A Cyber-Physical Testbed for Measuring the Impacts of Cyber Attacks on Urban Road Networks

Marielba Urdaneta, Antoine Lemay, Nicolas Saunier, Jose Fernandez

Chapter 10

A CYBER-PHYSICAL TESTBED FOR MEASURING THE IMPACTS OF CYBER ATTACKS ON URBAN ROAD NETWORKS

Marielba Urdaneta, Antoine Lemay, Nicolas Saunier and Jose Fernandez

Abstract Efficient and safe transportation of people and goods are key requirements in a modern economy. Traffic control systems are installed at complex intersections to ensure the safe and efficient flow of traffic. However, there are concerns that an adversary could launch cyber attacks that exploit flaws in traffic control systems to cause mayhem and accidents.

This chapter presents a co-simulation framework for cyber-physical systems that enables researchers to execute cyber attacks on traffic control systems and measure their impacts on road traffic. The approach integrates an emulated supervisory control and data acquisition master station with a microscopic traffic simulation tool that provides all the functions of a traffic signal control system. The impacts of cyber attacks on road traffic are measured from the outputs provided by the traffic simulation. Experimental results for a corridor of six coordinated signalized intersections are presented, where the impacts are measured in terms of vehicle travel time and queue length. The results reveal that the physical impacts of compromising a single intersection could be felt at other intersections in the road network. This type of emergent result could only have been observed using a co-simulation framework.

Keywords: Road networks, traffic control systems, cyber attacks, testbed

1. Introduction

Traffic congestion is a growing problem and road safety is a major issue in cities around the world [4]. Traffic congestion impacts the economy and the urban environment as well as the quality of life and health of inhabitants. To mitigate congestion, cities are constantly seeking measures that improve and expand their traffic infrastructures and public transportation systems.

A road traffic infrastructure comprises road networks and traffic control devices such as signs, markings and traffic signals, which regulate and control traffic at intersections. Traffic signals and sensors are often connected to centralized systems that collect real-time traffic data, which is analyzed in order to design and implement control strategies. The control strategies seek to optimize traffic conditions and increase network capacity and user safety. Also, they attempt to reduce delays, stops, fuel consumption and pollutant emissions.

Modern traffic signal control systems typically incorporate traffic light controllers, sensors, communications networks and a computer-based central system that controls traffic signals and monitors traffic conditions and equipment status [17]. However, as newer technologies are introduced, traffic signal control systems are exposed to increased cyber risks. For example, wireless technologies are used in modern communications networks and by traffic detection systems due to their low maintenance costs and high scalability [6, 19]. However, the cyber risks are also increased.

Despite its benefits, wireless technology renders traffic signal control systems vulnerable to cyber attacks. In particular, wireless communications networks can be accessed remotely. Once a communications network is accessed, the control network is exposed and vulnerable to exploitation as demonstrated by Cerrudo [3] and Ghena et al. [7]. In particular, the researchers exploited vulnerabilities related to weak or no authentication, absence of encryption and wireless access to network components and traffic light controllers. The researchers were able to control traffic signals by capturing and modifying wireless communications, sending fake data and commands to traffic light controllers and connecting to controllers in order to alter their programming.

The feasibility of cyber attacks on traffic control systems demands the investigation of their impacts on road congestion as well as the economic, environmental and social consequences. An experimental environment that faithfully reproduces cyber attacks on traffic control systems and their effects on road traffic would be most useful to municipal authorities, urban designers and homeland security personnel. The environment would support the evaluation of defensive strategies for communications and control networks, and help establish measures for mitigating the physical impacts of attacks. Furthermore, it would facilitate the determination of the best mitigation strategies based on attack impact, enhancing decision making during actual attacks.

This chapter describes a co-simulation-based testbed that enables these capabilities. The testbed incorporates a microscopic traffic simulation package and an emulated supervisory control and data acquisition (SCADA) master station that provides traffic control system functionality. The principal innovation is the creation of a low-cost, reusable and reconfigurable testbed that integrates road traffic control and traffic behavior simulation components to enable the evaluation of cyber attacks and their impacts on road traffic. Unlike other approaches that only include one of the two components, the co-simulation approach significantly enhances the evaluation of cyber security issues because attacks can be conducted against the central control station and the traffic light

controllers. Indeed, it is believed that this is the first cyber-physical testbed based on a co-simulation framework that has been created to advance security research activities in the road traffic control domain.

2. Traffic Control Systems

This section describes the key notions related to traffic control systems drawn from various sources [8, 11, 17, 18].

Road traffic comprises pedestrians, cyclists, vehicles, trucks and on-road public transportation systems that concurrently share public roads. The components form traffic movements (or traffic flows) when they move together on the same roadway and in the same direction. At an intersection, two or more traffic movements are considered to be in conflict when their trajectories cross each other at the same level. In such a situation, it is necessary to establish which traffic flow has priority over the other (e.g., yield- or stop-controlled intersections) and when each traffic flow is allowed in the intersection. This assignment is called priority or right-of-way.

Traffic signals are equipped with controllers that switch the lights that inform road users when they have the right to move. Controllers may also be connected to vehicle-presence and pedestrian-presence detectors for real-time adaptation to traffic demand, and to a traffic management center that monitors and controls road traffic conditions and equipment status at intersections.

Traffic signal controllers follow a set of rules that establishes the order in which right-of-way is assigned to the different traffic movements. In addition, the rules establish the duration of the green light for each movement. The element that contains all the rules is called the timing plan and is used by traffic engineers to regulate traffic. The timing plan incorporates control parameters such as cycle length, phases, splits and intervals. A cycle is a complete sequence of phases in which right-of-way is given to all the traffic movements. The time required to complete this sequence is called the cycle length. A phase is the part of a cycle that is assigned to a traffic movement or to multiple traffic movements simultaneously. The part of the cycle assigned to each phase is called a split. The portion of a cycle during which lights do not change is called an interval. Clearly, an attacker with the ability to alter controller configuration (i.e., the timing plan) could disrupt traffic flow.

Traffic signals operate as part of a coordinated system or as isolated nodes. When working in coordination with other signalized intersections, the time (or offset) between the beginning of the cycle of each successive signalized intersection is computed so that vehicles do not have to stop at intermediate intersections.

In contrast, isolated traffic signals are not coordinated and are oblivious to how neighboring intersections are configured. Traffic regulation at isolated intersections employs pre-timed control, actuated control or a combination of the two. Pre-timed traffic lights use pre-elaborated timing plans in which the numbers, sequences and durations of the phases are fixed. Pre-elaborated plans are computed based on historic traffic conditions at intersections. Actuated

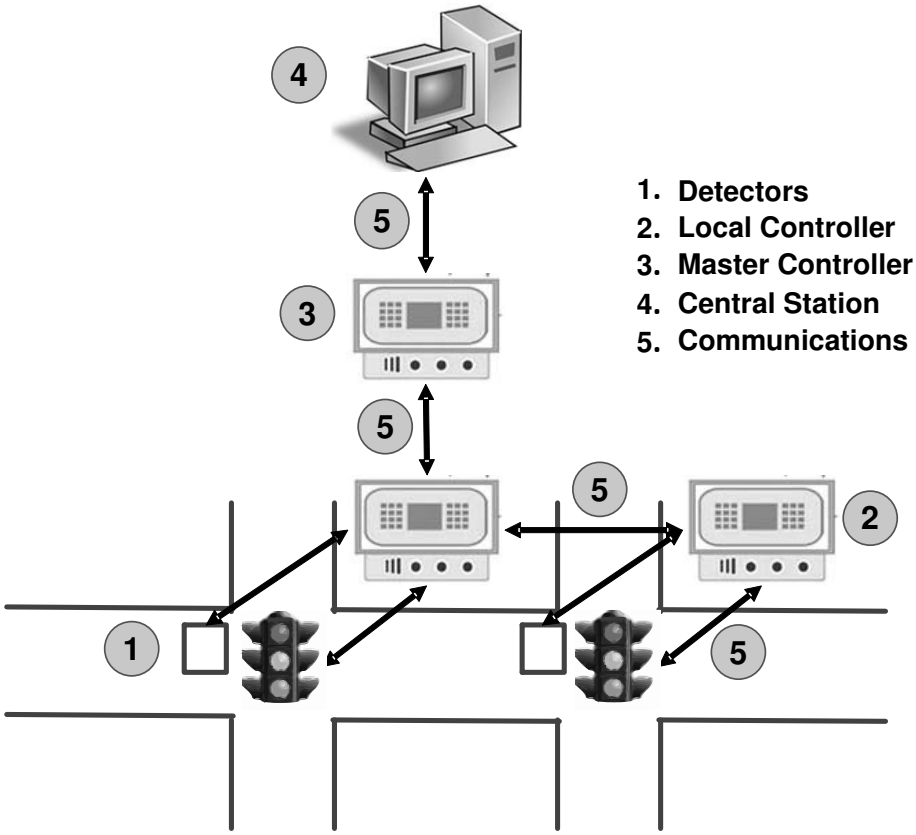

Figure 1. Traffic signal control system [11].

traffic lights use traffic condition information collected by sensors to activate phases when vehicles or pedestrians are detected.

Figure 1 shows the hardware components and architecture of a typical traffic signal control system. It comprises detectors, local controllers, on-street master controllers, a traffic management center and communications networks. Detectors are used to determine vehicle presence and pulse duration, which are needed to compute vehicle volume, occupancy, speed, etc. Local controllers are responsible for switching head lights at intersections using stored timing plans and schedules provided by operators. The controllers receive traffic data from detectors, process the data to obtain volume and occupancy parameters, and send the parameters to on-street master controllers.

Master controllers located at intersections are connected to all the local controllers belonging to the same control area to facilitate communications with the traffic management center. The master controllers are responsible for selecting traffic-responsive timing plans, processing and storing the data collected by detectors, and monitoring the equipment status at intersections.

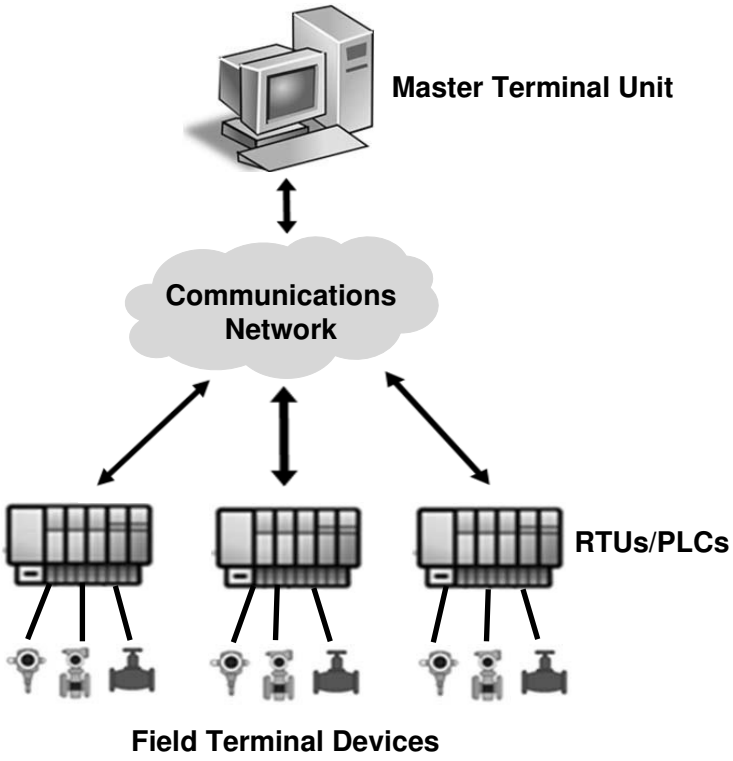

Figure 2. SCADA system.

They communicate with the traffic management center in the case of critical alarms, on a regular predetermined basis and when requested by operators.

The main functions of the traffic management center are to gather and display information about traffic conditions and intersection equipment status. In addition, it calculates the timing plans and schedules. After the timing plans and selection schedules are generated, they can be downloaded to on-street master controllers. Operators at the traffic management center can issue commands to master controllers, for example, to set the time or upload information saved in the master controllers.

The traffic signal control system has the same distributed architecture, control and monitoring elements as a SCADA network. Figure 2 shows a typical SCADA network for an industrial process. The SCADA network has a central station or master terminal unit (MTU) at the highest control level. The master unit processes the data collected from field devices, saves the data and displays it on a human-machine-interface (HMI) to enable operators to monitor and control the industrial process. The master terminal unit is connected to remote terminal units (RTUs) and/or programmable logic controllers (PLCs). The remote terminal units and programmable logic controllers are data ac-

quisition and control devices that are connected to measurement and control points in the field. They collect the measurement data, convert it to a suitable format and send it to the master terminal unit. Additionally, they pass commands from the master terminal to field devices. The communications network provides the required connectivity and data exchange functionality.

3. Related Work

This section discusses research related to traffic control system vulnerabilities, experimental scenarios for risk assessment and traffic control system threat assessment.

3.1 Traffic Control System Vulnerabilities

To demonstrate the exposure of control systems to cyber threats, Luallen [16] asked a group of cyber security students to study an industrial control system in order to find its known vulnerabilities and exploit them. The students leveraged the Internet to search for information about security flaws and proceeded to use a commercial cyber security training kit to launch attacks against the system. This work demonstrates that attackers do not require advanced expertise to attack cyber-physical systems. Valuable information about targets – including vulnerabilities – can be obtained from Internet resources such as technical reports, vendor websites and control system user forums. Having obtained information about a target, commercial products or open-source tools can be used to exploit the vulnerabilities.

Cerrudo [3] and Ghena et al. [7] have described several security flaws in traffic control systems currently deployed in the United States. Although they studied different systems, they findings were very similar: (i) lack of authentication or poor authentication mechanisms to prevent unauthorized access to traffic light controllers; (ii) lack of encryption of data and commands; (iii) use of default credentials supplied by vendors to access traffic light controllers and communications network devices such as switches, access points and repeaters; and (iv) authentication credentials published on vendor websites that are hardcoded in the systems and are not modifiable. Cerrudo and Ghena and colleagues demonstrated that they could gain access to system components and change traffic light states on command.

Krotofil and A.D. [14] state that launching a successful attack on a cyber-physical system involves five fundamental steps: (i) gain access to the system; (ii) discover the system; (iii) take control of the system; (iv) cause damage or disruption to the physical process; and (v) clean up all the evidence pointing to the cyber attack.

To illustrate their approach, Krotofil and A.D. created an experimental cyber-physical testbed that reproduced a traffic light control system for a four-way intersection. The testbed integrated a commercial control system and a cyber security training kit. Credentials provided by the vendor were used to gain access to the system. Having gained access, they acquired knowledge

about the system configuration and behavior using tools available on the system for diagnosis, development and visualization. Additionally, they reverse engineered binary files and communications messages to deduce information about the monitoring system and the corresponding elements in the physical system. This enabled them to manipulate the traffic lights at will. To ensure stealth, they manipulated system data so that operators would not notice the unauthorized changes to the traffic lights during the attacks.

The three research efforts demonstrate that flaws in deployed traffic signal systems could be exploited by adversaries. However, the research efforts did not measure the impacts of the attacks on traffic congestion and traffic safety.

3.2 Experimental Scenarios for Risk Assessment

Experimental setups based on co-simulation frameworks have been used to assess the security of various cyber-physical systems. Huang et al. [10] have employed such a framework to evaluate the impact of cyber attacks on a chemical reactor system. Their objective was to measure the impacts of the attacks on the physical process being controlled. Therefore, when conducting attacks, they modeled and monitored the chemical reactor so that they could determine the attacks with the greatest impact. Huang and colleagues discovered that, under steady-state conditions, attacks such as denial-of-service had minor impacts whereas the combination of denial-of-service and integrity attacks could damage the chemical reactor system. They also determined that the costs resulting from the attacks varied depending on the controllers and sensors targeted during the attacks.

Krotofil [13] has developed an open-source framework for controlling a chemical plant based on the well-known Tennessee Eastmann and Vinyl Acetate Monomer models. The previous Matlab models were redeveloped as Simulink models. Krotofil used the framework to develop cyber attacks that targeted sensors and actuators in the plant. Following this, she coupled it to the industrial control network and launched cyber attacks that captured and modified data and commands exchanged between the control system and physical plant.

Bernieri et al. [1] have used a co-simulation framework to evaluate the impacts of cyber attacks on the monitoring elements of a water supply control system. They conducted integrity and availability attacks on the water supply system and employed FACIES [9], an online fault detection and intrusion detection system, to evaluate attack detection performance. The experiments demonstrated that the fault diagnosis system was able to detect replay attacks and attacks that targeted the states of actuators. However, the system failed to identify flooding attacks and attacks that targeted sensor data. More significant was the fact that poor detection performance could induce operators to make unnecessary or erroneous decisions that negatively impacted the physical process.

Lemay et al. [15] have used co-simulation in a testbed that evaluates the effects of cyber attacks on the cyber and physical components of an electric power grid. They employed a virtualized cluster approach that emulates an informa-

tion technology network with high fidelity [2] and interfaced it with an electrical power flow simulator to model the industrial control network of an electrical grid. The testbed reproduced network attacks such as denial-of-service and data falsification (or injection) attacks, as well as malware infections. Moreover, it efficiently evaluated their impacts on the control network and the power grid.

Testbeds employing co-simulation frameworks are useful for modeling cyber-physical systems and evaluating the effects of cyber attacks. However, no such testbed has, as yet, been developed to assess the security of road traffic control systems.

3.3 Traffic Control System Threat Assessment

Ernst and Michaels [5] have presented a threat assessment framework that evaluates the impacts of vulnerabilities that provide access to field devices in a traffic control system. Their framework considers four access levels whose security flaws may be exploited: (i) vehicle detector; (ii) corridor synchronization; (iii) traditional Internet; and (iv) physical access. Ernst and Michaels employed the Simulation of Urban Mobility (SUMO) package [12] to simulate a road network comprising a corridor with six signalized intersections. They simulated attacks on the first three access levels and measured the attack impacts in various traffic demand scenarios.

Ernst and Michaels used the traffic simulation to investigate how attacks on traffic control system elements would impact road congestion. However, this simulation-only approach does not incorporate the important cyber component of the traffic signal control system. Since the resulting simulation has to rely on broad assumptions of the impacts of cyber attacks, it cannot be used to evaluate network defenses.

4. Testbed Functional Requirements

The goal of this research was to develop an experimental testbed that would enable security researchers to execute cyber attacks on traffic control systems and evaluate the impacts of the attacks on road traffic in real time. To accomplish this goal, it was decided to develop a co-simulation framework that incorporates a two-level distributed control system for an urban road network.

The co-simulation framework would couple a monitoring and control system (e.g., SCADA system) with a microscopic road traffic simulation. The SCADA system would provide the real-time monitoring and control functions required for a large road network. The microscopic traffic simulation would model a road network and traffic conditions to support the development of road traffic control strategies. Additionally, the microscopic traffic simulation would provide data about various road network entities such as pedestrians, vehicles, public transport systems and traffic lights at a suitable level of granularity.

The traffic simulation must provide adequate outputs that enable measurements of the economic, environmental and social effects of road congestion

resulting from cyber attacks on the modeled road network. Example outputs include fuel consumption, greenhouse gas emissions, pollutant emissions, noise emissions, vehicle densities, vehicle travel times and vehicle waiting times. All this information could be provided by the microscopic traffic simulation.

Finally, a mechanism must be incorporated that properly couples the cyber and physical components of the traffic control system. This mechanism would handle the time difference between the supervisory and control system sampling time and the traffic simulation step time (if any). Additionally, it would support seamless data exchange between the control system and road traffic simulation.

5. Testbed Architecture

The testbed is designed to support research activities by the cyber security community. To ensure availability, reusability and adaptability, a number of open-source software applications were employed to construct the testbed. Figure 3 presents the testbed architecture and components.

5.1 Monitoring and Control

The high-level control component of the system was reproduced using the ScadaBR 1.0 CE open-source SCADA software, a web-browser-based application that supports access to monitoring, control and automation equipment using various protocols (`www.scadabr.com`). In particular, ScadaBR: (i) provides the monitoring and control functions of a master terminal unit; (ii) displays and saves information about traffic conditions and traffic light states received from the low-level control system; (iii) enables operators to send commands to change traffic light operation modes (e.g., NORMAL/DISABLE); and (iv) runs a Modbus client to communicate with each control and data acquisition device in the low-level control system. ScadaBR can be configured to implement all the functions of a traffic management center that monitors and controls several traffic lights.

The low-level control system was implemented using Python scripts that emulated programmable logic controller functions. The scripts read master terminal unit commands and road network data, converted the data to the proper format and transmitted it to the required control system level. Moreover, the scripts implemented the logic that controlled traffic signals in the network, thereby acting as traffic light controllers. Programmable logic controllers were designed to control all the traffic signals at each signalized intersection. Each programmable logic controller script ran a Modbus/TCP server to communicate upstream with ScadaBR using the Modbus/TCP server functionality provided by the Modbus TK Python library. In addition, each programmable logic controller ran a TCP client to communicate downstream with the road traffic simulation via a communications server.

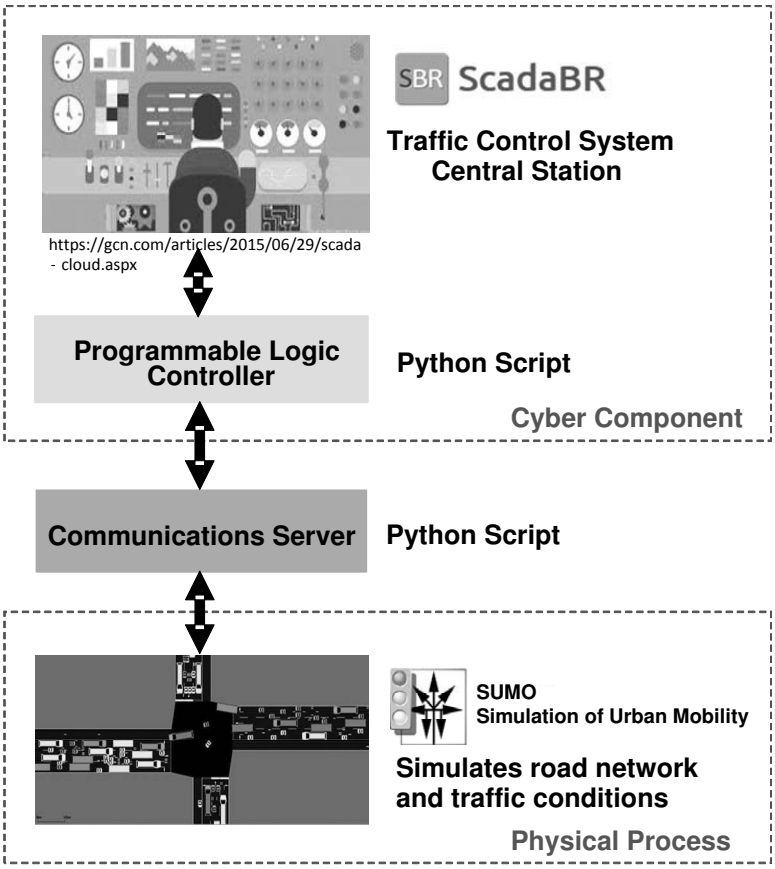

Figure 3. Testbed architecture and components.

5.2 Road Traffic Simulation

The physical process controlled in the testbed is road traffic. The open-source SUMO package developed by the German Aerospace Center [12] was employed to simulate road traffic. SUMO offers the flexibility of creating large-scale road networks from common formats such as shapefiles and Open Street Map files. A SUMO road network incorporates signalized intersections and traffic light plans. Additionally, origin/destination matrices can be converted to single vehicle trips and loaded in the SUMO simulation.

At each time step, SUMO generates outputs that provide information about all the simulated elements in the road network, including vehicles, intersections, roads, lanes, traffic lights and inductive loops. Data produced at this level of granularity is adequate for the monitoring component. Also, SUMO generates noise emission, pollutant emission and fuel consumption outputs required to quantify the economic, environmental and societal effects of road congestion.

SUMO incorporates a Python traffic control interface (TraCI) for interacting with external applications via TCP socket connections. This enables SUMO to connect to other monitoring and control systems. The interface also enables users to set and modify the simulation conditions at any time. For example, the user could change vehicle speeds, driver behavior, road priority and traffic light state as well as force vehicles to change lanes. These features were used to enforce state changes dictated by the control component.

SUMO performs a discrete-time simulation with adjustable step durations from 1 ms and upwards. It also offers two simulation alternatives, one without visualization and the other with visualization via a graphical interface.

5.3 Communications Server

A Python TCP multi-threaded communication server was developed to couple the monitoring and control system with the physical process. Multi-threading enabled the server to handle and serve multiple concurrent incoming client requests at the same time. Moreover, it dealt with communications synchronization issues arising from differences between the programmable logic controller sampling interval and the simulation time step.

At every simulation step, the server received data and requests from SUMO and the programmable logic controllers. The data received from SUMO pertained to each signalized intersection and its traffic light states provided by the simulation. This data was stored in separate tables according to the signalized intersection and its programmable logic controller; the data was transmitted upon request to the corresponding programmable logic controller.

Data received from a programmable logic controller identifies the signalized intersection and the traffic light states set during the simulation. This data was stored in a table that matched each programmable logic controller with its signalized intersection. The data was transmitted to SUMO upon request.

The SUMO traffic control interface was employed to execute a script running a TCP client. At each simulation step, the client transmitted the simulation results to the server and requested new commands from the programmable logic controller. SUMO adjusted the state of the traffic lights according to the information received from the server.

6. Validation and Experimental Setup

This section discusses the initial validation of the co-simulation framework and the experimental setup.

6.1 Initial Validation

For configuration and testing purposes, a preliminary setup was created that connected all the components of the proposed co-simulation framework. This preliminary setup was used to validate: (i) proper integration of all the components; (ii) proper system operation; and (iii) correct conversion/transmission

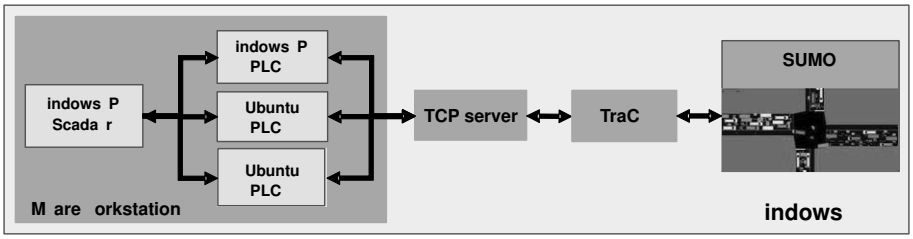

Figure 4. System used for initial validation.

of information from the master terminal unit to the traffic simulation, and vice versa.

The first simulation scenario involved a road network with three signalized intersections spaced 100 m apart and running in the pre-timed or semi-actuated mode. The traffic light control logic replicated the logic specified by Krotofil [14]. The control logic implemented a finite-state machine with eight states and nine transition conditions to model the traffic lights. It employed four control signals: (i) AUTO; (ii) DISABLE; (iii) MAIN ROAD; and (iv) SIDE ROAD. These signals enabled the traffic light operation modes to be set by the master terminal unit. When the operation mode was set to AUTO, the traffic lights commuted automatically based on the finite state machine program. In this case, the traffic lights operated in the pre-timed control mode with fixed control parameters; the timing plans could be changed by modifying the timing conditions and the state sequence in the finite state machine program. When the operation mode was set to DISABLE, the lights changed to yellow in all directions at the intersection; they remained in this state until the DISABLE signal was no longer set. When either the MAIN ROAD or SIDE ROAD signal was set, the traffic lights operated in the semi-actuated control mode. This assigned the green light to the corresponding road (MAIN or SIDE) until vehicles were detected on the opposite road.

All the system components were installed and configured on a desktop computer running the Windows 10 operating system (Figure 4). The SUMO software, simulation update script and communications server ran directly on the computer. ScadaBR and the programmable logic controllers executed in virtual machines. Specifically, ScadaBR and PLC 1 ran on Windows XP virtual machines whereas PLC 2 and PLC 3 ran on Ubuntu Linux virtual machines. All the virtual machines were created using VMWare Workstation software.

After validating the integration and operation of the preliminary setup, cyber attacks were launched to evaluate the fidelity of the testbed. For this purpose, a Kali Linux virtual machine was connected to the same network as the programmable logic controllers and ScadaBR. The Kali Linux machine was then used to conduct man-in-the-middle (MiTM) packet captures and packet injection attacks.

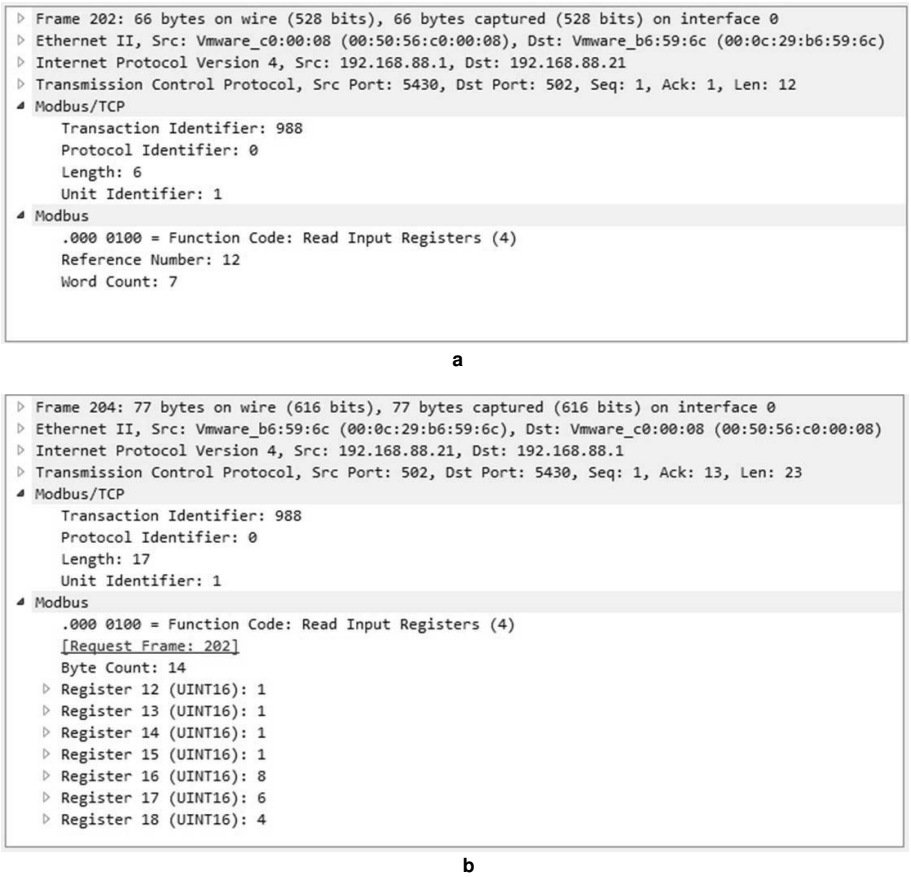

Figure 5. ScadaBR request and PLC 1 response during normal operations.

The scenario assumed that an attacker had gained access to the communications network and intercepted the data exchanged between the master terminal unit and the controller. Since the Modbus protocol does not incorporate authentication and encryption mechanisms, the attacker could inject control packets that would be accepted by the traffic controller. Furthermore, with the help of Internet resources, it would be easy to reproduce the content of Modbus messages and create arbitrary control messages for transmission to the controller.

The man-in-the-middle packet capture attack was executed using a Python script, which performed an address resolution protocol (ARP) cache poisoning that targeted ScadaBR and PLC 1. The attack enabled the adversary to impersonate the ScadaBR and PLC 1, and intercept the messages exchanged by them. Figures 5(a) and 5(b) show a ScadaBR request and the corresponding PLC 1 response during normal operations, before ARP cache poisoning.

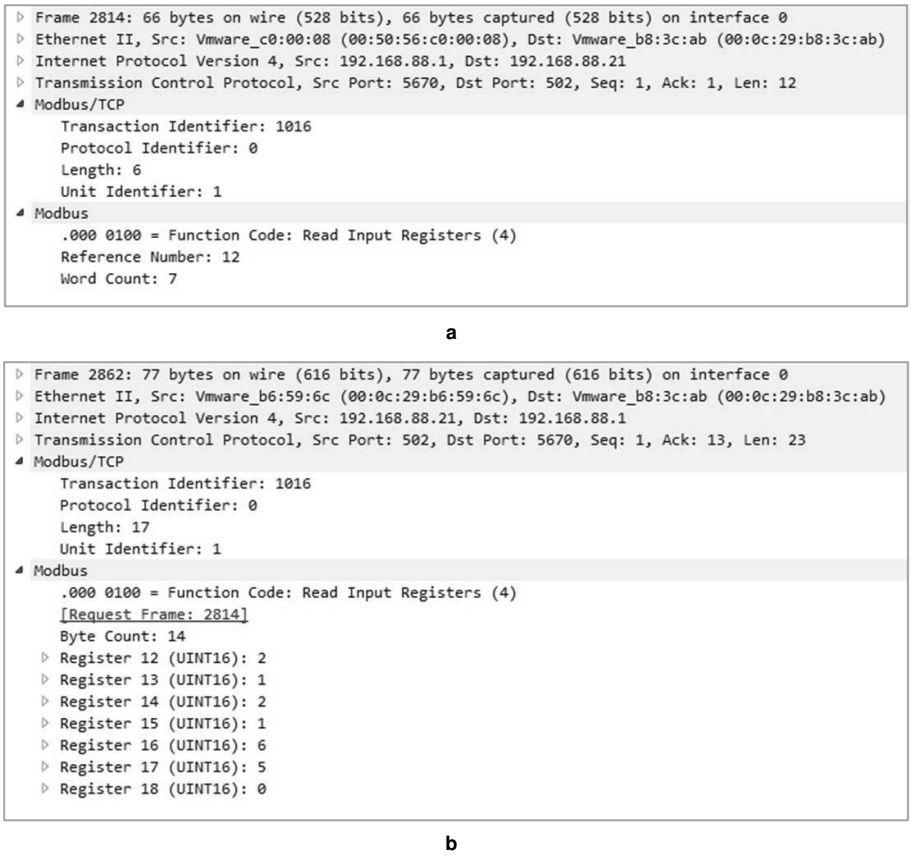

Figure 6. Intercepted ScadaBR request and PLC 1 response during the attack.

Figure 6(a) shows a request generated by ScadaBR and intercepted by the attacker (MAC address `00:0c:29:b8:3c:ab`) who impersonated PLC 1. Figure 6(b) shows the corresponding response generated by PLC 1 and intercepted by the attacker who impersonated ScadaBR.

The packet injection attacks were executed by a separate Python script that sent Modbus commands from the attacker's machine to PLC 1. Figure 7(a) shows a request generated by the attacker to set the mode of the traffic light to DISABLE (function code Write Single Coil and register reference number 3). Figure 7(b) shows the response generated by PLC 1 confirming the setting of the register value.

6.2 Experimental Setup

Following the initial validation, it was decided to execute a cyber attack on a coordinated traffic light system. This was accomplished by recreating the

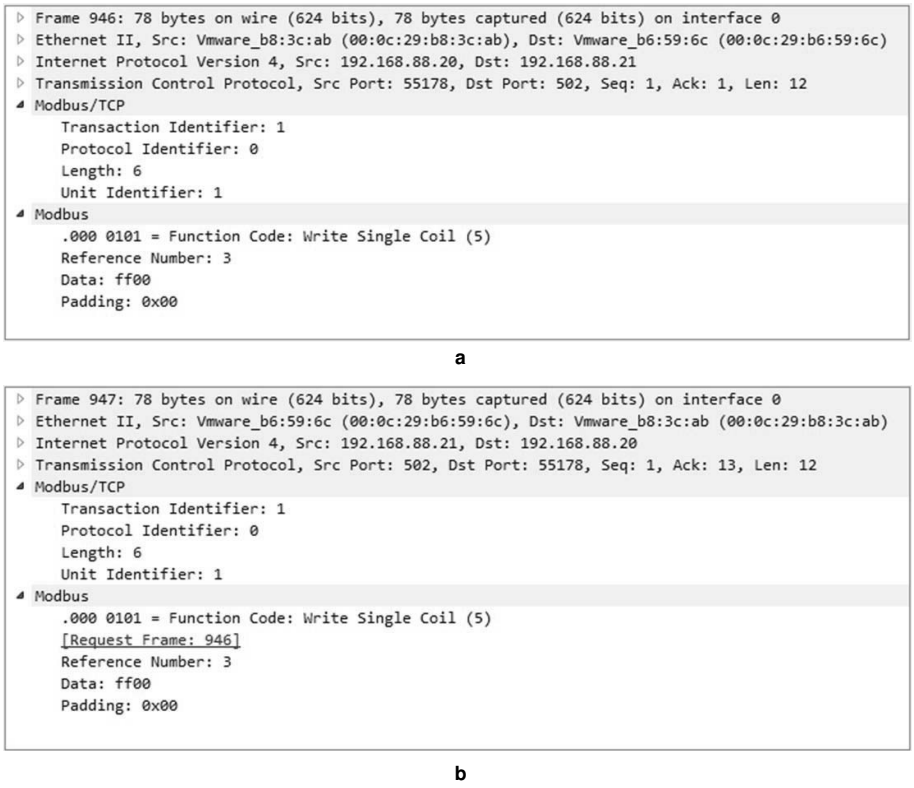

Figure 7. Messages exchanged during the packet injection attack.

road corridor used by Ernst and Michaels [5]. The experimental setup shown in Figure 8 comprised six coordinated signalized intersections, each spaced 100 m apart. An additional intersection was placed 2,000 m from the east entry of the corridor to generate vehicle platoons. As in the case of the Ernst and Michaels model, no turns were allowed and, to keep the model simple, each road had only one lane in each direction. Nonetheless, the model was adequate to demonstrate the impacts of the attacks on a corridor of signalized intersections.

The corridor in the experimental setup was coordinated to favor eastbound flows. Table 1 shows the simulation parameters. Table 2 shows the timing plan parameters used in the experimental setup.

In order to achieve coordination in the corridor, intersection C1 was chosen as the master intersection. Intersections C2 through C6 were coordinated with offsets of 5.8 s, 11.6 s, 17.4 s, 23.2 s and 29 s, respectively. One programmable logic controller was set up to manage intersection C1 while another was used to manage intersection C5, which was the target of the attacks. The control logic for the four remaining intersections (C2, C3, C4 and C6) was implemented by

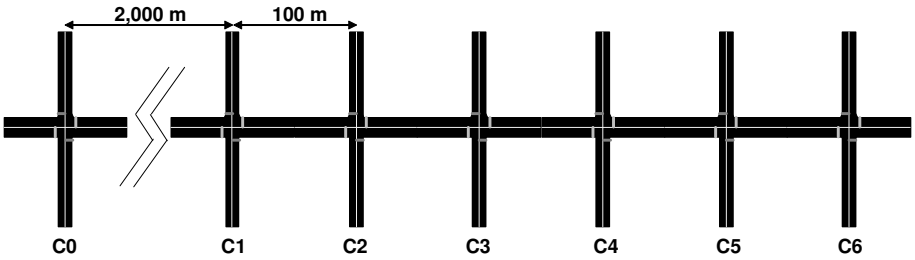

Figure 8. Road network used in the experimental setup.

Table 1. Traffic simulation parameters for various flows.

Parameter	**Eastbound Flow**	**Westbound Flow**
Maximum Speed	16.67 m/s	16.67 m/s
Acceleration	4.5 m/s^2	4.5 m/s^2
Deceleration	0.8 m/s^2	0.8 m/s^2
Length	5 m	5 m
Minimum Gap	2.5 m	2.5 m
Sigma	0.5	0.5
Demand	1,000 vehicles/h	500 vehicles/h
Car Following Model	Krauss	Krauss

Table 2. Timing plan parameters for the coordinated corridor.

Cycle Length	98 s
Main Road Green Duration	60 s
Side Road Green Duration	20 s
Yellow Duration	6 s
All Red Duration	3 s

SUMO instead of simulated programmable logic controllers. The decision not to use fine-grained emulation for these four intersections was made to conserve computing resources. There is no loss of generality because nothing, apart from computational power, would prevent the virtualization of all the programmable logic controllers if they were required.

After configuring the corridor, packet injection attacks were launched on signalized intersection C5. The same Kali Linux machine and script used in the initial validation were used to send Modbus/TCP messages to change the programming of the traffic lights at the intersection. Specifically, the main green light time was changed to 22 s and the side green light time was changed to 10 s.

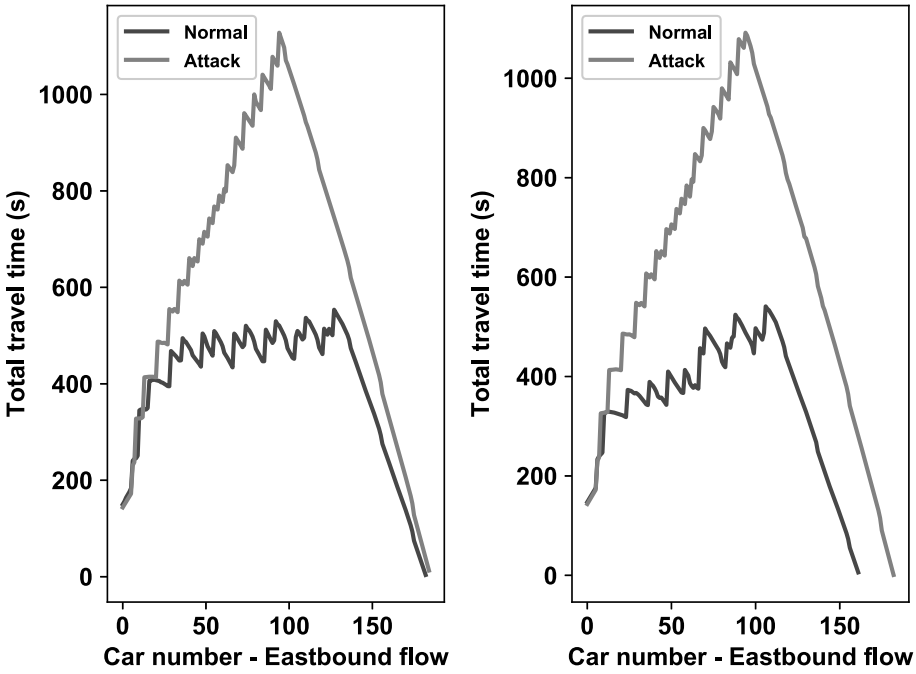

Figure 9. Eastbound vehicle travel times for two simulation runs.

7. Experimental Results

The attack impacts were measured in terms of travel time and queue length. The travel time for each vehicle in the main corridor in the eastbound direction and going through all the intersections was recorded. The travel time was plotted as a function of vehicle number in the order of vehicles entering the intersection. The queue lengths were measured at each simulation step and reported for each corridor section. Following this, the mean queue length for each section was computed based on the results of five simulation runs.

Figure 9 shows the travel time results for two simulation runs under normal conditions and during the attacks.

Figure 10 shows the mean values of the queue length for each corridor section between intersections C0 and C6 under normal conditions and during the attacks.

The results reveal that the travel times increased two to three times during the attacks. The queue lengths increased even more – they were practically non-existent under normal conditions (about two vehicles at most intersections) and increased four to five times (up to eleven vehicles). The effect on queue length was greater for intersections in the middle of the corridor, with queue spillback from downstream intersections.

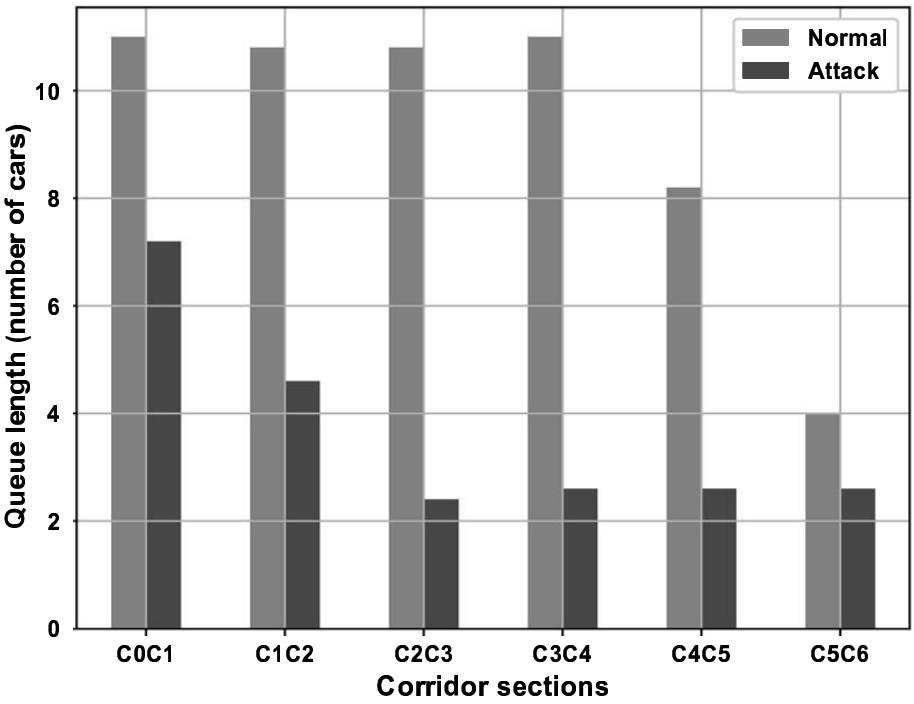

Figure 10. Mean queue length values for each corridor section.

The results also demonstrate that the co-simulation approach is very useful for evaluating the physical impacts of real cyber attacks. Moreover, unlike the work by Ernst and Michaels [5], no assumptions had to be made about the effects of cyber attacks on the traffic light control components.

8. Conclusions

The testbed described in this chapter successfully integrates a microscopic traffic simulation with an emulated SCADA master station to reproduce a traffic control system for a coordinated corridor of signalized intersections. This testbed is well-suited for evaluating the impacts of cyber attacks on traffic control systems. The impacts were measured in terms of traffic performance measures such as travel time and queue length rather than information technology performance metrics. In the man-in-the-middle attack scenario considered in the experiments, the travel time was increased two to three times and the vehicle queue length was increased four to five times over normal operations. Moreover, attacking one intersection produced impacts at other intersections in the road network. The results highlight the importance of understanding the local and global impacts of cyber attacks that target road networks. These

emergent results could have only been observed in a co-simulation framework of the type implemented in the testbed.

The testbed incorporates generic simulation and control software. While this has supported the execution of certain attacks and the evaluation of their impacts, it limits investigations of complex attacks and their impacts. This limitation can be overcome by enhancing the fidelity and the capabilities of the testbed by modeling real-world road networks and traffic demand scenarios, and by replacing ScadaBR with real traffic control software. Nevertheless, the testbed can help identify the critical signalized intersections in road networks and the attacks that produce the greatest impacts on traffic conditions. This information can be used to implement security and mitigation strategies, as well as to develop plans for reducing the negative impacts of attacks on traffic performance.

Future research will employ the testbed to evaluate the resilience of road networks to cyber attacks. This will involve the modeling of large road networks and replicating advanced cyber attacks on centrally-controlled traffic control systems whose impacts cannot be evaluated using a traffic simulator alone.

References

[1] G. Bernieri, E. Etcheves Miciolino, F. Pascucci and R. Setola, Monitoring system reaction in a cyber-physical testbed under cyber attacks, *Computers and Electrical Engineering*, vol. 59, pp. 86–98, 2017.

[2] J. Calvet, C. Davis, J. Fernandez, W. Guizani, M. Kaczmarek, J. Marion and P. St-Onge, Isolated virtualized clusters: Testbeds for high-risk security experimentation and training, *Proceedings of the Third International Conference on Cyber Security Experimentation and Test*, 2010.

[3] C. Cerrudo, Hacking US (and UK, Australia, France, etc.) traffic control systems, *IOActive*, Seattle, Washington, April 30, 2014.

[4] G. Cookson and B. Pishue, INRIX Global Traffic Scorecard, INRIX Research, Kirkland, Washington, 2017.

[5] J. Ernst and A. Michaels, Framework for evaluating the severity of a cyber vulnerability of a traffic cabinet, *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2619, pp. 55–63, 2017.

[6] S. Faye, C. Chaudet and I. Demeure, Control of Urban Road Traffic by a Fixed Network of Wireless Sensors, Technical Report 2012D002, Telecom ParisTech, Paris, France, 2012.

[7] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek and J. Halderman, Green lights forever: Analyzing the security of traffic infrastructure, *Proceedings of the Eighth USENIX Workshop on Offensive Technologies*, 2014.

[8] R. Gordon and W. Tighe, Traffic Control Systems Handbook, Publication No. FHWA-HOP-06-006, Federal Highway Administration, Washington, DC, 2005.

[9] C. Heracleous, E. Etcheves Miciolino, R. Setola, F. Pascucci, D. Eliades, G. Ellinas, C. Panayiotou and M. Polycarpou, Critical infrastructure online fault detection: Application in water supply systems, *Proceedings of the Ninth International Conference on Critical Information Infrastructures Security*, pp. 94–106, 2014.

[10] Y. Huang, A. Cardenas, S. Amin, Z. Lin, H. Tsai and S. Sastry, Understanding the physical and economic consequences of attacks on control systems, *International Journal of Critical Infrastructure Protection*, vol. 2(3), pp. 73–83, 2009.

[11] P. Koonce, Traffic Signal Timing Manual, Publication No. FHWA-HOP-08-024, Federal Highway Administration, Washington, DC, 2008.

[12] D. Krajzewicz, G. Hertkorn, P. Wagner and C. Rossel, SUMO (Simulation of Urban Mobility) – An open-source traffic simulation, *Proceedings of the Fourth Middle Eastern Symposium on Simulation and Modeling*, pp. 183–187, 2002.

[13] M. Krotofil, Rocking the pocket book: Hacking chemical plants for competition and extortion, presented at *Black Hat USA*, 2015.

[14] M. Krotofil and A.D., Hack like a movie star: Step-by-step guide to crafting SCADA payloads for physical attacks with catastrophic consequences, presented at *ZeroNights*, 2015.

[15] A. Lemay, J. Fernandez and S. Knight, An isolated virtual cluster for SCADA network security research, *Proceedings of the First International Symposium on ICS and SCADA Cyber Security Research*, pp. 88–96, 2013.

[16] M. Luallen, Critical Control System Vulnerabilities Demonstrated – And What to Do About Them, InfoSec Reading Room, SANS Institute, Bethesda, Maryland, 2011.

[17] Ministry of Transportation Ontario, *Book 19 – Advanced Traffic Management Systems*, St. Catherines, Canada, 2007.

[18] Minnesota Department of Transportation, *Traffic Signals 101*, St. Paul, Minnesota, 2018.

[19] M. Tubaishat, Y. Shang and H. Shi, Adaptive traffic light control with wireless sensor networks, *Proceedings of the Fourth IEEE Consumer Communications and Networking Conference*, pp. 187–191, 2007.