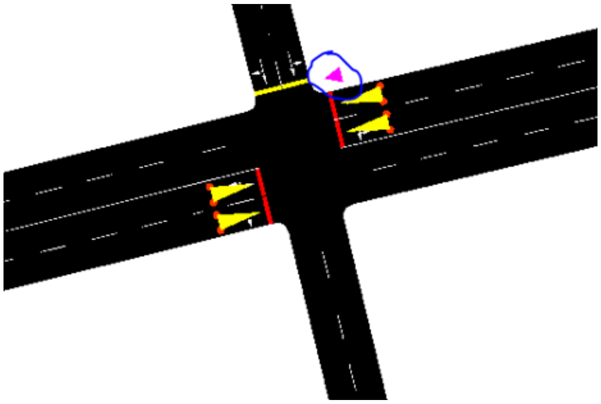


Figure 6. Detector Vehicle (Magenta) Parked At Intersection

This method of modeling the wireless detectors is not a perfect representation of how real detectors would be deployed, but provides sufficient fidelity to achieve the research goals. A real detector would likely be a directional antenna which would only receive data from a few lanes, possibly requiring 2n detectors for an n-way intersection. This could actually improve tracking algorithms by giving travel direction, but for this simulation only binary detections are considered at intersections: Was a vehicle present at this intersection, and at what time? Based on previous research, and working within time constraints, this is a feasible type of input for later association and tracking stages. An additional benefit of this approach is that timestamped vehicle position data could come from detectors other than TPMS, which is particularly of interest as VANET technologies evolve, but can be applied to existing systems such as Automatic Number Plate Readers.

SUMO generates wireless observation data, after which post-processing converts this data into TPMS packets provided to the Tire ID Associator. In this post-processing step, most of the data is stripped so that later stages do not have access to insider information, such as the true vehicle ID, speed, or route. This stage begins with a dictionary containing the time and location data for all wireless observations, indexed by the observed vehicle ID. For each unique vehicle, four random 32-bit Tire IDs are generated. At each observation for that vehicle, the simulation must decide which tires would have been observed. Based on past experiments involving a directional antenna and measuring the attenuation due to a vehicle, a simple probabilistic model was assumed as follows. The transmission from the two tires nearest a road-side detector would always be received, (i.e., Right Side tires are always received), and the Left Front and Left Rear tires are detected with a 50% and 10% probability, respectively. These percentages are derived from the signal strengths observed in the eavesdropping range experiments in [7]. Future work includes a more detailed modeling scheme that incorporates directional antennas and speed of travel for a more accurate representation. If a tire is considered to be detected at an observation point, then that location and timestamp is placed into a new dictionary indexed by tire ID, such that the resulting data structure no longer contains the true vehicle ID, and is at most 4x larger than the original.

3.5 Experimental Metrics

This thesis introduces possible algorithms to identify vehicles and reconstruct their routes. In order to evaluate the effectiveness of these algorithms, metrics comparing the results of these algorithms with truth data from the simulation are needed. Note that these metrics are intended to have value across simulations, to form a relative sense of "goodness" for various configurations, but do not map to any external metrics. The performance of the algorithms in this document will be evaluated using metrics particularly suited for the data they operate on. Data from the simulation phase flows first into the Associator, which transforms TPMS observations into vehicle identities. Those identities and associated observations are sent to the Route Reconstruction phase, which transforms individual observations into complete routes. The two metrics used here are Jaccard distance for the Tire ID Association phase, and Graph Edit Distance for the Route Reconstruction phase.

Jaccard Distance.

The Tire ID Association phase relies on comparing sets to determine which combinations of IDs are commonly found with one another. Jaccard distance is a set similarity metric that is commonly used for spell checking on strings [11]. It is defined as

$$JaccardDistance(A, B) = \frac{(|A \cap B|)}{(|A \cup B|)}$$

The Jaccard distance can be used to compare a proposed set of Tire IDs with those observed within a time window at a specific intersection along a route. For example, if the proposed set is $(0xa1, 0xb2, 0xc3)$ and all the IDs observed in a 5 second window at an intersection form the set $(0xcc, 0xdd, 0xff, 0xa1, 0xb2)$, then the Jaccard distance can be calculated as:

$$\frac{|0xa1, 0xb2|}{(|0xa1, 0xb2, 0xc3, 0xcc, 0xdd, 0xff|)} = \frac{2}{6} = 0.3\bar{3}$$

This score has value in determining the best grouping of Tire IDs, which will be used in Section 3.6. It is also used as an evaluation metric for final identities produced by the Associator, by measuring the distance between a proposed identity and the closest true identity. Note that the discrete values possible vary based on how many tires the algorithm has grouped together, and how many tires a vehicle may have.

Graph Edit Distance.

Graph edit distance is a method for comparing the similarity between two graphs, useful for pattern matching [27]. It can be defined as the minimum number of modifications required to transform one graph into a target graph. These modifications can take the form of insertions, deletions, or substitutions, on nodes or edges. Each can be weighted differently to reflect the impact an operation would have, but in this usage insertions and deletions are weighted as a cost of one, and substitutions are weighted at two (representing one deletion and one insertion).

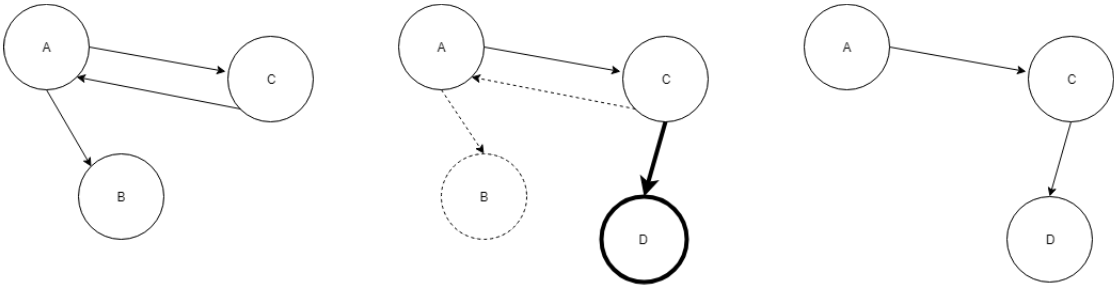


Figure 7. Graph Edit Distance Example

Figure 7 illustrates the conversion of one graph to another, with an edit distance of five. Dashed lines represent deletions, of which there are three. Node D and edge C-D are insertions, represented in bold, incurring a cost of two, for a total cost of five.

A weakness of graph edit distance in this application is that it does not account for the length of the road. Therefore, a different metric would need to be used to answer questions such as "What percentage of the route is the vehicle accounted for?" One manipulation of graph edit distance that helps to account for this is the Edits Per Node Travelled Ratio, which is used in Chapter IV. It represents the Graph Edit Distance between the real route and predicted route, divided by the number of nodes in the real route. It aids in predicting how many mistakes the algorithm will make based on the length of a vehicle's route, and removes some bias in experiments where the routes may have been longer.

3.6 Tire ID Association

Grouping Algorithm.

The next step in the pipeline is to attempt to associate the observed tire IDs with one another, forming a tuple that ideally belongs to one vehicle. Each tire ID has an associated list of observations, forming a route. Due to the improbability of getting all four tires at every intersection, tire IDs that belong with each other (are from the same car) will have similar, but not identical lists of observations. All tire IDs observed within a specified time window (chosen as one second in these experiments), and at a specified location are examined. For every tire ID, the frequency that each other tire ID was observed "near" the selected one are tallied across the entire observed route. Because two cars may be near enough to overlap tire IDs, it is expected that the algorithm will have to filter tire IDs from nearby vehicles, arising from a noisier environment as traffic density increases. This filtering occurs in the next stage of association.

All combinations of the four most frequently observed tire IDs, in relation to the one being evaluated, are then compared with the sets observed at each location using Jaccard Distance. The set with the highest average Jaccard Distance across the route is considered to be an identity, and is saved in a scoring matrix. The purpose of this is twofold; first, it allows for a tunable metric to be used to manipulate the risk/reward of associating more tires. Secondly, because the route for tires may appear different even if they belong to the same vehicle, it is possible that a sparsely observed tire appears to be associated with a different set of tires. The scoring matrix adds a second layer of filtering to reduce that error.

```
Input: Dictionary $Obs_{in}$, containing a location-timestamp list of observations
       indexed by Tire ID
Output: Dictionary $D_{out}$, containing a location-timestamp list of observations
        indexed by vehicle ID
Initialization:
scoreMatrix=2-D array indexed by all observed Tire IDs, initialized to 0
vehNumber = 0
for id in $Obs_{in}$ do
    associated = []
    for (loc,time) in $Obs_{in}$[id] do
        Add nearby Tire IDs to associated
    end
    tireSet = id+three most frequent Tire IDs in associated
    x = Combinations(tireSet)$_1$
    bestSet = Jaccard(x,$Obs_{in}$[id])$_2$
    for tID in bestSet do
        for $tID_2$ in tireSet do
            scoreMatrix[tID][$tID_2$] += 1
        end
    end
end
```

Algorithm 1: Association Algorithm: Initial Association

```
for id in scoreMatrix do
    confirmed = [id]
    proposed = id+three most frequent IDs in scoreMatrix[id]
    for $id_{prop}$ in proposed do
        related = three most frequent IDs in scoreMatrix[$id_{prop}$]
        if id in related then
            append $id_{prop}$ to confirmed
        end
    end
    $D_{out}$[vehNumber] = MergeObservations(confirmed)
    vehNumber += 1
    for $ID_{used}$ in confirmed do
        Delete $ID_{used}$ from scoreMatrix
    end
end
1: All combinations of size 1,2,3 or 4 including id
2: Returns set with highest average Jaccard Similarity along id's route
```

Algorithm 2: Association Algorithm: Final Association

After all tire IDs have been considered independently, the scoring matrix is evaluated to form the final virtual vehicle identities; i.e., the set of tire IDs belonging to one unique vehicle. If a set of tire IDs were consistently grouped into a set of four or less with each other, those are proposed to be a specific vehicle and are removed from further consideration. If the algorithm had attempted to group tire IDs such that more than four appeared to be related, the four that were most frequently associated with each other were taken as one vehicle.

Scoring the Associator.

To evaluate the effectiveness of the Associator under various conditions, a scheme was selected to satisfy a small number of principles. First, the tire IDs that the algorithm grouped as a proposed vehicle should be evaluated with respect to each other, rather than attempting to match them to a specific true vehicle, and second, the risk and reward of attempting to add a third or fourth tire to a set should increase. The method used to score the output of the Associator considers each set that was grouped together, then performs a reverse lookup in the truth data to find which true vehicle ID each tire ID was associated with. This creates a tuple of one to four elements, each of which may correspond to the same vehicle or perhaps a different vehicle. The most frequently occurring vehicle ID is used to score, by comparing the set of predicted sensor IDs with that vehicle's true set of IDs using Jaccard Similarity. This produces a value ranging from 0.14 to 1 for four-wheeled vehicles, based on how many correct and incorrect values were in the predicted set. This scoring system is fully explained in Table 3.

Table 3. Possible Values for Scoring the Associator

# of Matches	Size of Proposed Set	Notes	Score
0	4	Four tires incorrectly associated - the Worst Case	1/6 = 0.14
0	3	Three separate tires incorrectly associated	1/7 = 0.17
0	2	Two tires incorrectly associated with each other	1/5 = 0.2
0	1	A single associated tire is effectively no association	1/4 = 0.25
1	4	One correct pair and two unrelated tires	2/6 = $0.\bar{3}$
1	3	One correct pair and a third misidentified tire	2/5 = 0.4
1	2	Two tires correctly associated with each other	2/4 = 0.5
2	4	Three tires that belong together and a fourth misidentified tire	3/5 = 0.6
2	3	Three tires correctly associated	3/4 = 0.75
3	4	Four tires correctly associated - the Ideal Case	4/4 = 1

A score of 0 is not possible because the algorithm does not generate identities for empty sets. The minimum value possible is related to the number of tires a vehicle may have. Future iterations that are able to account for vehicles with more than four TPMS-equipped tires would have more possible values, and the minimum value would follow the equation for n tires: $\frac{1}{2n-1}$. Evidence of this decreasing score appears as a practical reality in the TORI experiments, where a single vehicle has dozens of IDs associated with its identity, driving the minimum score to a far lower possible value.

3.7 Route Reconstruction

Algorithm.

The output from the Associator phase, and input to the Route Reconstruction phase, is a list of proposed vehicles and their associated observations, which are timestamped locations. The algorithm assumes full knowledge of the roads in the geographic area where detectors are placed, and this map is stored as a graph in which roads are edges and intersections are nodes. The edges are weighted based on the estimated travel time to traverse that edge. Each vehicle's observations are examined in order, and the algorithm attempts to predict the most likely route between observations for a given vehicle. To accomplish this, it takes two consecutive observations and finds the simple path (no loops) whose estimated travel time most resembles the difference in time between the points. This is repeated between all observations for a vehicle, such that a list of sparse observations becomes a complete route with no gaps. This entire process is repeated until proposed routes have been generated for all proposed vehicles. An overview of the algorithm is presented in Algorithm 3.

```
**Input:**
$G_{roads}$ = graph structure of roads
vehicles = sparse observations (timestamped locations) indexed by vehicle ID
**Output:**
vehicles is updated to contain full vehicle paths
for each path p in vehicles do
    $p_{full}$ = Empty Graph
    for Observation $x_i$ in $p$ do
        Add $x_{i,loc}$ to fullpath
        if $x_{i+1}$ exists then
            elapsed = $x_{i+1,time}$ - $x_{i,time}$
            $paths_{possible}$ = SimplePaths(G, $x_{1,loc}, x_{i+1,loc}$)
            $p_{best}$ = NearestTime($paths_{posible}$,elapsed)
            Add $p_{best}$ to $p_{full}$
        else
            break
        end
    end
    Update $p$ to $p_{best}$
end
```

Algorithm 3: Route Reconstruction Algorithm

Scoring.

Because the road network is modeled as a graph, a natural choice to quantify the correctness is graph edit distance. The truth data is used to build a directed graph containing only the nodes and edges actually traversed, which is then compared to the graph constructed by the algorithm. Graph edit distance tallies up the minimum number of insertions, deletions, or substitutions required to convert one graph to the other. In this case, to evaluate the proposed routes of proposed vehicles to the true routes of true vehicles. These operations can be weighted differently to reflect the cost that each one incurs. In this case, an insertion would mean that the algorithm missed a node or edge, and a deletion means it incorrectly guessed that a vehicle visited a node or edge.

3.8 Experiment Factors

To evaluate the tracking pipeline under different conditions, the sensor density and vehicle density are varied across simulations. Low, Medium, and High sensor density, corresponding to detectors placed at 10%, 50%, or 100% of intersections, will provide insight into the optimal number of sensors that should be placed when cost or infrastructure would be a consideration. Vehicle density is likewise set to Low, Medium, or High, corresponding to 200, 500, or 2000 vehicles in the simulation. These were manually determined based on how much traffic the map could reasonably contain without becoming overwhelmingly gridlocked. The Low Traffic Density is the lowest number of vehicles that still ensures that most of the map will be used. The Moderate setting was determined by the largest number of vehicles that would not add extra travel delays. The High Traffic Density setting is set as the largest number of vehicles the simulation can handle without experiencing total gridlock. These settings provide sufficiently differing driving conditions to test the algorithms used for tracking.

Each of the nine experiments was run with 30 unique seeds on the Route Reconstructor, varying the exact routes and detector placement while keeping the densities the same, while the Associator and TORI Associator were tested with 150 unique seeds per experiment due to less processing constraints. In the TORI Variable Prevalence experiments, vehicles were also assigned TORI technology with a probability ranging from 0-100% in 10% increments. That set of experiments were run with 25 unique seeds per experiment per prevalence. In the TORI experiments, for each of the nine simulation conditions. This reduction in seeds saved processing time, due to the tenfold increase in experiments that varying prevalence introduces.

3.9 Tracking Via Sensor Data

The existing problem that currently enables tracking via TPMS is the exposed, and unchanging, sensor IDs. The solution proposed in this document is primarily concerned with changing these IDs in some form of hopping sequence, while leaving the remainder of a packet unmodified to reduce potential costs and allow for more compatibility with existing systems such as typical shop tire readers. However, with more sophisticated algorithms it could be feasible to use the temperature and pressure data to create identities, and then routes, for vehicles. This can be solved by obfuscating this data, but it warrants investigation to discover how necessary such a modification would be. In [7] the authors calculated the likelihood of two or more cars having the same set of tire pressure IDs using the birthday problem calculation. Assuming a uniform 28 bit ID-space, they calculated that it would require more than 10^{15} vehicles on the road to have greater than a 1% chance of ID-set collision. This vast number, coupled with the fact that the vehicles would not be geographically co-located, means that the sensor IDs are good candidates for identifying vehicles. The inputs and assumptions to that calculation break down when considering the physical sensor data, rather than the identifiers. The sensors do not transmit precise data, and reserve a much smaller portion of the packet for the pressure and temperature data. During initial experiments, it was discovered that the Toyota TPMS sensors being tested reserved eight bits for the temperature and six bits for the pressure. This reduces the space for identifying a vehicle from 28 bits to 14 bits per tire. This alone reduces the number of vehicles required for a data collision to

$$R = \sqrt{\frac{2^{57}}{4!} ln(\frac{1}{1-P})}$$

In this scenario, the number of vehicles required for a 1% chance of data collision is roughly eight million. The problem is further exacerbated by the fact that during normal operation, only a small range actually varies, so the assumption of equal distribution is violated. The Toyota sensors encoded the data as integers with no decimal precision, in Celsius for temperature and psi for pressure. The only modification made to that data was to shift it so that negative values would not need to be considered. TPMS typically will not warn a driver unless a tire more than 25% different than specified by the manufacturer, so for a recommended pressure of 35 psi, it could go nine psi in either direction and still be within bounds. The typical values could then only have the entropy of five bits, versus the full six. Based on the results of [28], a temperature variation of 40°C or less during operation in an urban environment is reasonable. This reduces the entropy of the temperature field from 8 bits to 6 bits. The number of vehicles required for a data collision now becomes

$$R = \sqrt{\frac{2^{45}}{4!} ln(\frac{1}{1-P})}$$

which places the number of cars required for a 1% expected chance of collision on the order of 100,000 vehicles. This begins to cast doubt on the efficacy of tracking in congested environments using the pressure and temperature data available in the clear. The formulation here is optimistic for the attacker (getting data from all four tires is unlikely, and the true range of pressure and temperature is likely smaller in practice) and demonstrates that associating tires with identities using the sensor data alone becomes increasingly difficult as traffic density increases. The options, then, are to use more data to identify a vehicle, or shrink the effective traffic density by placing more sensors such that a smaller geographic area can be considered when combined with time data. It seems reasonable to conclude that the cost to an attacker is significantly increased by rendering the sensor IDs unusable, even if the sensor data is left unchanged, i.e., non-obfuscated.

3.10 TORI Experiment

A full implementation of TORI is not undertaken in the scope of this thesis; but with the infrastructure for the TPMS association constructed, it is straightforward to demonstrate the effectiveness of an ID hopping scheme. The post-processing step that generates TPMS packets is modified for these experiments such that every observation for a tire rolls the ID using SIMON, with the sensor's 64 bit secret key and the previous ID as input to generate a new 32 bit ID. These packets are then fed into the Associator created for the current generation of TPMS. This was expected to have a dramatic negative impact on the average scores for the Associator. This is due, in part, to some assumptions that are built in to the algorithm, but also to the greatly increased difficulty of association with hopping IDs. The goal of this subset of experiments is to show how the problem of association becomes significantly harder when basic security features are implemented.

The score is calculated using Jaccard Distance, but the sets that are compared are slightly modified. The proposed identity set is compared to the true vehicle's set of all IDs that were used in the simulation. The simulation conditions are for active collection, which stimulates packet transmission, and every transmission creates a new ID. Each tire then has at least as many associated IDs as observations, leading to potentially large sets of IDs being associated with one vehicle when including all four tires. The experiments were run with the Low, Moderate, and High Traffic Density and Low, Moderate, and High Detector Density conditions (nine combinations total) and 150 seeds per experiment. The results of these experiments are presented in

TPMS Collisions.

The potential exists, particularly in TORI, for sensors to transmit the same ID. If the devices transmit compatible packets, this is essentially an accidental spoofing attack. The impact of this varies depending on which type of device is transmitting and receiving the packet. These possible combinations are explored in the following sections.

Static-TPMS collision with Static-TPMS.

It is possible, though unlikely, for modern-day Static TPMS devices to transmit the same ID. Sensors from different manufacturers use differing frequencies, encoding schemes and packet structure, and even if those conditions were equivalent, there is enough ID space that a manufacturer can ensure that collisions do not naturally occur. It is certainly possible to generate a collision with spoofing, so the handling of a collision is still important with Static TPMS.

Rouf et al. tested how a vehicle reacted to spoofed packets. On their test vehicle, multiple error packets, spaced at least 225 ms apart and less than 4 seconds apart, were required to illuminate a warning light [7]. Additionally, once one error packet was detected, the vehicle sent activation signals to trigger more packets from that tire's sensor. They discovered that as long as at least one error packet was sent within a certain window, it didn't matter how many non-error packets were received. The timing constraints were relatively strict, however, such that natural collisions (such as those from nearby vehicles) are unlikely to have an impact, unless the activation signal also reached the colliding vehicle. Different manufacturers and models surely have different schemes and windows for dealing with these issues, but the lesson is that single error packets can already be handled in the current generation of TPMS, so unintentional collisions are not currently a concern.

TORI collision with Static TPMS.

If the rolling ID for a TORI device collides with a compatible Static TPMS device, the receiver for the Static TPMS vehicle will most likely discard the single packet, based on the work in [7]. Even if the vehicles are travelling closely, the next transmission of the TORI device will use a different ID, such that a collision is no longer present. The worst-case scenario is that the Static TPMS receiver accepts the single packet, with an impact of displaying erroneous data for up to two minutes.

Static-TPMS collision with TORI.

Suppose a vehicle equipped with TORI is driving near a vehicle with Static TPMS devices, and one of those devices transmits a compatible packet with an ID that matches the next expected ID in the rolling sequence. The receiver for the TORI vehicle will accept that packet as valid, but there is no potential for serious impact. As long as the TORI receiver requires multiple error packets to display a warning to the driver, there are no adverse effects except for the rejection of the forthcoming, valid packet. Filtering based on expected timing, received signal strength, or utilizing directional information from multiple antennas could alleviate this issue.

TORI collision with TORI.

TORI always has the potential for a collision when within range of other devices. This is very unlikely in a 32 bit space with low transmission power, as referenced in Section 2.3, and still unlikely even if ID space were reduced. The effect is essentially the same as when a Static TPMS device collides with a TORI expected ID. Because they are transmitting a different sequence, only the one packet will have a valid ID. Multiple warning packets are required to warn the driver, and non-critical data will either be ignored, averaged in with the recent valid packets, or at worst, displayed until the next valid packet arrives. A vehicle's own registered tires may even transmit colliding IDs, though this would be trivial to predict and detect within the receiver.

3.11 Summary

This chapter established the simulation setup and test methodology for the vehicle tracking experiments, and described an implementation of TORI. Using the features built into SUMO allows for rapid prototyping, upon which TPMS post-processing scripts were added to convert the data to a usable format. An Association algorithm was introduced to match individual TPMS packets from individual tires into a single vehicle identity, which the Route Reconstruction algorithm can turn into full paths for a vehicle, modelling the map and routes as a graph data structure.

IV. Results & Analysis

4.1 Introduction

This chapter presents analysis of the results of the experiment designed to examine the ability to track vehicles using Static TPMS. Those results are then presented in comparison to evaluate the effectiveness of Tire Obfuscation through Rolling ID (TORI), described in Chapter III. The first section analyzes the feasibility of tracking via Static TPMS. It is further divided into simulation results of the association phase and the route reconstruction phase. The next section investigates how vehicles equipped with TORI perform in the association phase of tracking. It is also evaluated on how it would perform against potential attacks that have been demonstrated or theorized against Static TPMS systems. The final set of experiments present results on the effects of combining Static TPMS vehicles and TORI vehicles, to examine the potential impact of a gradual rollout of this new technology.

4.2 Static TPMS Experimental Results

The variables changed between experiments are traffic and detector densities. These are set at Low, Medium, or High as discussed in Section 3.8.

Associator Results.

This section examines the results of testing the Associator. Recall that Jaccard distance provides a metric for the difference between the virtual identity proposed by the algorithm, and the true identity from the simulation. Because only vehicles with four tires are considered, there is a discrete number of possible values. These described in the case of four tires in Chapter III.

Figure 8 presents the boxplots of each experimental condition, when run with 150 seeds. Trends are apparent when grouping the results by traffic density, which is how the remainder of the section will be organized. The horizontal axis categorizes each experiment by Traffic Density and Detector Density. The vertical axis ranges from zero to one and represents the Jaccard Distance across that set of experiments. Higher values are better, with one being the ideal case.

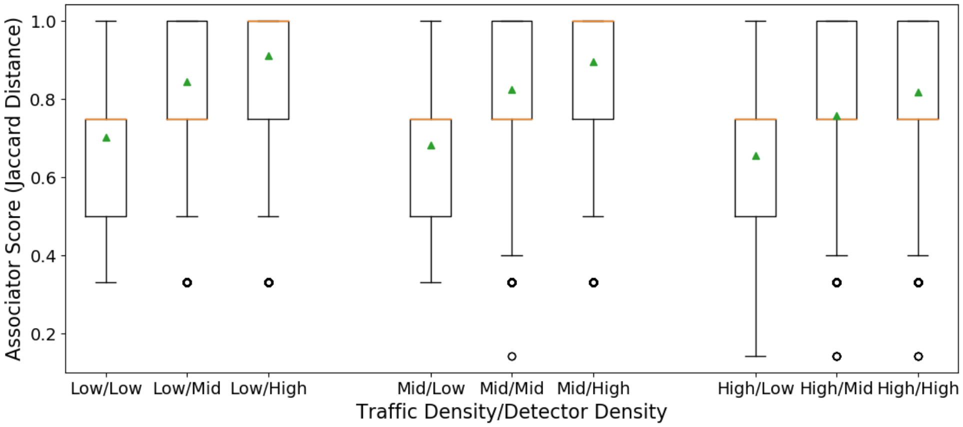


Figure 8. Associator Results - Static TPMS - All Experiments

Triangles represent the mean of the data, and outliers are drawn as circles. The median is represented as a red horizontal line, which always overlaps a quartile boundary due to the discrete number of values possible. This figure shows an overview of the trends across every experimental condition. Increasing the number of detectors always improves the mean score, however the amount by which it increases is not constant. The Low detector density experiments have a very similar average, regardless of Traffic Density, but the range of scores gets wider in the High Traffic Density experiment. The efficacy of adding detectors is reduced as Traffic Density increases, demonstrating a diminishing returns effect and hinting at a critical point in Traffic

Density that could allow for optimizations for a known set of traffic conditions. This data is explained in deeper detail in the following sections.

Low Traffic Density.

Low traffic density presents the most ideal conditions for the Associator to work. In the most simple case, a single vehicle could always be correctly associated because it would not require distinguishing between vehicles. It also presents the largest gains when increasing the number of detectors. Figure 9 charts the distribution of scores for each of the Low Traffic Density conditions. These experiments were run with 200 vehicles injected into the simulation over 200 seconds.

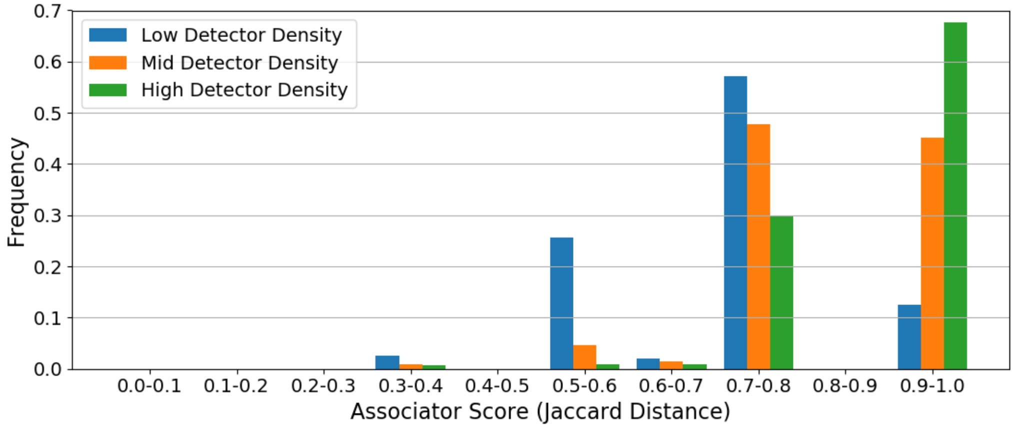


Figure 9. Low Traffic Density Associator Score Distribution

Note that the distribution for each Detector Density does not have a continuous shape because certain values are less likely to be output. The algorithm is more likely to select smaller, incomplete sets than to include tires that are less confidently associated. This means that values such as 0.5 and 0.75 are more likely to occur because they represent incomplete sets with no mismatches, versus a value such as 0.6, which has one mismatched tire. This is a form of risk aversion, and could be adjusted by using different evaluation metrics when the algorithm is making decision.

As detector density increases, the weight of the distribution rapidly shifts to the right, as more perfect matches occur. This is also observable in the means of the data, which aids in smoothing the discrete nature of scores. The mean values for this data with a 95% confidence interval are shown in Table 4.

Table 4. Mean Jaccard Distance for Low Traffic Density Associator Experiments

Detector Density	Mean Jaccard Distance
Low	0.70 ± 0.0023
Moderate	0.85 ± 0.0017
High	0.91 ± 0.0015

There is a 0.15 increase between the Low Detector Density (10% intersection coverage) to Moderate Detector Density (50% intersection coverage), and a 0.06 increase between Moderate and High (100% coverage). The histogram plots show less variation as Detector Density increased, which is reflected here in the smaller confidence interval with higher detector levels.

Moderate Traffic Density.

These experiments were conducted with 500 vehicles injected into the simulation over 200 seconds. This represents over twice as many vehicles as the Low Traffic Density experiment, but still allows for fairly normal traffic patterns. Figure 10 shows the distribution of scores across the Moderate Traffic Density Experiments.

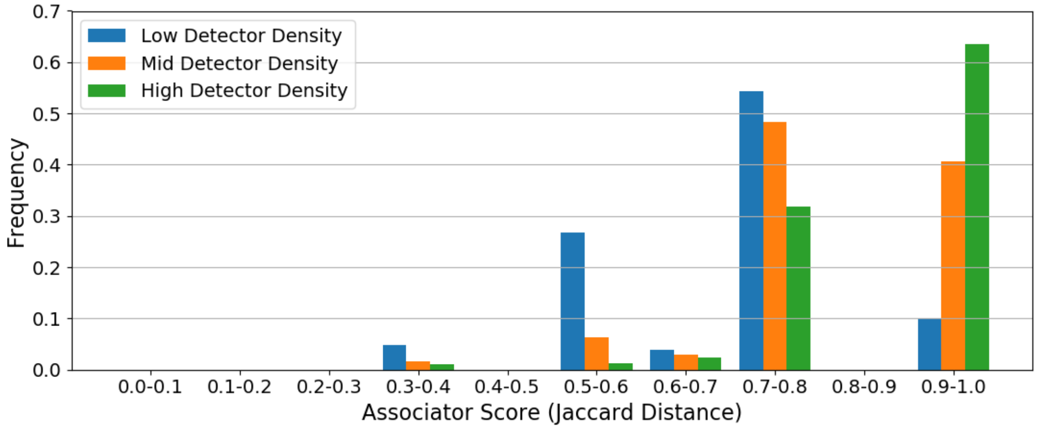


Figure 10. Moderate Traffic Density Associator Score Distribution

These graphs hold the same trend as before, with the distribution shifting to the right as detector density increases. The mean values for this set of experiments with a 95% confidence interval are shown in Table 5.

Table 5. Mean Jaccard Distance for Moderate Traffic Density Associator Experiments

Detector Density	Mean Jaccard Distance
Low	0.68 ± 0.0015
Moderate	0.82 ± 0.0013
High	0.90 ± 0.0011

In this set of experiments, the mean increased by 0.14 from Low to Moderate, and 0.08 from the Moderate to High case, a slight improvement in efficacy over the Low Traffic Density experiments. This is appears to start a slight trend, where as the difficulty of the problem increases, there is more to be gained from adding detectors.

High Traffic Density.

These experiments were run with 2000 vehicles injected into the simulation over 200 seconds. This created scenarios with occasional gridlock as the traffic system attempted to cope with the large influx of vehicles. This helps to provide perspective on how such a system would perform in periods of heavy traffic, such as rush hour times or days of special events. Figure 11 shows the score distribution for this set of experiments.

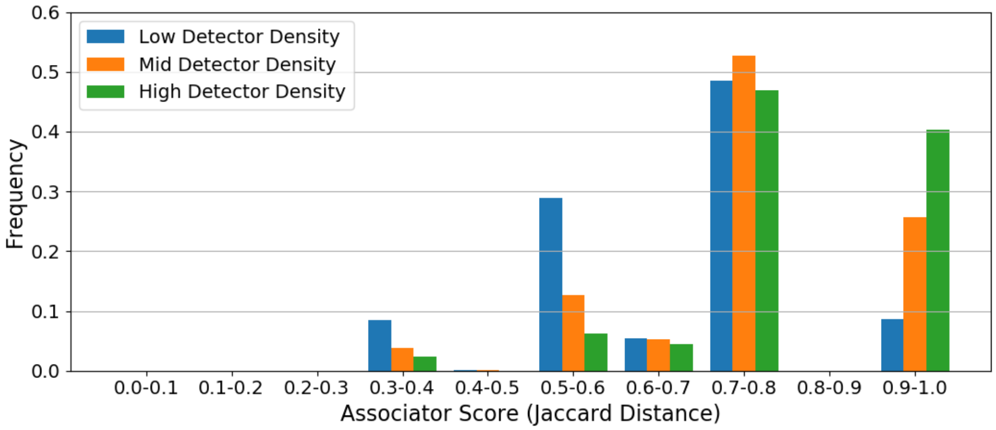


Figure 11. High Traffic Density Associator Score Distribution

The same trend as previous experiments is present, with higher scores appearing more frequently as detector density increases, but the relation is not as strong as before. This can be seen explicitly in the means in Table 6.

Table 6. Mean Jaccard Distance for High Traffic Density Associator Experiments

Detector Density	Mean Jaccard Distance
Low	0.66 ± 0.00095
Moderate	0.76 ± 0.0011
High	0.82 ± 0.00064

In this set of experiments the mean increased by 0.10 from Low to Moderate, and 0.06 from Moderate to High. This has a lower impact in both cases than the Moderate experiments, however it does demonstrate a slight increase in efficacy when going from Moderate to High compared to the Low Traffic Density Experiments. This points to there being an optimization problem to be explored in this data, testing more Detector Densities to see where the optimal point is for different levels of Traffic Density. This knowledge would allow someone wishing to implement a tracking system to set a desired cost per improvement, and deploy the optimal number of sensors.

There are certain trends that span more than just the traffic densities. Increasing detector density always improves the mean score, although the relative benefit of adding more detectors is not equivalent at each traffic density. All experimental conditions had cases where all four tires were correctly associated, which is to be expected as long as the routes are long enough and there are opportunities to detect all four tires. The minimum score present in the data was tied to traffic density rather than detector density, where the worst cases get worse for a given detector density when more cars are injected into the simulation. This is likely the result of traffic congestion, which causes vehicles to group at intersections and effectively form caravans. Caravans make association difficult, as multiple vehicles passing by a detector in a short window increases the likelihood of an error.

Route Reconstruction Results.

This section presents the results of the route reconstruction experiments on Static TPMS. The Route Reconstructor is scored based on the Graph Edit Distance between the true route a vehicle traveled, and the estimated route generated by the algorithm. Every node or edge that is inserted or deleted incurs a cost of one, such that a zero is a perfect reconstruction. Note that because routes are random, the length of each route varies, and the number of nodes or edges traveled does not have a constant relation to the physical distance traveled. As such, graph edit distance is used to observe trends between experiments as a natural metric to use when the maps are modeled as directed graphs. Figure 12 illustrates the distribution of scores for the Route Reconstructor. Trends in route reconstruction emerge primarily as a function of detector density, as expected. When detector density is high, some vehicle routes are perfectly recreated. Under the conditions of low vehicle density and a high number of detectors, the average score is the best of any of the experiments conducted. However, when vehicle density is high, adding more detectors does not improve the average score.

It is plausible and intuitive to expect that the algorithm would reconstruct routes with complete accuracy when 100% of intersections are monitored, as in the High Detector Density case. This does not occur in practice for multiple reasons. First is that the Route Reconstruction algorithm is not given knowledge of the detector coverage; if provided, it could have heuristics to choose paths that are more likely. Another factor is the imperfect reconstruction of identities and their routes from the Association phase. Complete detector coverage still does not detect every single tire at every intersection. When an identity was formed, it may have selected a subset of tires that do not have a route that is completely covered, thus is functionally equivalent to having a lower Detector Density. Finally, the metric used to decide on the most likely path travelled between observations is expected travel time. The farther actual travel times are from expected, the more errors that may be introduced, so in heavily congested traffic the algorithm is prone to error. Future iterations of this algorithm could use knowledge of the map and detector locations to inform the decision points, and use a changing expected travel time metric that is updated to reflect true conditions.

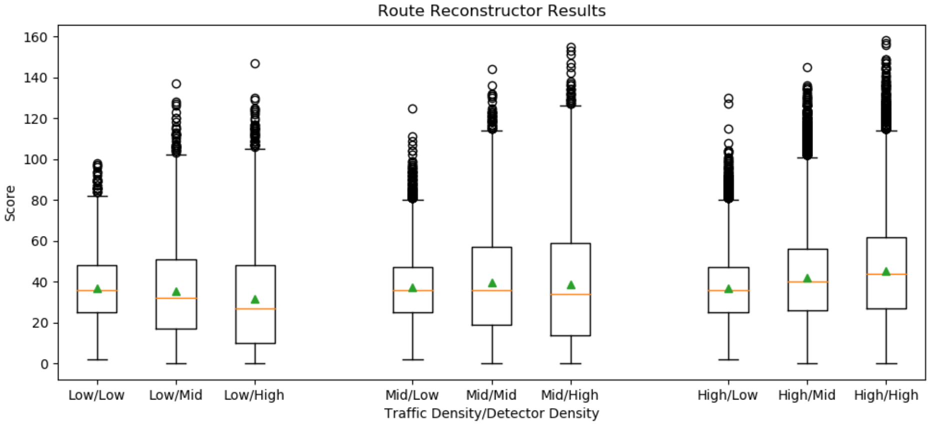


Figure 12. Route Reconstructor Edit Distance All Experiments

Low Traffic Density.

This section presents the route reconstruction results for Low Traffic Density and varying detector density. The mean values with 95% confidence intervals are presented in Table 7.

Table 7. Mean Graph Edit Distance for Low Traffic Density Route Reconstruction Experiments

Detector Density	Mean Value
Low	36.93 ± 0.4941
Moderate	35.62 ± 0.5952
High	31.53 ± 0.6460

The trend across these experiments is a steadily decreasing average Graph Edit Distance, improving by 3.55% from Low to Moderate and 11.5% from Moderate to High. The confidence interval showed the most variability in the high case, but remained under one edit in every case. Figures 13 show the distribution of scores in each experiment across several seeds. Figures 14 show the distribution of the Edits Per Node Traveled Ratio in each experiment.

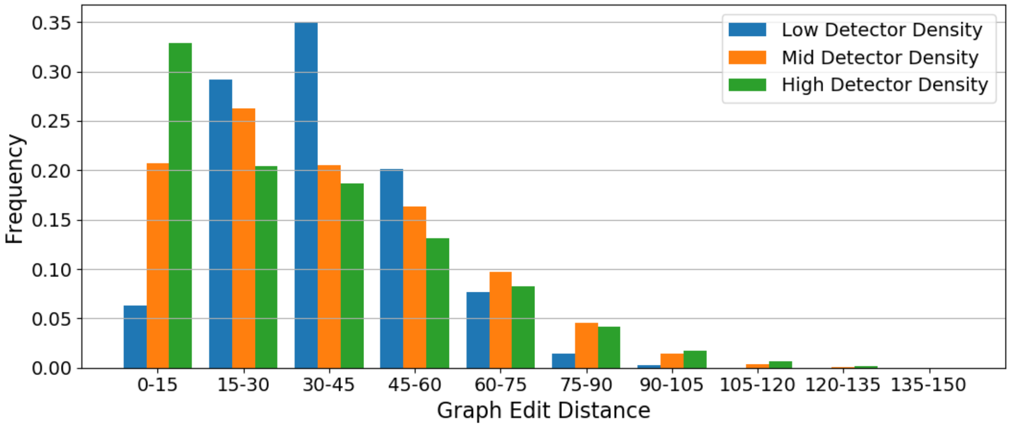


Figure 13. Low Traffic Density Graph Edit Distance Distribution

Figure 13 demonstrates the expected trend in the distribution of scores. Note that lower Graph Edit Distance is better, denoting fewer differences between a proposed and true vehicle route. With Low Detector Density, it shows a relatively Normal distribution centered around 30-45 edits. This distribution shifts to the left as Detector Density increases, representing a higher percentage of reconstructed routes with a low edit distance.

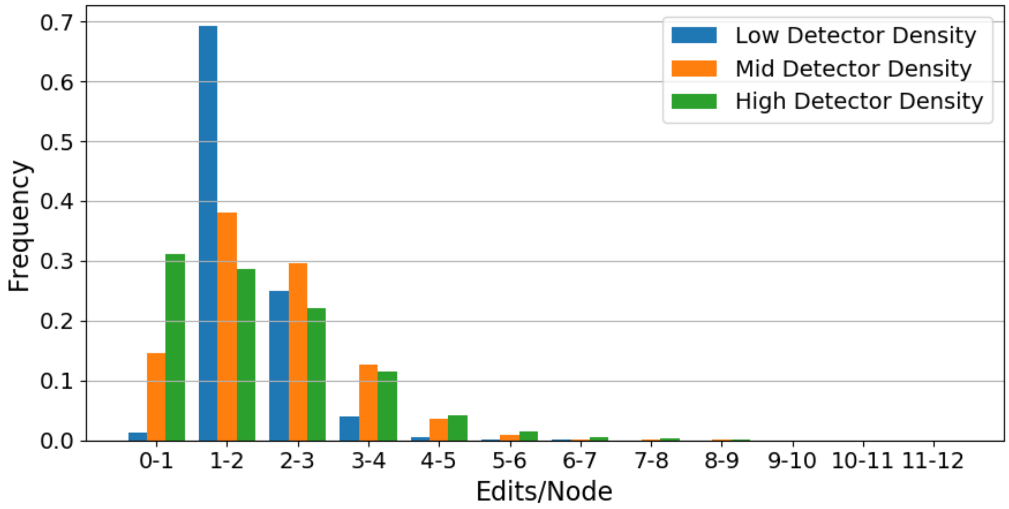


Figure 14. Low Traffic Density Edit Ratio Distribution

Figure 14 shows the distribution of Edit Ratios for the Low Traffic Density experiments. Recall that Edit Ratio represents the number of edits made to a route divided by the total number of nodes traveled on the true route. This helps account for long routes, where a higher number of absolute errors would be expected. The ideal case is to score a zero, representing a perfectly reconstructed route. The Low Detector Density experiment shows a strong peak centered around one to two edits per node, but as Detector Density increases, the distribution flattens into a more normal distribution and shifts left.

Moderate Traffic Density.

This section presents the route reconstruction results for Moderate Traffic Density and varying detector density. The mean values with 95% confidence intervals are presented in Table 8.

Table 8. Mean Graph Edit Distance for Moderate Traffic Density Route Reconstruction Experiments

Detector Density	Mean Value
Low	37.17 ± 0.3211
Moderate	39.74 ± 0.4037
High	38.92 ± 0.4654

This set of experiments did not demonstrate a trend, increasing by 6.9% from Low to Moderate, but decreasing 2.07% from Moderate to High. The confidence interval did increase as the number of detectors increased, demonstrating increased variability. This may point to an area for exploration, where there may be a critical point for each Traffic Density where more detectors can actually hinder average accuracy.

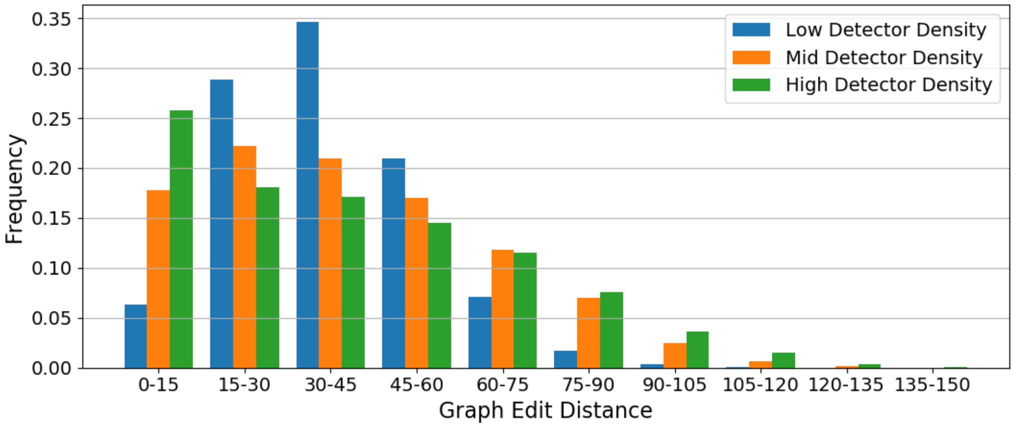


Figure 15. Moderate Traffic Density Graph Edit Distance Distribution

Figure 15 shows the distribution of edit distance values for the Moderate Traffic Density experiments. The Low Detector Density case shows a very similar distribution to the Low/Low experiment, a normal distribution centered around 30-45 edits. It also continues the trend of shifting the distribution left (improving in accuracy) as Detector Density increases.

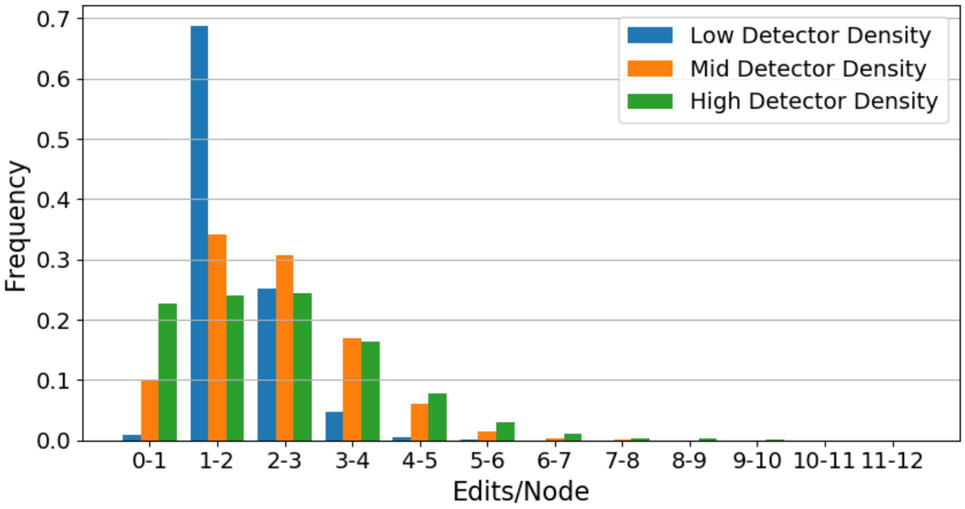


Figure 16. Moderate Traffic Density Edit Ratio Distribution

Figure 16 shows the distribution of Edit Ratios for the Moderate Traffic Density experiments. The Low Detector Density experiment shows a strong peak centered around one to two edits per node, as was also observed in the Low/Low experiment. Increasing the number of detectors reduces the peak and shifts the distribution towards zero edits. Performance is diminished overall when compared to the Low Traffic Density experiment, with no experiment having the lowest (best) bin contain the peak frequency.

High Traffic Density.

This section presents the route reconstruction results for High Traffic Density and varying detector density. The mean values with 95% confidence intervals are presented in Table 9.

Table 9. Mean Graph Edit Distance for High Traffic Density Route Reconstruction Experiments

Detector Density	Mean Value
Low	36.86 ± 0.1868
Moderate	42.18 ± 0.1904
High	45.27 ± 0.2087

The trend across these experiments is an increasing average edit distance, increasing by 14.4% from Low to Moderate and 7.32% from Moderate to high. This continues the trend from the Moderate Traffic Density experiment, with the Route Reconstructor performing worse as the number of detectors increases. The confidence interval remained under one in every case, and is more consistent than in the previous set of experiments. Figure 17 show the distribution of scores in each experiment across several seeds. Figure 18 shows the distribution of the Edits Per Node traveled Ratio in each experiment.

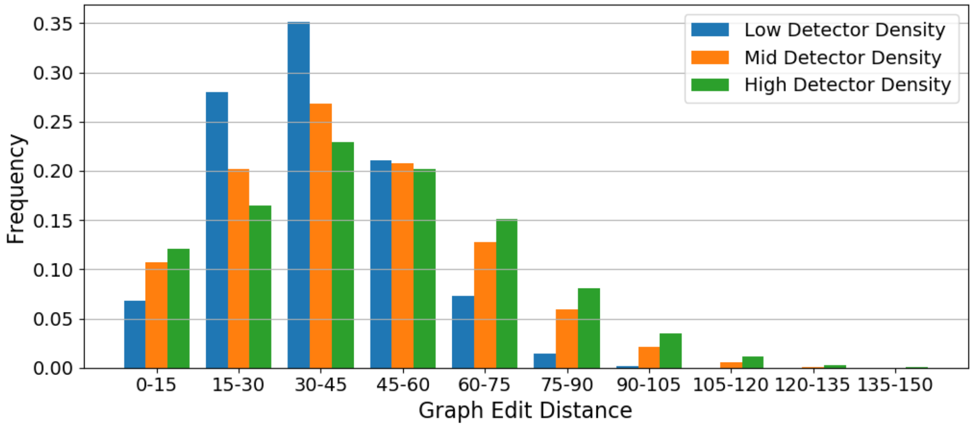


Figure 17. High Traffic Density Graph Edit Distance Distribution

Figure 17 shows the distribution of edit distance values for the High Traffic Density experiments. These did not demonstrate the same strong left shift as Detector

Density increased, as was observed in the lower Traffic Density experiments. This set of experiments showed a flattening effect, with the peaks getting weaker and the distribution spreading. They are still heavy towards the left, demonstrating that the algorithm is still producing better results with more detectors, but this effect is not nearly as pronounced as the previous cases. This points to a scaling issue with the simulation, detection or reconstruction, where there is a critical point in Traffic Density where the efficacy of the algorithm decreases.

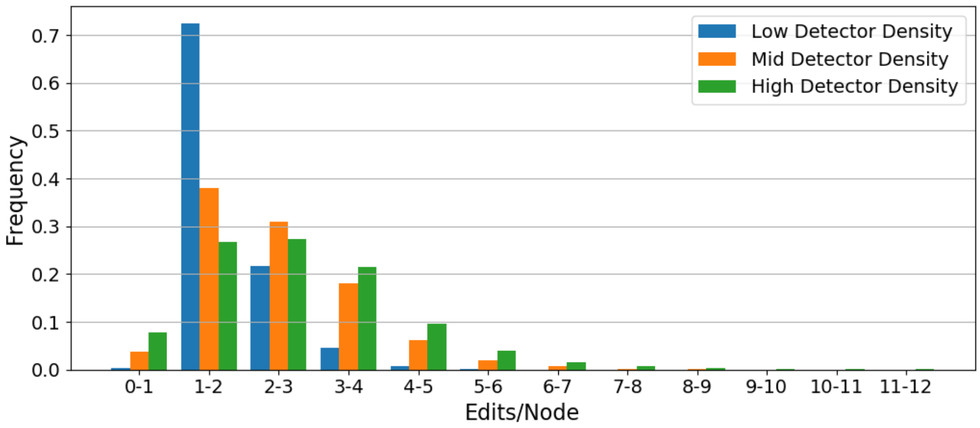


Figure 18. High Traffic Density Edit Ratio Distribution

Figure 18 shows the distribution of Edit Ratios for the High Traffic Density experiments. The Low Detector Density experiment shows a strong peak centered around one to two edits per node, as was also observed in the previous experiments. Increasing the number of detectors no longer improves the results as much as previous experiments, still peaking at one to two edits per node but with more weight towards the higher numbers at High Detector Density, representing worse performance.

4.3 TORI

Associator Results.

An additional set of experiments were run to demonstrate how a security implementation that incorporates rolling tire IDs would disable the Associator that was working effectively for Static TPMS. An overview of the statistical distribution over 150 seeds for each condition is shown in Figure 19. The horizontal lines that make up each plot represent the minimum, first quartile, median, third quartile, and maximum (not including outliers). Outliers are plotted as circles, while the triangle represents the mean for that experiment.

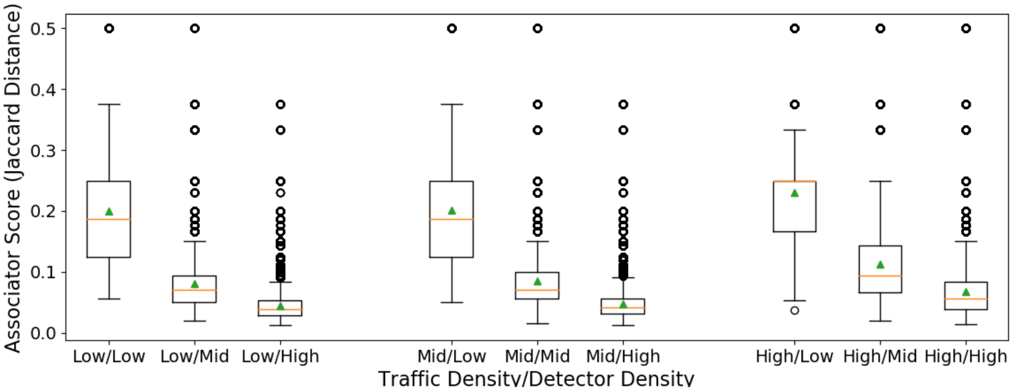


Figure 19. TORI Associator Results

As expected, the results are poor in comparison to Static TPMS results. Rolling IDs dramatically reduce the effectiveness of an Associator designed for the static TPMS devices, and would require more effort and more complex algorithms to associate, if possible at all. The means are nearly equal for a given detector density, regardless of the traffic density. Outliers exist due to short vehicle routes, where a vehicle may have been rarely observed, and therefore generated fewer IDs during the simulation. The fewer IDs that belong to a true vehicle, the more opportunity exists for a proposed identity to receive a high score. In these experiments, detector coverage is inversely related to the performance. This will be explored in greater detail in the following sections, but appears to be due to the increase in IDs that belong to a true vehicle as detectors stimulate more packets. Recall that this simulation is for active detection, which stimulates the sensors to broadcast every time the vehicle passes a detector. When a proposed identity is scored, that score is proportional to how many IDs the algorithm correctly assigned to a vehicle, divided by the number of IDs that actually belonged to the true vehicle. In the previous experiments, each vehicle had four tire IDs assigned, so a perfect association would assign those four tires to one identity, and score a perfect 1.0 using Jaccard Similarity. Using the same system for scoring a TORI equipped vehicle is expected to generate a lower score in nearly all cases, because even in a scenario in which four IDs were assigned an identity and belonged to the same true vehicle, that vehicle could have dozens of IDs that belong to it. Every detector it drives by adds two to four more IDs, depending on how many sensors it stimulates. The end result is that the algorithm splits a single vehicle into many low-scoring identities, and the more IDs a vehicle generates during a trip, the lower those scores will be. Rolling IDs inject so much digital chaff into the Associator that the overall score greatly suffers.

Low Traffic Density.

The mean results of TORI association during Low Traffic Density, with a 95% confidence interval are shown in Table 10. The table also shows the percentage change when compared to the same experiment conditions for Static TPMS. The score distributions for this set of experiments are shown in Figure 20.

Table 10. Mean Jaccard Distance for Low Traffic Density TORI Associator Experiments

Detector Density	Mean Value	Reduction From Static TPMS Experiment
Low	0.20 ±0.00090	71.5%
Moderate	0.081 ± 0.00022	90.4%
High	0.045 ± 0.000087	95.1%

The average score for the TORI association is reduced by 0.119 from Low to Moderate, and by 0.036 from Moderate to High. The mean score trends towards zero as detector density increases. Though Detector Density is at its functional maximum on the High setting (100% intersection coverage), if more detectors were added, this score could reasonably be expected to decrease even further. In a practical implementation adding detectors in areas besides intersections would only serve to gain higher resolution for tracking vehicles between intersections. While beneficial for Static TPMS, during active collection this triggers more TORI packets, in turn confusing the Associator.

The "Reduction From Static TPMS Experiment" column of the table shows the relative decrease in mean Jaccard Distance, when compared to the same experiment conditions tested on Static TPMS vehicles. This set contains the largest score reduction of all TORI experiments, in the Low/High experiment, which received an average score 95.1% lower than the experiments with the same conditions, but tested on the insecure Static TPMS.

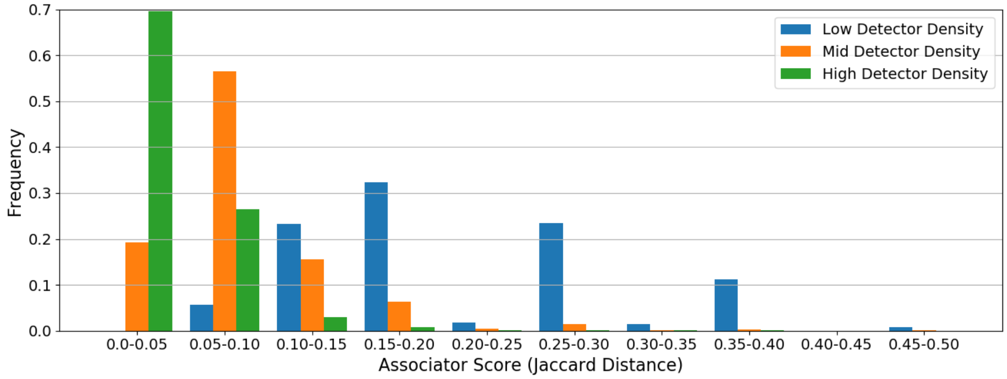


Figure 20. Low Traffic Density TORI Associator Score Distribution

The Low Traffic Density experiment has a moderately spread distribution of scores, which rapidly shifts towards zero as detector coverage increases. In the High Detector Density experiment, over 90% of the proposed identities from the Associator scored less than 0.1. This means that even with complete intersection coverage, which should be the best case, over 90% of the identities covered less than 10% of the IDs actually associated with a vehicle.

Moderate Traffic Density.

The mean results of TORI association during Moderate Traffic Density, with a 95% confidence interval are shown in Table 11. The table also shows the percentage change when compared to the same experiment conditions for Static TPMS. The score distributions for this set of experiments are shown in Figure 21.

Table 11. Mean Jaccard Distance for Moderate Traffic Density TORI Associator Experiments

Detector Density	Mean Value	Reduction From Static TPMS Experiment
Low	0.20 ±0.00060	70.4%
Moderate	0.085 ± 0.00015	89.7%
High	0.047 ± 0.000060	94.7%

The data shows a sharp decrease when compared to the earlier Static TPMS experiments, and rapidly decreasing as Detector Density increases. The average score is reduced by 0.115 from Low to Moderate, and by 0.038 from Moderate to High. These reductions are very similar to the Low Traffic Density experiment, showing that scaling for the scores between experiments holds constant, at least for these Traffic Densities.

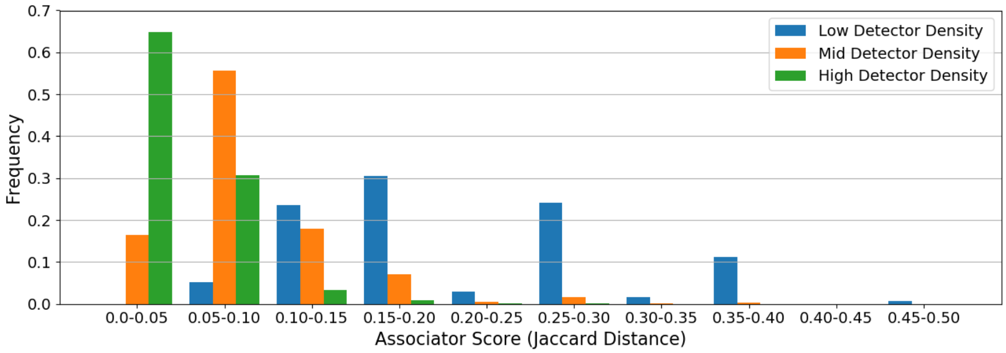


Figure 21. Moderate Traffic Density TORI Associator Score Distribution

The Moderate Traffic Density experiment showed the same trends as the Low Detector Density experiments, with a high percentage of the results being less than 0.1. The same summary statement applies to these Moderate experiments, as even at full intersection coverage, over 90% of the identities covered less than 10% of the IDs actually associated with a vehicle.

High Traffic Density.

The results of TORI association during High Traffic Density are shown in Table 12. The score distributions for this set of experiments are shown in Figure 22.

Table 12. Mean Jaccard Distance for High Traffic Density TORI Associator Experiments

Detector Density	Mean Value	Reduction From Static TPMS Experiment
Low	0.23 ±0.00043	65.9%
Moderate	0.11 ± 0.00014	85.1%
High	0.068 ± 0.000070	91.6%

This data also shows a dramatic drop in Associator Performance, worsening as detector coverage increases. The average score is reduced by 0.12 from Low to Moderate Detector Density, and by 0.042 from Moderate to High. These relative scores, while significant, are not as high as the previous experiments. The comparison experiment experienced similar diminishing returns, so this result is not surprising.

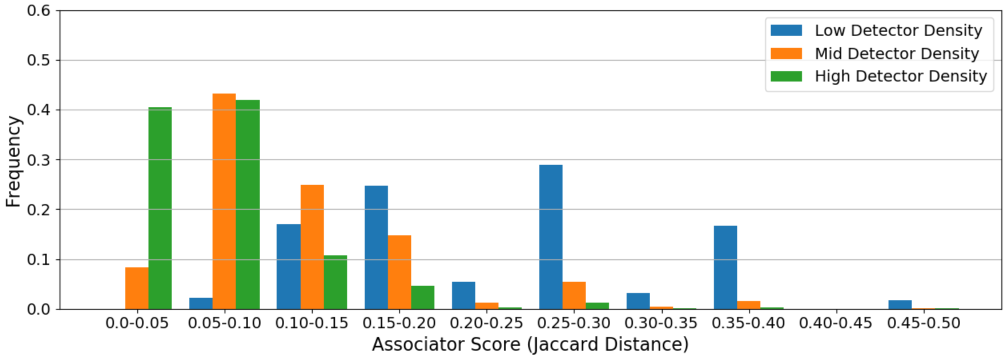


Figure 22. High Traffic Density TORI Associator Score Distribution

The High Traffic Density experiment showed the same trends as the previous experiments, shifting left as detector density increased. One trend to note across experiments is that in the Moderate Detector Density is that as Traffic Density increases, the weight of the distribution shifts towards 0.1. This may point to a critical point Traffic and Detector Density where scores may be slightly improved, or the rapid downward trend of the scores may slow. Though the shapes of the distribution very closely resembles the previous experiments, the peak frequency gets lower as Traffic Density increases, such that the scores do not decrease at the same rate or to the same degree in High Traffic scenarios versus others.

Variable TORI Prevalence.

These experiments sought to examine how mixing TORI-equipped vehicles with Static TPMS vehicles affects association of the Static TPMS vehicles. Experiments were run for Low/Moderate/High Traffic Density, Low/Moderate/High Detector Density, and 0-100% TORI Prevalence in 10% increments. Each simulation condition was executed with 25 different seeds and averaged. The individual scores were separated based on whether they mapped to vehicles with TORI technology or not. The results are grouped by Traffic Density, containing a table of the means in each experiment, which is also graphed. Note that the line graphs remove the point that sits on the x-axis for the sake of continuity, as these represent experiments with zero cars of that particular TPMS technology.

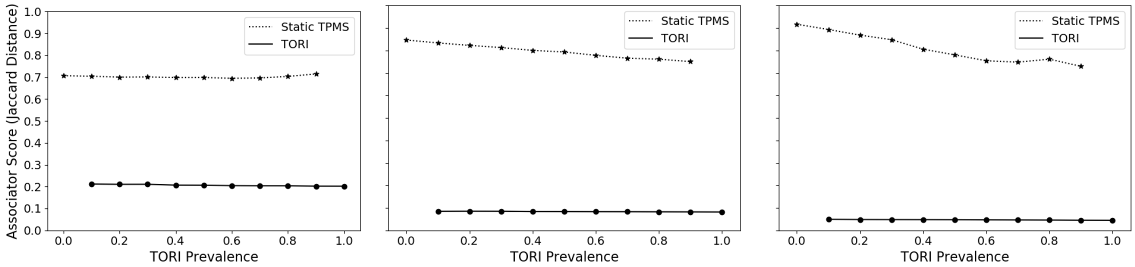
Low Traffic Density.

This section presents the results for the Variable TORI Prevalence experiments with Low Traffic Density. Table 13 presents the average means over 25 seeds for each experimental condition, which are graphed in Figure 23.

Table 13. Low Traffic Density Variable TORI Prevalence Mean Associator Scores

	Low Detector Density		Moderate Detector Density		High Detector Density	
Prevalence	Static TPMS	TORI	Static TPMS	TORI	Static TPMS	TORI
0	0.7065	0	0.8468	0	0.9170	0
0.1	0.7041	0.2116	0.8339	0.08472	0.8937	0.04924
0.2	0.7000	0.2103	0.8229	0.08520	0.8687	0.04821
0.3	0.7008	0.2105	0.8133	0.08495	0.8483	0.04791
0.4	0.6982	0.2067	0.8007	0.08375	0.8060	0.04769
0.5	0.6984	0.2064	0.7936	0.08354	0.7808	0.04736
0.6	0.6946	0.2043	0.7787	0.08317	0.7545	0.04675
0.7	0.6963	0.2035	0.7660	0.08296	0.7484	0.04632
0.8	0.7032	0.2034	0.7616	0.08250	0.7617	0.04584
0.9	0.7146	0.2020	0.7508	0.08214	0.7297	0.04508
1	0	0.2018	0	0.0817	0	0.04474

This set of experiments demonstrated a 1.1% increase, 11.3% decrease, and 20.4% decrease in TPMS association averages for Low, Medium, and High Detector Density as TORI prevalence increased from 0 to 90% of the vehicles in the simulation. The average score for TORI association held fairly constant within any simulation, but as Detector Density increased that score dropped significantly. This is thought to be the result of the detectors stimulating more packets from the TORI devices, which causes scores to lower as the vehicle generates more IDs.



(a) Low/Low

(b) Low/Mid

(c) Low/High

Figure 23. Low Traffic Density TORI Associator Score Variable Prevalence Distribution

As expected from the results in Section 4.2, increasing detector coverage increases the starting average for Static TPMS. The Low/Low experiment holds scores relatively constant for any prevalence.

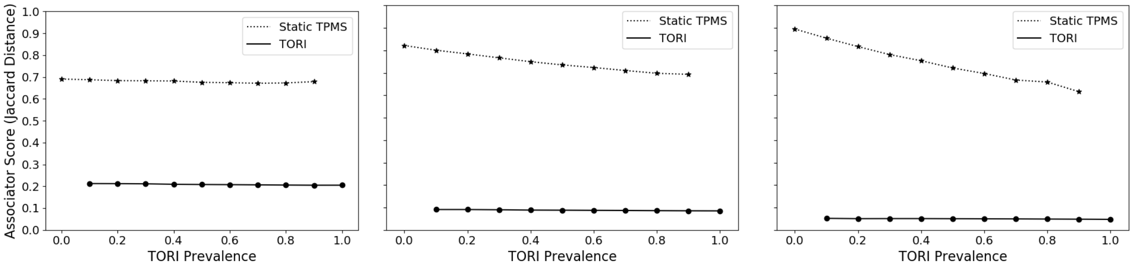
Moderate Traffic Density.

This section presents the results for the Variable TORI Prevalence experiments with Moderate Traffic Density. Table 14 presents the average means over 25 seeds for each experimental condition, which are graphed in Figure 24.

Table 14. Moderate Traffic Density Variable TORI Prevalence Mean Associator Scores

	Low Detector Density		Moderate Detector Density		High Detector Density	
Prevalence	Static TPMS	TORI	Static TPMS	TORI	Static TPMS	TORI
0	0.6904	0	0.8220	0	0.8954	0
0.1	0.6874	0.2123	0.8010	0.09097	0.8547	0.05176
0.2	0.6833	0.2119	0.7846	0.09095	0.8171	0.05024
0.3	0.6824	0.2112	0.7670	0.08988	0.7816	0.05059
0.4	0.6816	0.2089	0.7499	0.08848	0.7541	0.05057
0.5	0.6755	0.2079	0.7356	0.08804	0.7218	0.05005
0.6	0.6738	0.2073	0.7239	0.08733	0.6973	0.04960
0.7	0.6713	0.2063	0.7106	0.08678	0.6681	0.04922
0.8	0.6727	0.2054	0.6981	0.08603	0.6591	0.04856
0.9	0.6787	0.2047	0.6935	0.08537	0.6161	0.04792
1	0	0.2048	0	0.08497	0	0.04703

This Moderate Traffic Density experiments demonstrated a 1.7%, 15.6%, and 31.2% decrease in TPMS association averages for Low, Medium, and High Detector Density as TORI prevalence increased from 0 to 90% of the vehicles in the simulation. The same trend for the average TORI score as the Low Traffic Density experiments was present, with the score staying constant for a given Detector Density but dropping significantly as detector coverage increased. This data is shown graphically in Figure 24.



(a) Mid/Low

(b) Mid/Mid

(c) Mid/High

Figure 24. Moderate Traffic Density TORI Associator Score Variable Prevalence Distribution

These graphs illustrate the trend of a steady TORI score within an experiment, decreasing as Detector Density increases. The slope of the Static TPMS score decreases more rapidly in High Detector Density experiments, starting the highest score in this set of experiments and ending with the lowest score for Moderate Traffic Density Static TPMS.

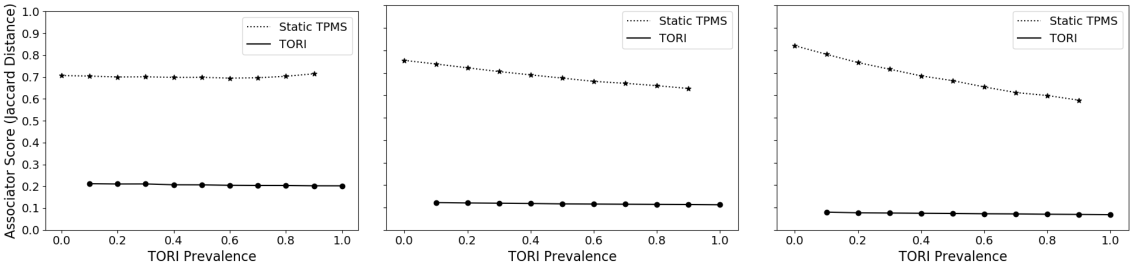
High Traffic Density.

This section presents the results for the Variable TORI Prevalence experiments with Low Traffic Density. Table 15 presents the average means over 25 seeds for each experimental condition, which are graphed in Figure 25.

Table 15. **High Traffic Density Variable TORI Prevalence Mean Associator Scores**

	Low Detector Density		Moderate Detector Density		High Detector Density	
Prevalence	Static TPMS	TORI	Static TPMS	TORI	Static TPMS	TORI
0	0.6557	0	0.7558	0	0.8211	0
0.1	0.6533	0.2394	0.7396	0.1220	0.7827	0.07948
0.2	0.6499	0.2423	0.7225	0.1203	0.7460	0.07622
0.3	0.6464	0.2413	0.7059	0.1192	0.7165	0.07540
0.4	0.6439	0.2386	0.6911	0.1179	0.6866	0.07439
0.5	0.6403	0.2378	0.6767	0.1161	0.6654	0.07344
0.6	0.6413	0.2351	0.6622	0.1154	0.6373	0.07210
0.7	0.6392	0.2344	0.6535	0.1147	0.6124	0.07140
0.8	0.6374	0.2334	0.6430	0.1140	0.5990	0.07005
0.9	0.6433	0.2321	0.6305	0.1133	0.5783	0.06921
1	0	0.2317	0	0.1122	0	0.06801

These experiments demonstrated a 1.9% , 16.6%, and 29.6% decrease in TPMS association averages for Low, Medium, and High Detector Density as TORI prevalence increased from 0 to 90% of the vehicles in the simulation. These returns are similar to those of the Moderate Traffic Density experiments, pointing towards a cap on the returns of TORI with this infrastructure.



(a) High/Low (b) High/Mid (c) High/High

Figure 25. High Traffic Density TORI Associator Score Variable Prevalence Distribution

The graphs again show a decreasing score for Static TPMS as TORI prevalence increases, with the slope becoming steeper with higher Detector Density. This trend was present for all Traffic Densities, and suggests that the more investment an eavesdropper puts in their infrastructure (deploying more detectors), the more potential TORI has to spoil the gains. This has the dual effect of lowering the effectiveness of eavesdropping and association in general, and requiring an increased effort that makes TPMS a less appealing attack vector.

Route Reconstruction.

The ability of the Route Reconstruction algorithm to generate quality results is highly dependent on the quality of the input. In the Static TPMS experiments, the input was nearly perfect as most of the identities for vehicles had consistent and realistic paths. Even with this near-ideal input, the results of converting sparse detections into complete paths had varying success. Integrating TORI into the simulation dramatically reduces the quality of the Associator data that would become the input to the Route Reconstructor. As one would expect, most proposed identities only contain one observation, from which a route cannot be constructed. Due to this, route reconstruction experiments were not run for TORI-equippped vehicles.

Resistance to Attack.

The purpose of this section is to note how TORI combats the attacks that work against the existing system, and discuss potential attacks on the proposed system. The goal of TORI was to significantly raise the bar of executing a successful attack against the devices. Any security features that are implemented significantly improve the difficulty of exploitation.

Denial of Service.

Denial of service is a threat in this space primarily because these devices exist for the safety of drivers, so lack of access to that data is a risk. Simple jamming of the frequency is trivial for all current systems, particularly because low power transmission is part of the design, and is considered a separate problem domain. In the proposed system, denial could potentially be achieved by filling the receiver's packet buffer with spoofed packets, such that it could never synchronize to the registered tires. To thwart this, a packet filter could be implemented using a feature such as signal strength, or dropping packets that don't have an expected ID while the receiver is in a trusted state with its sensors.

Spoofing.

TORI defeats the previous (trivial) methods of spoofing because an attacker cannot predict what the next packet will be without the secret key. Even if the spoofer happened to guess one correct ID, they would need to do so multiple times in the proper sequence for the vehicle to accept the packets. The chances of this happening are near-zero, and the difficulty can be increased by increasing the number of valid packets required before they are considered trustworthy.

Tracking.

The primary objective of this solution is to prevent tracking based on sensor ID's by changing the ID with every transmission. An eavesdropper without the secret key cannot guess the next ID, so they must resort to other methods to utilize these packets. It should be noted that it may be feasible to associate transmissions with vehicles based on the data that is usually sent without any obfuscation. Variations in sensors, tires, and vehicles could yield combinations of temperature and pressure data that are unique enough to associate the combined data with a specific vehicle. Under normal conditions, these should show little or predictable variance per tire. Given enough readings, this data of generally low value could be used to reconstruct that vehicle's driven route. To reduce this attack surface without resorting to computationally expensive encryption, the data could be obfuscated based on the secret key or sequence number, such that an eavesdropper could not easily determine the data values to associate with an identity. This was discussed in greater detail in Section 3.9.

Replay attack.

Static TPMS sensors are vulnerable to a simple replay attack, where a single packet could be recorded and rebroadcasted as often as an attacker wishes. Note that the sophistication of a replay attack is not needed in Static TPMS implementations because the attacker can craft any packet desired and the receiver will accept it, provided the correct, and trivially acquired, ID is used. The solution proposed here thwarts single-packet replay attacks by requiring a correct sequence of hopping ID's before the packets are trusted, therefore a single rebroadcast would not be trusted. The core proposal would still be vulnerable to a replay attack of a lengthy sequence, as it does not provide a way of knowing where in the hopping sequence the transmitter is, only if the future packets will be valid. This would be most dangerous when used to hide a real problem, where a tire is put in a dangerous state but the replayed packets are from a time period of normal operation.

This attack on TORI already requires much more difficulty than the existing system, but can be alleviated in many ways. Including a Message Authentication Code (MAC) that incorporates the secret key would prevent the data from being modified before transmission, lowering the impact of the attack at the cost of hardware complexity and power usage. Sequence numbers can be used to cause the receiver to reject the replayed packets. Consider a mode on the sensor where the low frequency activation signal triggers the sensor to send a special packet containing an obfuscated sequence number, rather than sensor data. This would provide time-sensitive shared secret to allow the receiver to distinguish between the true sender and a replayed sequence, dramatically decreasing the potential for an attack to work. Such a modification leverages the current low frequency signal as a side channel for extra security.

4.4 Summary

The performance of the Associator and Route Reconstructor were tested under nine experimental conditions and several seeds to examine the feasibility of tracking via TPMS and how consistent the results are. The Associator performs well in most cases, with the typical case correctly associating three tires with each other under conservative conditions, and having a small variability. The Route Reconstructor was evaluated based on how many errors were made in the graph reconstruction of vehicles. In the Low Traffic Density experiments it performed well, with many cases requiring minimal edits, but as traffic density increases the performance drops, and it demonstrated a trend where increased detector density made performance worse.

The proposal for TORI was evaluated in theoretical terms to see how it is affected by the attacks to which the present-day Static TPMS devieces are vulnerable, and contemplate potential new, albeit more difficult attacks. The security measures added defeat most attacks, and the remaining attack such as an extended replay have lower impact and higher cost than against the previous generation of TPMS.

V. Conclusion

5.1 Introduction

The absence of security features in TPMS presents an attack surface that could be exploited by a moderately equipped attacker. This thesis showed the feasibility of one such attack, a potential solution, and the need for further research and manufacturer intervention before safety-critical flaws are discovered. This chapter summarizes the goals of this research in Section 5.2, draws conclusions about the research in Section 5.3, summarizes the contributions to the area of research in Section 5.4, presents avenues for future work in Section 5.5, and concludes with some final remarks in Section 5.6.

5.2 Motivation and Research Goals

TPMS adoption is already widespread, but could benefit from a more secure implementation. Given that these devices are mandated on most consumer vehicles, and are therefore found in large numbers on our roadways (Critical Infrastructure), they deserve a close examination. The goal of this research was to examine what effects the lack of security in TPMS can have, and show the feasibility of one type of attack in a large-scale environment. The results of this exploration could be used to deploy a sensor network to leverage the exposed data, but could also be used to secure the next generation of TPMS.

5.3 Conclusions

Chapter III presented the methodology for simulating realistic traffic patterns and generating TPMS packets from that data. This utilized features built into Simulator for Urban MObility with additional post-processing to generate TPMS packets. The resulting simulation provided observations at a variable number of intersections in a map of downtown Dayton, OH. An Association algorithm was used to create virtual identities from these observations by correlating tire IDs observed in close proximity. A basic Route Reconstruction algorithm was presented to turn the potentially sparse observations into complete paths using the knowledge of the area. Finally, this chapter proposed TORI, a solution to many of the vulnerabilities present in TPMS and operating within the design constraints of the current system.

Chapter IV presented the results of testing on the Associator and Route Reconstructor. The experiment was conducted under 9 different conditions, varying Traffic Density and Detector Density, with numerous seeds to examine variation. The Associator was shown to group tire IDs with fairly high accuracy, even in sparse detection environments. These observations could then be sent to a Route Reconstruction algorithm to turn sparse paths into complete paths, with an accuracy relative to traffic density. Finally, TORI was examined to describe how it combats the current attacks and what new attacks may be possible, albeit with a significantly higher cost to the attacker.

5.4 Contributions

This research presented new work using simulation to demonstrate vulnerabilities that had only been briefly explored before. Previous work had been primarily signals oriented and was demonstrated in a specific set of real-world experiments [7]. The construction of a simulation, Associator, and basic Route Reconstruction showed that the methods used by those researchers to harvest packets could be deployed in a wide area for a new effect. This research also proposed a new iteration of TPMS protocol that differs from other solutions by operating within the current design constraints and even leveraging them in new ways. Beyond the simulation infrastructure that was built and could be used for other experimentation, the results of these experiment demonstrated two primary conclusions. One is that tracking is feasible with relatively little infrastructure or complex technology, being able to associate tires with vehicles with high accuracy which can be leveraged for a variety of attacks. The other is that a rolling ID scheme can significantly raise the level of effort required to exploit such a system, in these cases reducing the effectiveness by over 90% and even reducing the ability to track the unsecured devices by 30%.

5.5 Future Work

This thesis focused on how the absence of security in TPMS can lead to one type of privacy exploitation. This required adapting and constructing new programs and limited the scope of this project. After a small foray into the signals involved in TPMS, research quickly moved into simulation to allow for further proof-of-concept research. There are many pieces that should be explored in greater detail, given more time or a different research angle. These include:

- More Proof-of-Concept attacks
- Increasing the size of the simulation and including more diverse traffic conditions
- Implementing intelligent selection of intersections for detectors, such as in traffic chokepoints, or at all n-way intersections requiring a minimum threshold for n
- Test new algorithms or heuristics
- Apply machine learning to the general association problem, or routing on a specific map
- Apply the workflow and algorithms for association and route reconstruction to a different identification technology

- Executing a small-scale real-world test
- Build hardware intermediaries to test the physical requirements and implications of TORI

5.6 Final Remarks

Tire Pressure Monitoring Systems are of great benefit to drivers, and have surely prevented accidents since their inception. Despite their seemingly insignificant value or impact from a cyber-attack perspective, the safety-critical nature and number of devices present demands that they be given a fair investigation. This thesis examined some of the issues that the current generation of TPMS is already vulnerable to, and demonstrated that an attacker with enough determination could use these devices to establish pattern-of-life or actual travel route data. This could be used for benevolent or nefarious means, but is not an intended outcome of the technology and should therefore be a concern for the consumer. The current state of TPMS security is not necessarily the result of neglect; rather, it shows that even the smallest detail could lead to exploitation, and the next generation of devices should take measures to combat the threats that are currently known, and proactive security measures to block new attacks.

Bibliography

1. DHS, "NIPP 2013: Partnering for Critical Infrastructure Security and Resilience," *Homeland Security*, p. 57, 2013.

2. N. H. T. S. Administration, "Evaluation of the Effectiveness Of TPMS in Proper Tire Pressure Maintenance," DOT HS 811 681, 2012.

3. ——, "Tire-Related Factors in the Pre-Crash Phase," DOT HS 811 617, 2012.

4. ——, "Federal Motor Vehicle Safety Standards; Tire Pressure Monitoring Systems; Controls and Displays; Final Rule," Federal Register Vol 70 Number 67, pp. 18 135–18 191, 2005.

5. E. Commission, "Top News from the European Commission 23 November to 20 December 2009," 2009, accessed 26 August 2018. [Online]. Available: http://europa.eu/rapid/press-release_AGENDA-09-40_en.htm

6. S. Velupillai and L. Guvenc, "Tire Pressure Monitoring [Applications of Control]," *Control Systems, IEEE,2007*, no. December, pp. 22–25, 2007.

7. I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Grutese, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study." *Proceedings of the USENIX Security Symposium*, vol. 39, no. 4, pp. 11–13, 2010.

8. C. Valasek and C. Miller, "Remote Exploitation of an Unaltered Passenger Vehicle," *Technical White Paper*, vol. 2015, pp. 1–91, 2015. [Online]. Available: http://illmatics.com/RemoteCarHacking.pdf

9. L. Pan, X. Zheng, H. X. Chen, T. Luan, H. Bootwala, and L. Batten, "Cyber security attacks to modern vehicular systems," *Journal of Information Security and Applications*, vol. 36, pp. 90–100, 2017. [Online]. Available: https://doi.org/10.1016/j.jisa.2017.08.005

10. S. Checkoway, D. Mccoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," *System*, p. 6–6, 2011. [Online]. Available: http://www.usenix.org/events/security/tech/full_papers/Checkoway.pdf

11. R. L. Hansen, J. A. Love, D. K. Melgaard, D. B. Karelitz, T. A. Pitts, J. D. Zollweg, D. Z. Anderson, P. Nandy, G. L. Whitlow, D. A. Bender, and R. H. Byrne, "Large Scale Tracking Algorithms," no. January, 2015.

12. T. Singh and Y. Cheng, "Efficient particle filtering for road-constrained target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1454–1469, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4441751

13. E. Barker and J. Kelsey, "Recommendation for random number generation using deterministic random bit generators (revised)," *NIST Special publication*, vol. 800, no. March, p. 90, 2012.

14. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 1997.

15. R. Beaulieu, D. Shors, J. Smith, and S. Treatman-clark, "The simon and speck families of lightweight block ciphers," *Cryptology ePrint Archive*, no. National Security Agency. USA, pp. 1–42, 2013. [Online]. Available: http://eprint.iacr.org

16. T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE : A Lightweight Block Cipher," *Lecture Notes in Computer Science*, vol. 7707, pp. 339–354, 2013. [Online]. Available: http://jpn-nec-com-org.onenec.net/rd/crl/code/research/image/twine_SAC_full_v5.pdf

17. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An Ultra-Lightweigtht Block Cipher," *Cryptographic Hardware and Embedded Systems - CHES 2007*, vol. 33, pp. 462–478, 2007. [Online]. Available: http://link.springer.com/10.1007/978-3-540-74735-2_31

18. K. Shibutani, T. Isobe, H. Hiwatari, and A. Mitsuda, "Piccolo : An Ultra-Lightweight Blockcipher," pp. 1–16.

19. C. De Canniere, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers," *Cryptographic Hardware and Embedded Systems-CHES 2009*, pp. 272–288, 2009.

20. Y. W. Law, J. Doumen, and P. Hartel, "Survey and benchmark of block ciphers for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 65–93, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1138127.1138130

21. K. Emura, T. Hayashi, and S. Moriai, "Toward Securing Tire Pressure Monitoring Systems : A Case of PRESENT-based Implementation," no. C, pp. 403–407, 2016.

22. M. Xu, W. Xu, J. Walker, and B. Moore, "Lightweight secure communication protocols for in-vehicle sensor networks," *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles - CyCAR '13*, pp. 19–30, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2517968.2517973

23. D. K. Kilcoyne, S. Bendelac, J. M. Ernst, and A. J. Michaels, "Tire Pressure Monitoring System encryption to improve vehicular security," *Proceedings - IEEE Military Communications Conference MILCOM*, pp. 1219–1224, 2016.

24. C. Solomon and B. Groza, "LiMon - Lightweight authentication for tire pressure monitoring sensors," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9588, pp. 95–111, 2016.

25. D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO – Simulation of Urban MObility," vol. 5, no. 3, pp. 128–138, 2012.

26. D. Krajzewicz, "Networks/Import/OpenStreetMaple," 2019. [Online]. Available: https://sumo.dlr.de/wiki/Networks/Import/OpenStreetMap

27. A. Sanfeliu, "Relational Graphs for Pattern Recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, pp. 353–362, 1983.

28. Y. J. Lin and S. J. Hwang, “Temperature prediction of rolling tires by computer simulation,” *Mathematics and Computers in Simulation*, vol. 67, no. 3, pp. 235–249, 2004.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 03–21–2019	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sept 2017 — Mar 2019

4. TITLE AND SUBTITLE Preserving Privacy in Automotive Tire Pressure Monitoring Systems	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Hacker, Kenneth L., 2d Lt, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering an Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GE/ENG/19M
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The automotive industry is moving towards a more connected ecosystem, with connectivity achieved through multiple wireless systems. However, in the pursuit of these technological advances and to quickly satisfy requirements imposed on manufacturers, the security of these systems is often an afterthought. It has been shown that systems in a standard new automobile that one would not expect to be vulnerable can be exploited for a variety of harmful effects. This thesis considers a seemingly benign, but government mandated, safety feature of modern vehicles; the Tire Pressure Monitoring System (TPMS). Typical implementations have no security-oriented features, leaking data that can be used for reliable tracking by a determined attacker, and being completely open to spoofing attacks. This research investigates potential privacy concerns of TPMS, first by demonstrating the feasibility of both identifying vehicles and reconstructing their routes without prohibitive cost or expertise. Then, an ID obfuscating scheme is proposed, called TPMS Obfuscation through Rolling ID (TORI), to mitigate these privacy threats while remaining true to the design requirements of TPMS. Various conditions are tested using a modified traffic simulator, which validate the ability to reconstruct the identities of vehicles even from sparse detections.

15. SUBJECT TERMS

TPMS, Privacy, Automotive Security, Wireless Sensors, Safety, Tracking, SUMO

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Scott R. Graham, AFIT/ENG
U	U	U	UU	96	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, x4581; scott.graham@afit.edu