Alternative Route-Based Attacks in Metropolitan Traffic Systems

Sidney La Fontaine*, Naveen Muralidhar*, Michael Clifford†, Tina Eliassi-Rad*, Cristina Nita-Rotaru*
*Northeastern University, Boston, MA, USA, lafontaine.s@northeastern.edu, {eliassi,crisn}@ccs.neu.edu
†Toyota InfoTech Labs, Mountain View, CA, USA, michael.clifford@toyota.com

***Abstract*—With the growing reliance on driving direction applications that dynamically account for live traffic updates, drivers are much more likely to act optimally, by taking the shortest path to their destination, and therefore more predictably. As city networks transition into being made up of connected and autonomous vehicles, autonomous driving pilots are even more likely to act optimally and predictably. The predictability that comes from acting optimally allows motivated attackers to manipulate driver(s) to travel chosen slower alternative routes by causing disruptions on road segments that are part of faster routes. A motivated attacker could use this method to cause a number of different harms such as forcing specific vehicles to take unnecessarily long routes, forcing all vehicles traveling between popular locations to follow a chosen route, or making vehicles travel specific road segments that the attacker chose, such as toll roads. In this work, we show the feasibility and practicality of conducting such attacks on several real traffic networks of major North American cities. We analyze several attack objectives under different attacker constraints and we demonstrated that an attacker could find an attack strategy in a matter of seconds.**

***Index Terms*—Security, Connected and Autonomous Vehicles, Alternative route-based attacks**

I. Introduction

Driving direction applications that dynamically account for live traffic updates are widely used, especially in cities because interruptions in traffic are much more likely to drastically increase travel time. This increase can be avoided by using such driving applications which dynamically re-route drivers to avoid interruptions. As connected and autonomous vehicles take over the streets of cities, it will lead to an increased, and potentially universal, reliance on applications and algorithms that find optimal routes and dynamically re-route drivers to avoid interruptions. Some examples of connected and autonomous vehicles include [2]–[4], [7], [12], [17]–[19].

Although connected and autonomous vehicles are designed with capabilities and features that have the potential to provide increased safety, satisfaction, comfort, and convenience, they also bring emerging challenges to security and privacy [15]. Previous work has shown how one compromised car, with different levels of adversarial control, can be used to target a victim to make it crash into another vehicle by simply disseminating incorrect information [1], [5], [6], [9], [10], [20]. Such attacks can be extended to have a higher impact on auto transportation in an area, preventing goods transportation, access to resources, access to services in target areas, and cascading effects on supply chains for specific industries. For example, the work on threat detection in collaborative adaptive cruise control by Jagielski et al. conducted a detailed analysis of various attacks that a motivated attacker can launch on a vehicle's adaptive cruise control by influencing acceleration reported by another car's LIDAR or RADAR sensors [13]. The study on vehicle platoon misbehavior done by DeBruhl et al. considers the design of a set of insider attacks and abnormal behavior attacks that can occur in a cooperative adaptive cruise control (CACC) platoon of vehicles [6]. Other works investigate the impact street networks can have on the overall resilience to non-adversarial scenarios. Specifically, the work done by Zhang et al. examines how a transportation network's topological characteristics can indicate its ability to cope with disasters [22], while the study by Feyessa et al. conducts empirical analysis on various network robustness measures to study the contribution of a node [8].

Unlike previous work that considered attacks against a single vehicle to create crashes, we consider what the larger impact of such attacks can be on a metropolitan traffic system. Specifically, we consider attackers that want to manipulate the routes connected and autonomous vehicles travel. Unfortunately, as the routes that vehicles take become more likely to be optimal, they also become more predictable for a motivated attacker to manipulate. Given such a possibility to compromise victim vehicles on roads, a motivated attacker equipped with such an ability, after careful planning and crafting a coordinated attack, could crash or halt victim vehicles at strategic points in a metropolitan area or highways to cause congestion of denial of traffic movement. We assume that the attacker coordinates a pre-planned attack and gains control over a set, S, of vehicles. For example, in the case of a cooperative adaptive cruise control platoon, each platoon could have an attacker at various parts of the metropolitan area and the attackers could coordinate crashes together as a team.

In this paper, we study the feasibility of city-wide coordinated attacks utilizing connected autonomous vehicles, by focusing on one specific attack: *alternate route-based attack*. In such an attack, the attacker knows the source and destination of the targeted victim vehicle(s) and blocks road segments preventing them from being used such that a specifically chosen sub-optimal alternative path becomes the shortest between the source and destination. While on a small scale, it may seem obvious that such attacks are feasible, it is not clear at large a scale how effective they can be and what factors influence them. We demonstrate experimentally that

such attacks are possible by using publicly available traffic maps from OpenStreetMap [16]. We model the attack as a graph problem where graph weights represent vehicle travel time and blocking a road segment means removing an edge, with edge removal costs representing the attacker's capabilities to shut down road segments. We adapt the Force Cut Problem [14] to attacks against traffic systems by modifying it to work for directed graphs, defining meaningful edge weights and edge removal costs, to model different attacks goals and capabilities. Our experimental results on the traffic systems of four major cities (Boston, San Francisco, Chicago, and Los Angeles) provide insights into the feasibility of the attack and the factors that influence it. Specifically, our contributions are:

- We evaluate four different algorithms in our experiments, one optimization-based (LP-PathCover) and three heuristic-based (GreedyEdge, GreedyEig, and GreedyPathCover). We observed that while LP-PathCover removed the lowest cost edges, it took prohibitively longer to run compared to the other algorithms. We also observed that the baseline naive algorithms (GreedyEdge and GreedyEig) were most effective for the more lattice cities, such as Chicago and San Francisco. Overall, GreedyPathCover was the most effective algorithm because it was consistently as effective as LP-PathCover, but was 5 to 10 times faster. All the heuristic-based algorithms found attack strategies in a matter of seconds.
- We compare two different attack goals: one where the attacker is targeting road segments based on the LENGTH of the road, and one where the attacker is targetting road segments based on the TIME it would take a vehicle going the speed limit to travel the length of the road. Intuitively, TIME seems more realistic given the application and we consider LENGTH a baseline given that this information is readily available from OpenStreetMap. Our results show that using both attack goals did not drastically impact the effectiveness of the attack (i.e. average number of removed edges) or the cost of the attack (i.e. average cost of removed edges).
- We compare three attack cost methods, UNIFORM, where each edge removal has the same cost of 1, LANES, where the cost of an edge removal is set to the number of lanes in the corresponding road segment, and WIDTH, where the cost of removing an edge is set to the road width divided by the average width of an American car [21]. These costs model different types of attacker capabilities that we considered. Our results show that different cost types drastically affected the average number of removed edges and the average cost of removed edges and that an attacker with the same fixed budget (i.e. the cost of removing an edge from the graph) will be less successful in the LANES or WIDTH models.
- We investigate the impact of graph topology on the effectiveness of the attack by conducting experiments on the traffic systems of four different cities; Boston, San Francisco, Chicago, and Los Angeles. We observed that for cities with traffic maps that are more lattice-like, such as Chicago, the naive algorithms were more likely to find the ideal solutions the optimization-based algorithm found, while for cities with traffic maps less lattice-like, such as Boston, the naive algorithms were much less likely to find the ideal solutions the optimization-based algorithms found. Our results show that the gap in the average cost of an attack between the baseline naive algorithms (GreedyEdge and GreedyEig) and the optimization-based algorithm (LP-PathCover) was 1.6 times bigger for Boston, a less lattice street network, than it was for Chicago, a very lattice street network. Therefore, the much faster baseline naive algorithms were much more efficient for lattice cities, making it quicker and cheaper to attack lattice cities.

The rest of the paper is organized as follows. We describe the attacks in Section II and present our results in Section III. We overview related work in Section IV and conclude in Section V.

II. Alternative Route-Based Attacks

In this section, we describe the attacks we consider in this work. We first describe our attacker model, then describe the attacks.

A. Attacker Model

In this work we look beyond the scenarios with one attacker controlling a car and its attack impact on one platoon of autonomous vehicles. Instead, we examine the scenarios with an attacker (or set of coordinated attackers) controlling several vehicles trying to create disruptions in a metropolitan traffic system. We ask the question, given a set of attacker vehicles and a street map, what kind of denial of service connectivity-related attack can the attacker create with minimal resources? This can be seen as similar to denial of service attacks in computer networks where an attacker prevents communication on particular parts of the network or targets a specific victim. In the case of traffic systems, given a set of attacker vehicles, a street map, and a service depending on the correct functioning of the street system and implemented by a set of vehicles with coordinated routes, the goal of the attacker is to disrupt critical services. Examples of such services include: emergency, police, food delivery, supply chains for specific industries, and hospitals.

Attacker knowledge. We assume the attacker knows/has access to: (1) the starting location of the targeted vehicle(s), (2) the destination of the targeted vehicle(s), (3) and the street map of the city the targeted vehicle(s) is traveling in. The attacker only has access to public-domain information of road transportation networks. Such information could include - city street networks, city maps, highway networks, location of various amenities and points of interest, traffic volume information, travel time function of roads, traffic capacity of roads, road speed limits, average incident response times, peak hour and time period specific traffic data, etc.

Attacker's objective. The main goal of the attacker is to disrupt the traffic system. For example, an attacker can try to disconnect (partition) some target area of interest in a metropolitan city. Such an impact is intended to ensure that the target area of interest is not practically reachable from any other part of the city outside of the target area. By selecting a target area containing key points of interest such as hospitals, police stations, and other critical services, an attacker could potentially severely impact the accessibility to such services.

An attacker can perform topological analysis on the road network graph representation to find critical roads, as reflected by their high (edge) *betweenness centrality* values of their corresponding edges in the graph. The (edge) *betweenness centrality* of an edge, e, is the fraction of shortest paths between all possible pairs of nodes in the graph that pass through e. Since edges with high (edge) *betweenness centrality* are indicative of their control over information passing through them by the virtue of being part of many shortest paths, the metric is assumed to generally identify highly traveled roads in a road network

Another attacker goal can be to force a single vehicle to travel a chosen sub-optimal chosen route, and this attack could be easily adapted to have much broader implications. A motivated attacker could feasibly use the techniques discussed in this paper to coerce multiple drivers to take a chosen sub-optimal alternative route, make all drivers traveling between common locations take much slower routes, force victim vehicles onto a chosen road segment, such as a toll road, or to utilize vehicles to disrupt access to a critical resource such as a hospital or factory.

Attacker constraints. The attacker is constrained by physical capabilities in terms of the number and type of vehicles and the cost associated with shutting down a road segment. For example, a truck can be used to shut down a multi-lane road, while a small vehicle can only partially shut down the road, i.e. to shut down the entire road more than one vehicle would be needed, thus the cost will be higher.

B. Alternative Route-based Attacks

We focus on one specific attack namely, alternative route-based attacks. We model the attack as a graph problem, where the city street network is represented by a directed graph, $G = (V, E)$. The set of vertices, V, is a set of intersections and the set of edges, E, is a set of directed road segments representing the ability to move between the intersections. Furthermore, we have a non-negative edge weights $w : E \to \mathbb{R}_{\geq 0}$ denoting the path metric. We set the length of a path in G to be the sum of the weights of its edges, therefore the length of the shortest path will have minimum total weight. Also, each edge has a cost $c : E \to \mathbb{R}_{\geq 0}$ of being removed from G. Note that the metric associated with the path can be seen as the attacker's objective, as the attacker wants to prevent the victim to achieve its goal of traveling the shortest path according to path metric.

The attacker is also given two nodes $(s, d \in V)$, and has the goal of routing traffic from s to d along a given path $(p*)$. The attacker removes edges with full knowledge of G, *w*, and c. Given a budget (b) the attacker's objective is to remove a set of edges $E' \subseteq E$ such that $\sum_{\in E'} c(e) \leq b$ and $p*$ is the exclusive shortest path from s to t in the resulting graph $G' = (V, E \setminus E')$. This is the Force Path Cut Problem, defined by [14] and adapted to directed graphs for our application.

We used two different methods of assigning weights to each edge capturing different attacker objectives:

1) LENGTH: the length, in meters, of the road segment
2) TIME: the time, in seconds it takes to travel the road segment (assuming the driver is going the speed limit)

$$\text{TIME} = roadLength/speedLimit \quad (1)$$

We used three methods of assigning the cost of removing an edge, which in this context practically means causing a large enough interruption on the road segment to shut it down, at least temporarily:

1) UNIFORM: each road segment costs 1 to remove
2) LANES: each road segment costs the number of lanes it has to remove
3) WIDTH: each road segment costs its width divided by the average width of a car in the USA [21] to remove

$$\text{WIDTH} = roadWidth/widthOfAverageCar \quad (2)$$

III. Experimental Results

In this section, we describe our experimental methodology and results. We seek to answer the following questions:

- *How can a motivated attacker leverage existing tools and public-domain information (maps of metropolitan areas, city regulations in response to traffic accidents, etc.) to plan sophisticated alternative route-based attacks on metropolitan areas?*
- *What would the design of such sophisticated alternative route-based attack strategies look like?*
- *What are factors that influence the effectiveness of the attack?*

A. Experimental Methodology

We compare four different algorithms adapted from [14]. We changed them to work for directed graphs and to use several weight and cost options. The algorithms are:

1) LP-PathCover: Linear programming optimization approach
2) GreedyPathCover: Iteratively cuts the road segment, not in $p*$, that is part of the most routes shorter than $p*$
3) GreedyEdge: Iteratively cuts the shortest road segment, not in $p*$, on the current shortest route between the source and destination
4) GreedyEig: Iteratively cuts the road segment, not in $p*$, on the current shortest route between the source and destination with the highest eigenvalue to cost ratio

Attacker Goal and Cost. Our experiments were focused on comparing the effectiveness and the efficiency of the four Force Path Cut approximation algorithms, the options for assigning edge weights and edge removal costs, and the impact of the graph topology of the city street networks. We believe

TABLE I
CITY GRAPH SUMMARIES

City	Nodes	Edges	Avg. Node Degree
Boston	11171	25715	4.60
San Francisco	9659	269002	5.57
Chicago	29299	78046	5.33
Los Angeles	51716	141992	5.08

that the TIME option for assigning weights is much more reflective of real travel time, so we used the LENGTH option as a baseline for comparison. Similarly, for edge removal cost we implemented the UNIFORM option as a baseline to compare to the more realistic WIDTH and LANES options. Although, the method for assigning edge removal costs depends on the budget and scale of the disruptions the attacker can cause. For example, if they can cause large disruptions the UNIFORM option might be more realistic because all they need is one interruption to shut down a road segment.

Datasets. We used the Boston, San Francisco, Chicago, and Los Angeles street networks gathered from OpenStreetMaps and represented as NetworkX DiGraphs [16] [11]. We show the characteristics of the corresponding graphs in Table I.

Source and Target selection. The source is a randomly selected intersection and the destination is a hospital. We used four major hospitals in Boston, San Francisco, Chicago, and Los Angeles each for our experiments. Unfortunately, due to the nature of OpenStreetMap data, points of interest, including hospitals, frequently lie outside of any road segment [16]. In such cases, we find the closest point on the road by calculating the straight-line distance in the corresponding geographical projection. We create an artificial node on that point. We then join the point of interest with the artificial node on the road segment and connect it with an artificial road segment. We mark an attribute in the geodataframe for this road segment to indicate the artificiality.

Alternative Route. The alternative path is set to the 100th shortest path between the source and destination, which we refer to as the path rank.

Metrics. In each set of experiments we randomly choose 10 different source nodes for each hospital, 40 experiments per set total. All averages in this section are over 40 experiment sets. We use the following metrics:

- Average Algorithm Time (Avg. Runtime): The average time the algorithm took to run one experiment (in seconds)
- Average Number of Edges Removed (ANER): The Average number of edges (road segments) removed to ensure the specified alternative route, $p*$, is the shortest path between the source and destination
- Average Cost of Removed Edges (ACRE): The Average cost of removed edges (road segments) to ensure the specified alternative route, $p*$, is the shortest path between the source and destination
- Avg. Inc. to 100th/200th path: the average percent increase of the path length (using the TIME weight) between the shortest path between the source and destination and the 100th/200th shortest path between the source and destination.

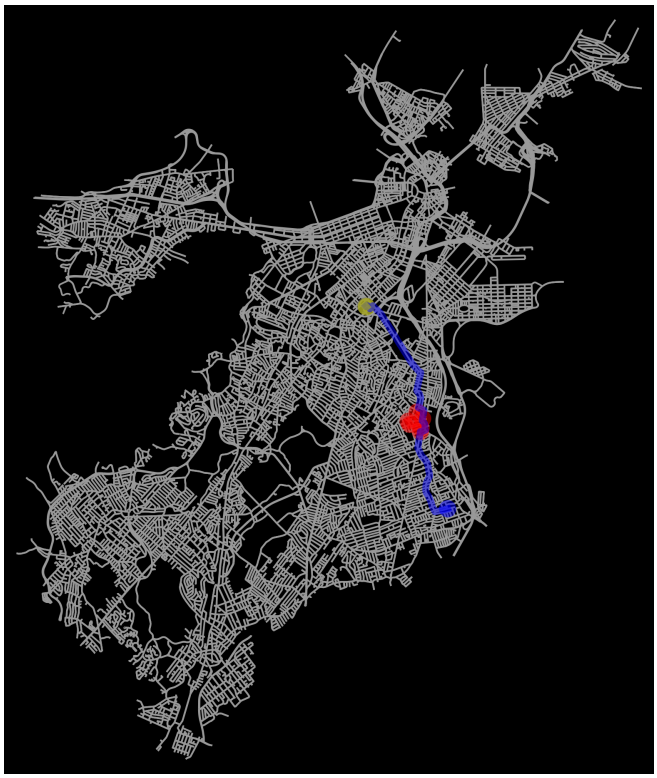

Fig. 1. A Boston experiment using Brigham and Women's hospital as the target destination, source was randomly selected. LENGTH is the weight type and WIDTH is the cost type.

Fig. 2. A San Francisco experiment using UCSF Medical Center at Mission Bay as the target destination, source was randomly selected. LENGTH is the weight type and WIDTH is the cost type.

TABLE II
BOSTON, WEIGHT TYPE: LENGTH

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	6.31	4	4	58.31	3.75	5	72.27	3.53	7.38
GreedyPathCover	2.83	4	4	6.72	3.78	5.03	6.09	3.53	7.38
GreedyEdge	1.03	4.5	4.5	3.78	5.25	6.5	2.64	4.5	9.42
GreedyEig	1.86	5	5	4.99	4.65	7.65	4.07	4.75	9.37

TABLE III
BOSTON, WEIGHT TYPE: TIME

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	66.82	3.78	3.78	21.17	4.18	6.6	19.56	3.58	7.48
GreedyPathCover	5.76	3.78	3.78	4.25	4.15	6.55	4.33	3.58	7.48
GreedyEdge	2.02	4.65	4.65	1.56	4.48	6.9	1.66	4.38	9.16
GreedyEig	3.22	4.65	4.65	2.77	4.48	8.33	2.92	4.4	9.21

TABLE IV
SAN FRANCISCO, WEIGHT TYPE: LENGTH

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	37.4	3.68	3.68	85.35	4.18	5.38	48.4	3.65	7.64
GreedyPathCover	6.44	3.68	3.68	5.81	4.43	5.68	5.74	3.65	7.65
GreedyEdge	2.2	6.58	6.58	2.14	7.5	8.45	2.33	6.28	13.13
GreedyEig	3.6	5.78	5.78	3.35	5.93	8.58	3.56	5.05	10.57

TABLE V
SAN FRANCISCO, WEIGHT TYPE: TIME

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	42.64	3.93	3.93	56.5	4.88	6.1	42.56	3.88	8.11
GreedyPathCover	4.98	3.9	3.9	5.57	4.85	6.1	4.85	3.88	8.11
GreedyEdge	1.36	4.48	4.48	1.56	6.18	7.48	1.12	4.68	9.78
GreedyEig	2.49	5.43	5.43	2.44	5.78	8.33	2	4.93	10.31

B. Results

Impact of Attacker's Objective. We observed that the attacker's objective (i.e. edge weight type) did not drastically affect the average number of edges removed or the average cost of removed edges, as seen in Table IX. For example, for all the Boston experiments LENGTH averaged 4.27 removed edges and 6.27 as the average cost of removed edges while TIME averaged 4.17 removed edges and 6.54 as the average cost of removed edges, as seen in Table IX. Furthermore, For all the San Francisco experiments LENGTH averaged 5.03 removed edges and 7.23 as the average cost of removed edges while TIME averaged 4.73 removed edges and 6.84 as the average cost of removed edges, as seen in Table IX. The goal of assigning weights to road segments was to determine how long those roads would realistically take to travel, therefore we wanted to use a method of assigning weights that was as realistic as possible, while remaining topological. In the end, we concluded that TIME was the best method of assigning weights because it was the most realistic representation of real travel time because LENGTH only represented the actual length of the road segment.

Impact of Attack Cost. We observed a clear increase in the average cost of removed edges across the different edge removal cost options (UNIFORM, LANES, and WIDTH), as seen in Tables II to VIII. The UNIFORM option was always going to be the cheapest because it simply assigns a cost of 1 to every edge, then the LANES option was the next most expensive because it included larger costs for removing multi-lane roads, and the WIDTH option was the most expensive because the width of an average American car is smaller than the width of an average lane. For example, in Table V the average cost of removed edges using UNIFORM cost was 4.43, using LANES cost was 6.84, and using WIDTH cost was 9.08. Therefore, an attacker should choose whatever option reflects their ability to cause an interruption. For example, if the attacker can cause large interruptions the UNIFORM option might make sense because they only need one interruption to shut down a road segment. If the attacker can only cause smaller interruptions, such as having a small car feign a breakdown on the road, the LANES option might be more realistic. Also, an attacker should consider their budget when choosing a cost type. For example, if they have an extremely large budget they might want to use WIDTH or LANES even if they can cause large interruptions just to be safe. However,

TABLE VI
CHICAGO, WEIGHT TYPE: LENGTH

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	125.21	3.58	3.58	175.51	3.5	7.33	199.8	3.85	5.15
GreedyPathCover	11.33	3.6	3.6	12.46	3.53	7.38	9.91	3.93	5.2
GreedyEdge	4.82	5.08	5.08	5.88	5.7	11.93	4.9	6.43	7.73
GreedyEig	5.34	5.18	5.18	6.4	4.7	9.84	5.41	5.23	8.55

TABLE VII
CHICAGO, WEIGHT TYPE: TIME

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	41.38	3.5	3.5	52.77	3.73	7.8	41.83	3.73	4.55
GreedyPathCover	8	3.5	3.5	8.41	3.73	7.8	7.3	3.73	4.55
GreedyEdge	1.51	4.1	4.1	1.53	4.18	8.74	1.6	4.58	5.4
GreedyEig	2.12	4.5	4.5	2.16	4.6	9.62	2.15	4.4	7.03

TABLE VIII
LOS ANGELES, WEIGHT TYPE: TIME

Algorithm	UNIFORM			LANES			WIDTH		
	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE	Avg. Runtime	ANER	ACRE
LP-PathCover	85.77	3.71	3.71	66.8	3.8	7.95	34.85	4.04	7.14
GreedyPathCover	22.13	3.73	3.73	22.51	3.8	7.95	11.09	4.01	7.16
GreedyEdge	5.11	4.51	4.51	4.98	4.5	9.42	2.75	4.51	9.15
GreedyEig	8.73	4.51	4.51	8.31	4.48	9.37	3.88	4.51	9.15

TABLE IX
AVERAGE ANER AND ACRE ACROSS ALL CITY AND WEIGHT TYPE COMBINATIONS

City	LENGTH		TIME	
	ANER	ACRE	ANER	ACRE
Boston	4.27	6.27	4.17	6.54
San Francisco	5.03	7.23	4.73	6.84
Chicago	4.52	6.71	4.02	5.92
Los Angeles	4.35	7.23	4.18	6.85

TABLE X
THRESHOLD TABLE, WEIGHT TYPE: TIME

City	Avg. Incr. to 100th path	Avg. Incr. to 200th path
Boston	7.93%	9.54%
San Francisco	4.23%	5.35%
Chicago	1.58%	1.93%

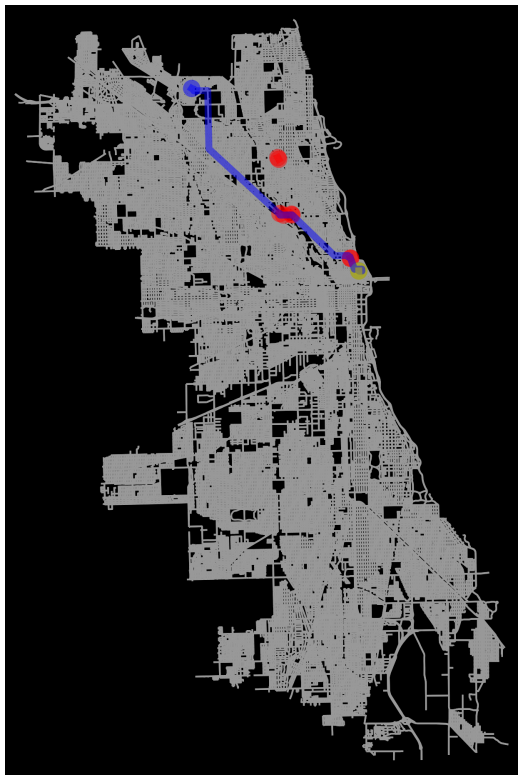

Fig. 3. A Chicago experiment using Northwestern hospital as the target destination, sources was randomly selected. LENGTH is the weight type and UNIFORM is the cost type.

if the attacker has a small budget they might need to use UNIFORM even if it does not reflect the size of interruptions they can cause.

Impact of Attack Algorithms Effectiveness. The time it takes the attacker to compute the attack strategy of what edges should be removed is a critical point for the feasibility of the attack as they determine the needed computational resources and time to obtain an effective strategy. While all the algorithms were effective enough to come up with viable solutions, the more intelligent algorithms often found solutions half the cost of the naive algorithm's solutions but took much longer to run, as seen in Tables II to VIII. The GreedyPathCover Algorithm was the most effective without taking prohibitively long to run, 2.83 to 22.51 seconds on average, as seen in Tables II and VIII. The LP-PathCover

Fig. 4. A Los Angeles experiment using LA Downtown Medical Center as the target destination, source was randomly selected. TIME is the weight type and LANES is the cost type.

algorithm consistently found the same or marginally cheaper solutions than the GreedyPathCover algorithm, but took about 5 to 10 times longer to generate its solution. The GreedyEgde and GreedyEig algorithms were consistently the quickest, 1.03 to 8.73 seconds on average as seen in Tables II and VIII, but their solutions were much more expensive than the LP-PathCover and GreedyPathCover algorithms.

Impact of City Traffic Networks. Throughout these experiments, we tested these algorithms on four different cities; Boston, San Francisco, Chicago, and Los Angeles. We used LP-PathCover as a near-optimal baseline for the other algorithms, [14] reported it found the optimal solution in terms of cost of removed edges in over 98% of experiments. We observed for more lattice city networks, such as Chicago, naive algorithms (GreedyEdge and GreedyEig) were able to find close to the optimal average cost of removed edges despite their relatively simple and fast approaches, as seen in Tables VI and VII. However, for less lattice cities, such as Boston, there was a more noticeable gap in the average cost of removed edges between the naive and intelligent algorithms, as seen in Tables II and III. This is largely because in less lattice cities, such as Boston, there is a much bigger gap between the length of the shortest path and the 100th shortest path between a source and destination, as seen in Table X. This larger gap in path length allowed the intelligent algorithms to use their superior capabilities to find the ideal edges to remove, while the naive algorithms we used either removed more edges or more expensive edges to compensate, as seen in Tables II and III. But for more lattice cities, such as Chicago, there is a smaller gap between the length of the shortest and 100th shortest path, as seen in Table X, which meant the naive algorithms were more likely to choose better edges to remove because there were so many near-optimal edges to remove. Our results show that the gap in the average cost of removed edges between the baseline naive algorithms (GreedyEdge and GreedyEig) and the optimization-based algorithm (LP-PathCover) was 2.3 for Boston, a less lattice street network, and 1.4 for Chicago, a very lattice street network, as seen in Tables II, III, VI, and VII. That means that the gap in attack cost between the naive and optimization algorithms for Boston was 1.6 times bigger than for Chicago. Therefore, the much faster baseline naive algorithms were much more efficient for more lattice cities, making it quicker and cheaper to attack more lattice cities.

As we stated in the introduction, an attacker should choose the algorithm they use by considering how lattice the street network is, their budget, and the time they have to run the algorithm. For example, if the city network is less lattice and the attacker has limited resources they may need to use the optimization-based algorithm, LP-PathCover, to find a viable solution, which takes more time. If the street network is very lattice then using much faster naive algorithms could be near as effective as LP-PathCover, which might be useful if the attacker has limited time to plan their attack.

C. Examples

Figures 1, 2, 3, and 4 depict visualization examples for Boston, San Francisco, Chicago, and Los Angeles. Each visualization shows the results from a single experiment, where the blue circle is the source, the yellow circle is the destination (a hospital), the blue path is the chosen alternative route, $p*$, between the source and destination, and the red roads are the ones that were removed to make the chosen alternative route the shortest path between the source and destination. We hope that these images provide helpful visualizations of interesting individual experiments.

In Figure 4, it is clear that a few edges above the blue path had to be removed so that the blue path could become the shortest path between the source and destination and that the original shortest path was very different than the chosen alternative route. Whereas, in Figure 2, the alternative path was clearly close to the original path except for a few roads removed around the middle of the path.

IV. Related Work

Lots of research has been done on attacks on city networks of connected and autonomous vehicles, especially on the hardware and software needed for these new vehicles and the topological properties of the city networks in conjunction with the new stresses of integrating connected and autonomous vehicles. For example, Jagielski et al. [13] conducted a detailed analysis of various attacks that a motivated attacker can

launch on a vehicle's adaptive cruise control by influencing acceleration reported by another car's LIDAR or RADAR sensors. They studied two attacks that are capable of causing a crash in a platoon. DeBruhl et al. [6] studied vehicle platoon misbehavior. They considered the design of a set of insider attacks and abnormal behavior attacks that can occur in a CACC platoon of vehicles. In particular, they studied the *collision induction attack*, in which an attacker can exploit the platoon controller to cause a fatal crash with the vehicle that is following it.

On the topological side, Zhang et al. [22] examined how a transportation network's topological characteristics can indicate its ability to cope with disasters. They studied how the transportation network's topological attributes can significantly affect resilience under disasters. In another example of similar research, Feyessa et al. [8] conducted empirical analysis on various network robustness measures to study the contribution of a node to the overall street network's robustness. Changes in network-level measures, such as efficiency and average clustering, were studied when individual nodes were removed.

We utilized recent work by Miller et al. [14]. They introduced the Force Path Cut Problem, where an adversary wants to promote a specific route by removing a minimum number of edges in the graph. They showed that the problem was NP-complete and formalized it as an instance of the Weighted Set Cover problem. By using constraint generation, they overcame the potentially factorial size of the universe for the set cover problem. Their work focused on undirected graphs, while our work focuses on directed graphs.

V. Conclusion

We showed that a motivated attacker could quickly and efficiently determine how to manipulate vehicle(s) into traveling specifically chosen alternative routes. This type of attack could have a wide range of effects such as targeting specific vehicles and their passengers, forcing all vehicles traveling between common locations to pass through certain road segments (such as toll roads), targeting emergency vehicles, or slowing traffic traveling between common destinations. Several other semantically meaningful options for modeling the city networks, choosing attack targets, and alternative routes are possible and we plan to explore them in future work.

References

[1] Mani Amoozadeh, Arun Raghuramu, Chen-nee Chuah, Dipak Ghosal, H. Michael Zhang, Jeff Rowe, and Karl Levitt. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine*, 53(6):126–132, 2015.

[2] Baidu Apollo Team. Apollo: Open source autonomous driving. https://github.com/ApolloAuto/apollo, 2017.

[3] Mariusz Bojarski, Chenyi Chen, Joyjit Daw, Alperen Degirmenci, Joya Deri, Bernhard Firner, Beat Flepp, Sachin Gogri, Jesse Hong, Lawrence D. Jackel, Zhenhua Jia, B. J. Lee, Bo Liu, Fei Liu, Urs Muller, Samuel Payne, Nischal Kota Nagendra Prasad, Artem Provodin, John Roach, Timur Rvachov, Neha Tadimeti, Jesper E. van Engelen, Haiguang Wen, Eric Yang, and Zongyi Yang. The NVIDIA pilotnet experiments. *CoRR*, abs/2010.08776, 2020.

[4] Comma.AI. Openpilot Github Repository. https://github.com/commaai/openpilot, 2022.

[5] Soodeh Dadras, Ryan M. Gerdes, and Rajnikant Sharma. Vehicular platooning in an adversarial environment. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '15, page 167–178, New York, NY, USA, 2015. Association for Computing Machinery.

[6] Bruce DeBruhl, Sean Weerakkody, Bruno Sinopoli, and Patrick Tague. Is your commute driving you crazy? A study of misbehavior in vehicular platoons. In *Proceedings of the 8th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 22:1–22:11, 2015.

[7] Murat Dikmen and Catherine Burns. Trust in autonomous vehicles: The case of tesla autopilot and summon. In *2017 IEEE International Conference on Systems, Man, and Cybernetics*, SMC, pages 1093–1098, 2017.

[8] T Feyessa and M Bikdash. Measuring nodal contribution to global network robustness. In *Proceedings of the 2011 IEEE Southeastcon*, pages 131–135, 2011.

[9] Ryan M. Gerdes, Chris Winstead, and Kevin Heaslip. CPS: An efficiency-motivated attack against autonomous vehicular transportation. In *Proceedings of the 29th Annual Computer Security Applications Conference*, ACSAC, page 99–108, New York, NY, USA, 2013. Association for Computing Machinery.

[10] Jason Haas. The effects of wireless jamming on vehicle platooning. *ResearchGate*, 04 2009.

[11] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory, Los Alamos, NM (United States), 2008.

[12] Jeff Hawke. A new approach to self-driving: AV2.0. https://wayve.ai/blog/a-new-approach-to-self-driving-av2-0, 2021.

[13] Matthew Jagielski, Nicholas Jones, Chung-Wei Lin, Cristina Nita-Rotaru, and Shinichi Shiraishi. Threat detection for collaborative adaptive cruise control in connected cars. In *Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 184–189, 2018.

[14] Benjamin A. Miller, Zohair Shafi, Wheeler Ruml, Yevgeniy Vorobeychik, Tina Eliassi-Rad, and Scott Alfeld. PATHATTACK: attacking shortest paths in complex networks. In *Proceedings of the 2021 European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 532–547, 2021.

[15] U.S. Department of Transportation. What are connected vehicles and why do we need them? https://www.its.dot.gov/cv_basics/cv_basics_what.htm, 2021.

[16] OpenStreetMap Project (openstreetmap.org). Planet OSM. https://planet.osm.org, 2017.

[17] Kyle Stock. Gatik's self-driving delivery vans head to north to canada. https://www.bloomberg.com/news/articles/2020-11-23/gatik-s-self-driving-delivery-vans-head-north-to-canada, 2020.

[18] Kyle Stock. Walmart stomps the pedal on self-driving delivery. https://www.bloomberg.com/news/articles/2021-11-08/walmart-stomps-the-pedal-on-self-driving-delivery, 2021.

[19] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR, June 2020.

[20] Rens van der Heijden, Thomas Lukaseder, and Frank Kargl. Analyzing attacks on cooperative adaptive cruise control (cacc). In *2017 IEEE Vehicular Networking Conference*, VNC, pages 45–52, 2017.

[21] The Zebra. STUDY: Average car size is increasing — will roads still be safe for small cars and pedestrians? The Zebra: https://www.thezebra.com/resources/driving/average-car-size/, 2022.

[22] X. Zhang, E. Miller-Hooks, and K. Denny. Assessing the role of network topology in transportation network resilience. *Journal of Transport Geography*, 46:35–45, 2015.