Vehicle Identification and Route Reconstruction via TPMS Data Leakage

Kenneth Hacker, Scott Graham, Stephen Dunlap

Chapter 7

VEHICLE IDENTIFICATION AND ROUTE RECONSTRUCTION VIA TPMS DATA LEAKAGE

Kenneth Hacker, Scott Graham and Stephen Dunlap

Abstract Tire pressure monitoring systems have become a mandatory feature of modern automobiles, but their presence opens a new attack vector for a potential adversary. These systems have minimal security features, allowing for eavesdropping and data injection with low technical and financial costs.

This chapter explores the potential for tire pressure monitoring systems to provide inputs to a remote sensing network, which leverages the data broadcast by the systems to identify vehicles and track their movements. A traffic simulation is employed to generate vehicle movements and tire pressure monitoring system packets. Experiments demonstrate that the tire pressure monitoring system data can help identify vehicles and reconstruct vehicle routes. They show that a determined adversary could deploy sensors to detect tire pressure monitoring systems and learn about the movements of individual vehicles without any insider information. Potential solutions to this privacy problem are discussed, focusing on low cost changes with the greatest consumer security benefits.

Keywords: TPMS data leakage, vehicle identification, route reconstruction

1. Introduction

The adoption of a new technology is exciting, with vendors and government regulators eager to lead the way, but often without considering the security risks. Even seemingly insignificant systems can provide avenues for an adversary to gain information or influence. This is the case with vehicular technologies, where manufacturers must balance consumer desires, company goals and safety obligations. Tire pressure monitoring systems (TPMSs), which employ wireless communications to provide tire status information on drivers'

dashboards, are a mandatory safety feature in all new vehicles. However, the wireless signals are neither protected from eavesdroppers nor are they authenticated, enabling a malicious actor to gain access to sensitive data, or worse, manipulate the system.

This chapter discusses potential privacy threats that result from TPMSs being installed in the majority of vehicles on roadways. Experiments were conducted using traffic simulations with realistic TPMS data that an adversary could collect and analyze. The experiments demonstrate that a determined adversary could deploy sensors to identify vehicles using TPMS data and learn about the movements of individual vehicles without any insider information. Potential solutions to this privacy problem are discussed, focusing on low cost changes with the greatest consumer security benefits.

2. Tire Pressure Monitoring Systems

This section provides an overview of TPMSs, including TPMS legislation, implementation, attacks and security.

2.1 Legislation

In the United States, steps toward mandating TPMSs in new vehicles initiated after a number of traffic fatalities related to defective tires. The Transportation Recall Enhancement, Accountability and Documentation (TREAD) Act of 2000, which was rapidly passed by the U.S. Congress [9], called for a mandatory system that would warn drivers when one or more vehicle tires were significantly underinflated.

The National Highway Traffic Safety Administration [5] drafted more detailed compliance requirements for all new vehicles starting from September 1, 2007. These requirements included reporting to the driver if one or more tires were 25% below minimum pressure within 20 minutes of the pressure dropping. The European Commission [3] mandated TPMSs in all new vehicles after 2012 as part of a major safety and emission-reduction program. As a result, millions of TPMS-equipped vehicles are on the roadways and their percentage is growing as older vehicles are removed from service.

2.2 Implementation

A TPMS unit embedded in the tire of a vehicle periodically reads its pressure and temperature sensors, constructs a network packet, encodes it (e.g., with Manchester encoding) and transmits it using amplitude shift or frequency shift keying to a vehicle TPMS receiver, which forwards it to a central control module for analysis. Tire pressure alerts are sent by the control module directly or indirectly to the dashboard where they are displayed.

The wireless data packets transmitted by most TPMS units are fairly simple. A typical packet includes 32 bits for an ID, eight bits of pressure data, eight

bits of temperature data, four bits of status flags and twelve bits for a cyclic redundancy check (CRC).

Unfortunately, the tire pressure and temperature data in the network packets are neither encrypted nor significantly obfuscated, allowing anyone in wireless proximity who captures the packets to read the data. Notably, information is only transmitted in one direction, from the tire device to the vehicle TPMS receiver, in order to conserve battery power for the sensors, which are in the sleep mode for the vast majority of the time.

The main reason for the lack of security in TPMSs is the increased power cost required for encryption and two-way communications. The lifespans of the batteries are five to ten years; size and weight requirements preclude the use of larger batteries [10]. The tire pressure unit, which includes a battery, is usually set in epoxy inside the tire; the individual components are, therefore, not replaceable. As a result, the entire tire pressure unit has to be replaced when the battery is depleted.

2.3 Attacks

In 2010, Rouf et al. [6] published an evaluation of TPMS attack scenarios as part of a case study of in-car wireless networks. They demonstrated that the lack of authentication and integrity checks made spoofing trivial, leading to malicious effects such as displaying false information and warning lights, and disabling the TPMS control unit.

In the same article, Rouf and colleagues [6] discussed the feasibility of tracking vehicles based on their TPMS sensor broadcasts. Given the static IDs of the four tires associated with a vehicle, it is simple to associate them with the identity of a vehicle. In fact, given the data from all four tires of a vehicle – and without considering any other data such as geographic locations – there would have to be more than one billion vehicles on the road to even approach a 1% chance of misidentifying the vehicle.

Creating an eavesdropping infrastructure can be challenging, especially for passive data collection. The low power transmissions from TPMS units require receivers to be positioned close to vehicles, so large numbers of receivers would have to be placed along roadways to ensure that the infrequently-transmitted packets are captured.

A more effective solution may be to stimulate a TPMS transmission using a low frequency activation signal. While this process is more complex and prone to noise, it could guarantee readings at points of interest along roadways.

Rouf and colleagues [6] also compared TPMS-based tracking of vehicles against the other alternative for tracking vehicles – automatic number plate reading. According to their study, tracking via TPMSs would have higher read rates (99% versus 90%) and would not require line-of-sight measurements. However, TPMS-based tracking by law enforcement would require changes to existing laws and regulations.

2.4 Security

Researchers have proposed approaches for rendering TPMS-based tracking and spoofing of vehicles more difficult by obfuscating the packet IDs. Xu et al. [11] have proposed a system incorporating pseudo-IDs, sequence numbers, message authentication codes and session keys, which addresses many of the privacy and integrity problems. However, their system, which requires a three-way handshake to establish keys, does not work with current TPMS sensors because they are not equipped to receive data.

Emura et al. [2] have examined sensor costs and have demonstrated a protocol that could be used under current TPMS constraints. Other researchers [4, 8] have shown that rolling IDs that change between TPMS transmissions are feasible and can defeat tracking methods. The next generation of TPMSs may incorporate these and other upgrades. However, automobile manufacturers are not as yet concerned about TPMS vulnerabilities, so the security problems persist.

3. Background

This section describes the traffic simulator and the performance metrics used in this research.

3.1 Simulator for Urban Mobility

Simulator for Urban Mobility (SUMO) is an open-source traffic simulation suite that provides several tools for mapping and traffic generation, manipulation and simulation. First released in 2002, SUMO continues to be actively enhanced [1], providing a platform for testing vehicular routing protocols, executing traffic congestion models and generating realistic traffic data that can be used for further research.

SUMO is a microscopic simulator in that the level of simulation goes down to individual vehicles and lanes, with the vehicles acting on their own and responding in a realistic manner. In contrast, macroscopic simulators abstract the individual vehicles into general traffic flows in sections of a map. SUMO models maps as nodes (intersections) and edges (roads) on a Cartesian grid; the maps can be constructed, randomly generated or imported from sources such as OpenStreetMap. 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 cars, trucks or buses, or customized vehicles can be developed to meet the simulation needs.

Simulations are defined by map and route files. The map establishes the places where a vehicle may travel, along with the road conditions and restrictions. The route defines the points at which a vehicle enters and exits the roadway, the roads on which it travels and its behavior during the trip. The simulation can be modified in real time using the Traffic Control Interface

(TraCI) to observe how changing conditions such as traffic lights or a collision may affect the simulation.

3.2 Measurement Metrics

This chapter discusses algorithms for identifying vehicles and reconstructing their routes. In order to evaluate the effectiveness of the algorithms, metrics are needed to compare their results against truth data in the simulation. Each metric is intended to provide relative assessments of the "goodness" of various configurations.

Data from the simulation is passed to the tire ID association phase, which transforms TPMS observations into vehicle identities. The vehicle identities and associated observations are sent to the route reconstruction phase, which processes the individual observations to create complete routes.

Two metrics were selected: (i) Jaccard distance used in the tire ID association phase; and (ii) graph edit distance used in the route reconstruction phase:

- **Jaccard Distance:** During the tire ID association phase, sets are compared to determine the combinations of IDs that are commonly found together. The Jaccard distance is a set similarity metric that is commonly used for spell checking strings [12]. The Jacquard distance J of two sets A and B is given by:

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

  The Jaccard distance is used to compare a candidate set of tire IDs against tire IDs observed during a time window at a specific intersection along a route. For example, if the candidate set is {`0xa1`, `0xb2`, `0xc3`} and if the set of tire IDs observed during a five-second window at an intersection is {`0xcc`, `0xdd`, `0xff`, `0xa1`, `0xb2`}, then the Jaccard distance is computed as:

$$\frac{|\{\texttt{0xa1}, \texttt{0xb2}\}|}{|\{\texttt{0xa1}, \texttt{0xb2}, \texttt{0xc3}, \texttt{0xcc}, \texttt{0xdd}, \texttt{0xff}\}|} = \frac{2}{6} = 0.33$$

  This score helps determine the best association of tire IDs. Additionally, it is used to compare candidate associations against true sets belonging to vehicles in order to judge their goodness.

- **Graph Edit Distance:** A target route on a road map and a candidate route are modeled as directed graphs where the nodes are intersections and the edges are roads. The graph edit distance, which compares the similarity between two graphs, is widely used in pattern matching [7]. It is employed in the route reconstruction phase to score candidate routes.

The graph edit distance is defined as the minimum number of modifications required to transform the graph corresponding to a candidate route to the target graph. In this work, the modification operations correspond to insertions, deletions or substitutions of nodes or edges. Each modification operation can be weighted differently to reflect the impact of the operation. Specifically, insertions and deletions have weights of one. A substitution has a weight of two because it corresponds to a deletion followed by an insertion.

4. Simulation Methodology

This section describes the simulation methodology.

4.1 Simulation Setup

The main steps in the simulation setup phase are: (i) geographical map generation; and (ii) traffic generation.

- **Geographical Map Generation:** The map employed in the simulation covered a section of downtown Dayton, Ohio. SUMO provides a tool that uses OpenStreetMap to download real data for an area, which accurately represents traffic lights, speed limits, one-way streets and other elements of traffic flow.

  The map size is a simulation parameter that may be adjusted to serve various purposes. The map size chosen for the simulation was approximately 600 nodes and 1,200 edges. This map was selected for reasons of familiarity and to represent sufficiently diverse traffic conditions that could demonstrate the feasibility of the approach. An urban deployment with a relatively high density of intersections was of particular interest in this study.

- **Traffic Generation:** The SUMO Python script *randomTrips* employed the network description, simulation time, optional seed and traffic density to generate an XML trip file that described every vehicle created along with its source node and destination. The SUMO DUAROUTER tool converted the source/destination pairs to actual routes that described the roads that each vehicle could take during the simulation.

- **Wireless Communications Detection:** SUMO includes a package for wireless communications that can model technologies such as Bluetooth and vehicular ad-hoc networks (VANETs) [1]. Vehicles may be given receivers and transmitters independently, and the assignments can be made explicitly or randomly using a user-specified percentage.

  All the vehicles in the simulation were assumed to have transmitters (i.e., they were equipped with tire pressure sensors). The TPMS detectors were modeled as vehicles equipped with receivers that were parked at intersections. Edge cases, corresponding to situations where multiple intersec-

tions were very close to each other, were handled by manually removing overlapping detectors.

When a vehicle enters detector range in the simulation, data is recorded in an XML file associated with the vehicle. At the end of the simulation, this XML file contains considerable details about the vehicle route and travel conditions. Packets may be optionally dropped by eliminating a percentage of detectors, simulating heavy versus sparse detector deployments.

This method of modeling wireless detectors differs from real-world deployments, but it has enough fidelity to achieve the research goals. A real detector would likely be a directional antenna that could only receive data from a few lanes, possibly requiring $2n$ detectors for an n-way intersection. This could actually improve a tracking algorithm by providing travel directions. However, the simulation conducted only considered binary detections at intersections – was a vehicle present at the intersection and at what time? Based on previous research and working within the time constraints, these inputs were deemed adequate for purposes of tire ID association and route reconstruction. The additional benefit is that this type of data could come from detectors other than TPMS sensors (which are of interest as VANET technologies enter the roadways), but the data could still be applied to existing systems such as automatic number plate readers.

4.2 TPMS Packet Generation

In the TPMS packet generation phase, wireless observation data generated by SUMO is post-processed to produce TPMS packets needed for tire ID association. In this phase, most of the data is stripped to prevent sensitive information such as actual vehicle IDs, speeds and routes from being accessed in the later phases.

The TPMS packet generation phase starts with a dictionary containing time and location data for all wireless observations; the data is indexed by the observed vehicle ID. For each unique vehicle, four random 32-bit tire IDs are generated. For each vehicle observation, the simulation must decide which tires have been observed.

A simple probabilistic model was developed based on previous experiments that used a directional antenna to measure the attenuation due to vehicles. The model assumed that transmissions from the two tires closest to a roadside detector (i.e., right-side tire transmissions) would always be detected, and the left-front and left-rear tire transmissions would be detected with probabilities of 50% and 10%, respectively. If a tire is deemed to be detected at an observation point, then the location and timestamp are placed in a new dictionary indexed by the tire ID. This ensures that the resulting data structure does not contain the true vehicle ID, and is at most four times larger than the original data structure.

4.3 Tire ID Association

The main steps in the tire ID association phase are: (i) candidate association creation; and (ii) candidate association scoring:

- **Candidate Association Creation:** This step attempts to associate the observed tire IDs with one another to create a tuple called a candidate (tire ID) association, which ideally belongs to one vehicle. Each tire ID has an associated list of observations that form a route. Because it is unlikely to obtain data about all four tires of a vehicle at every intersection, tire IDs that belong with each other (i.e., from the same car) would have similar, but not necessarily identical, lists of observations.

  All the tire IDs observed at a given location during a certain time window (chosen as one second in the experiments) are examined. For every tire ID, the frequency with which every other tire ID is observed near the selected tire ID is tallied across the entire observed route. In a high-density traffic environment, two vehicles could be close enough to yield overlapping tire IDs. Thus, tire IDs from nearby vehicles have to be filtered.

  This is accomplished by creating sets of the four most frequently observed tire IDs with respect to the tire ID being evaluated. The Jaccard distance metric is used to compare these sets against the sets observed at each location. The set with the highest average Jaccard distance across the route is considered to be an identity and is saved in a scoring matrix. This enables a tunable metric to be used to manipulate the risk/reward of associating more tire IDs. Additionally, because the route for a set of tire IDs may appear to be different even if they belong to the same vehicle, it is possible for an infrequently observed tire ID to appear to be associated with a different set of tire IDs.

  A scoring matrix adds a second layer of filtering to reduce this error. After all the tire IDs are considered independently, the scoring matrix is evaluated to create the final virtual vehicle identities, which are the sets of tire IDs belonging a unique vehicle. If a set of four or fewer tire IDs are consistently grouped with each other, the tire IDs in the set are assumed to belong to a specific vehicle, the set is designated as a candidate association and the tire IDs are removed from further consideration. Otherwise, if a set of more than four tire IDs appear to be related, then four tire IDs that are most frequently associated with each other are assumed to belong to a specific vehicle; this set is also designated as a candidate association.

- **Candidate Association Scoring:** When scoring a candidate association, the tire IDs grouped as corresponding to a vehicle identity should be evaluated with respect to each other instead of attempting to match them against a true vehicle. Additionally, the risk/reward of attempting to add a third or fourth tire ID to the group should increase.

Table 1. Candidate association scores.

Matches	Set Size	Comments	Score
0	4	Four incorrectly associated tires (worst case)	0.14
0	3	Three incorrectly associated tires	0.17
0	2	Two incorrectly associated tires	0.2
0	1	One associated tire is effectively not an association	0.25
1	4	Two correctly associated tires and two unrelated tires	0.33
1	3	Two correctly associated tires and one unrelated tire	0.4
1	2	Two correctly associated tires	0.5
2	4	Three correctly associated tires and one unrelated tire	0.6
2	3	Three correctly associated tires	0.75
3	4	Four correctly associated tires (ideal case)	1.0

The method used to score a candidate association performs a reverse lookup in the truth data to find the true vehicle ID that is associated with each tire ID in the candidate association. This creates a tuple of one to four tire IDs, each of which may correspond to the same vehicle or, in adverse cases, multiple vehicles.

Jaccard similarity is used to compare a candidate association against the most likely true vehicle. As shown in Table 1, in the case of a four-wheeled vehicle, the Jaccard similarity score ranges from 0.14 to 1.0. Note that the first column corresponds to the number of tire IDs in a candidate association that belong to the same vehicle. Thus, the values range from zero (all the tires belong to different vehicles) to three (all the tires belong to the same vehicle).

4.4 Route Reconstruction

The main steps in route reconstruction are: (i) candidate route creation; and (ii) candidate route scoring:

- **Candidate Route Creation:** The output from the tire ID association phase, which is input to the route reconstruction phase, comprises a list of candidate vehicles and their associated observations (location-

timestamp pairs). The algorithm used in route reconstruction is assumed to have complete knowledge of the roads in the geographic area where the detectors are placed. This knowledge is encoded in a graph where the intersections are nodes and the roads are edges. The edges are weighted based on the estimated travel times to traverse the edges.

The algorithm examines the observations for a given vehicle and attempts to predict the most likely route corresponding to the observations. This is accomplished by taking two consecutive observations and finding a simple path (without loops) whose estimated travel time is the closest to the difference between the observed times. The complete vehicle route is created by repeating this step for all the observations corresponding to the vehicle. This entire process is repeated until candidate routes are generated for all the vehicles.

- **Candidate Route Scoring:** Because the road network is modeled as a graph, the graph edit distance is the natural choice for quantifying the correctness of candidate routes. The truth data is used to build a directed graph containing only the nodes and edges that are actually traversed by a vehicle. The graph corresponding to the candidate route is compared against the truth graph.

  The candidate route score is computed by tallying the weights corresponding to the minimum number of insertions, deletions or substitutions required to convert one graph to the other. Note that an insertion implies that a node or edge is in the candidate route whereas a deletion implies that an extraneous (incorrect) node or edge exists in the candidate route.

4.5 Simulation Variables

Two variables, detector density and vehicle density, were employed in the simulation experiments. The values of these variables were varied in the simulation runs.

Three detector density values were employed, low, medium and high, corresponding to detectors placed at 10%, 50% and 100% of intersections, respectively. The detector densities were selected to provide insights into the optimal number of detectors that should be used when cost and infrastructure size are considerations.

Three vehicle densities were employed, low, medium and high, corresponding to 200, 500 and 2,000 vehicles, respectively. These densities were manually determined based on how much traffic could be handled without becoming overwhelmingly gridlocked.

5. Simulation Results

This section presents the tire ID association and route reconstruction results.

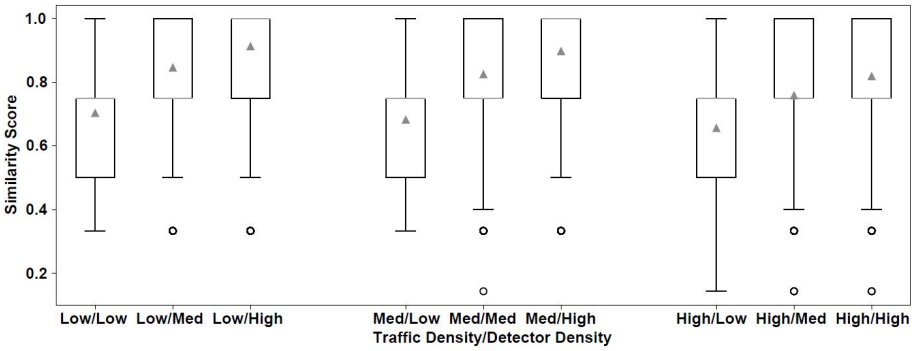

Figure 1. Tire ID association results.

5.1 Tire ID Association Results

A candidate tire ID association was scored based on pairwise matches in a reverse lookup of tire IDs that were believed to be associated with each other. This metric provided a relative measure of goodness as the experimental conditions changed.

Each experimental configuration was run over 150 seeds, which varied the routes and detector placements while keeping the detector and traffic densities constant. The resulting scores are shown in the boxplots of Figure 1. Note that the triangles denote the means of the experiments whereas the circles denote outliers. All the experiments with medium or high detector coverage achieved mean scores greater than 0.75. This demonstrates that the correct 3-tuples were identified frequently and that significant mismatches rarely occurred.

Certain trends that followed expected patterns emerged from the data. The means always improved with increasing detector density because more observations provided more opportunities to discern patterns. Every experiment had at least one case where all four tire IDs were correctly associated; this is likely to occur when there are enough vehicles and long enough routes to observe all the tires. The minimum scores appeared to be affected more by traffic density than detector density, with the worst cases getting worse as the traffic density increased. This is likely the result of traffic congestion, which causes vehicles to gather at intersections, effectively forming caravans. Multiple vehicles passing by a detector in a short window increased the likelihood of errors in tire ID associations. This issue may be alleviated by adjusting the locations of detectors so that important intersections are adequately covered.

5.2 Route Reconstruction Results

A candidate route was scored based on the graph edit distance between the candidate route and the true route travelled by the vehicle. Since every node or edge inserted/deleted incurred a cost of one and every modified node or edge incurred a cost of two, a score of zero corresponded to a perfect route

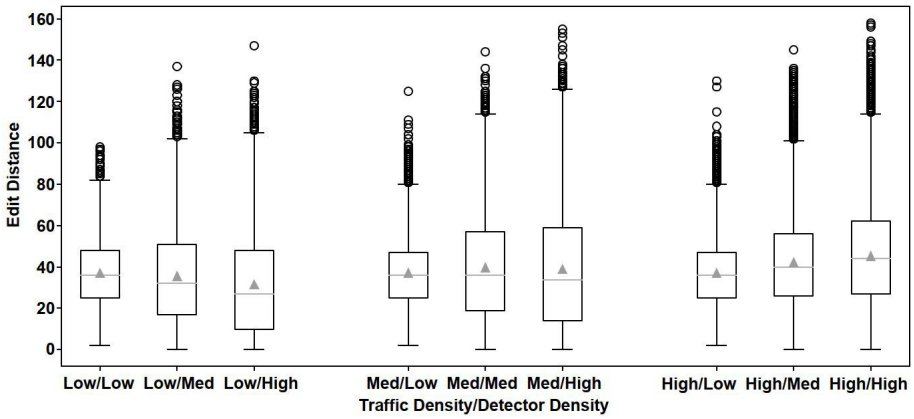

Figure 2. Route reconstruction results.

reconstruction. Note that, because the routes were random, the length of each route varied and the numbers of nodes and edges travelled did not have constant relations to the physical distance travelled. As such, the graph edit distance was used as a natural metric to observe trends between experiments when maps were modeled as directed graphs. Figure 2 shows the distribution of scores for the route reconstruction experiments.

Expected trends in route reconstruction emerged primarily as a function of detector density. When the detector density was high, some vehicle routes were reconstructed perfectly. In the case of low vehicle density and a high number of detectors, the average score was the best in every experiment. However, when the vehicle density was high, adding more detectors did not improve the average score. This occurred because the heuristic that was used guessed the route between observations based on travel time. As the difference between actual and expected travel times on a given roadway increased, the route reconstruction accuracy decreased. In the high traffic density experiments, traffic congestion greatly increased the travel time over the average, leading the algorithm to assume that vehicles took longer routes. Improved graph tracking heuristics or engaging traffic congestion data could alleviate this issue when unusual road conditions are encountered.

6. Conclusions

This research has examined the security consequences of TPMSs, highlighting some vulnerabilities and demonstrating their potential negative effects. It extends the seminal work of Rouf et al. [6] by exploring TPMS security concerns in a large simulated environment. The open-source SUMO tool was leveraged to rapidly generate realistic data and conduct extensive simulations to evaluate the feasibility of associating tire pressure ID packets with vehicle identities, and subsequently track vehicles of interest.

The use of intersection-based wireless observations and realistic TPMS detection parameters resulted in high tire ID association rates despite employing a fairly simple algorithm. With knowledge of vehicle identities and timestamped locations, sparse observations could be processed to reconstruct vehicle routes with reasonable, albeit varying, accuracy. The simulation experiments demonstrate that an adversary could deploy current roadside sensors to glean pattern-of-life data for large numbers of vehicles. The low level of effort required to breach privacy should motivate further research into the proper use of the technology and push manufacturers to implement advanced security features.

Future research would be facilitated by creating a SUMO plug-in that would handle TPMSs in a simple and consistent manner. Another avenue is to develop algorithms with new heuristics that would provide improved accuracy and speed. Another potential improvement is the application of machine learning techniques, which appear viable due to the abstract association tasks and the availability of scoring metrics. Finally, the concepts and techniques developed in this research could be applied to other wireless vehicular technologies such as Bluetooth and vehicular ad-hoc networks.

The views expressed in this chapter are those of the authors, and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense or U.S. Government. This document has been approved for public release, distribution unlimited (Case #88ABW-2018-6333).

References

[1] DLR – Institute of Transportation Systems, Eclipse SUMO – Simulation of Urban Mobility, Berlin, Germany (`dlr.de/ts/sumo`), 2019.

[2] K. Emura, T. Hayashi and S. Moriai, Toward securing tire pressure monitoring systems: A case of PRESENT-based implementation, *Proceedings of the International Symposium on Information Theory and its Applications*, pp. 403–407, 2016.

[3] European Commission, Top News from the European Commission, 23 November to 20 December 2009, AGENDA/09/40, Press Release, Brussels, Belgium, November 20, 2009.

[4] D. Kilcoyne, S. Bendelac, J. Ernst and A. Michaels, Tire pressure monitoring system encryption to improve vehicular security, *Proceedings of the IEEE Military Communications Conference*, pp. 1219–1224, 2016.

[5] National Highway Traffic Safety Administration, Federal Motor Vehicle Safety Standards; Tire Pressure Monitoring Systems; Controls and Displays; Final Rule, 49 CFR Part 571, Docket No. NHTSA 2000-8572, RIN 2127-AI33, Washington, DC, 2003.

[6] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe and I. Seskar, Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study, *Proceedings of the Nineteenth USENIX Conference on Security*, article no. 21, 2010.

[7] A. Sanfeliu and K. Fu, A distance measure between attributed relational graphs for pattern recognition, *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-13(3), pp. 353–362, 1983.

[8] C. Solomon and B. Groza, LiMon – Lightweight authentication for tire pressure monitoring sensors, in *Security of Industrial Control Systems and Cyber Physical Systems*, A. Becue, N. Cuppens-Boulahia, F. Cuppens, S. Katsikas and C. Lambrinoudakis (Eds.), Springer, Cham, Switzerland, pp. 95–111, 2016.

[9] U.S. Congress, Transportation Recall Enhancement, Accountability and Documentation (TREAD) Act, Public Law 106-414, 106th Congress, Washington, DC, 2000.

[10] S. Velupillai and L. Guvenc, Tire pressure monitoring [Applications of Control], *IEEE Control Systems*, vol. 27(6), pp. 22–25, 2007.

[11] M. Xu, W. Xu, J. Walker and B. Moore, Lightweight secure communications protocols for in-vehicle sensor networks, *Proceedings of the ACM Workshop on Security, Privacy and Dependability for Cyber Vehicles*, pp. 19–30, 2013.

[12] S. Yadav, A. Reddy, A. Reddy and S. Ranjan, Detecting algorithmically-generated domain-flux attacks with DNS traffic analysis, *IEEE/ACM Transactions on Networking*, vol. 20(5), pp. 1663-1677, 2012.