A cyber-physical test bed to measure the impacts of cyber attacks in urban road networks

Marielba Urdaneta[(1)], Antoine Lemay[(1)], Nicolas Saunier[(2)], Jose M. Fernandez[(1)]
(1) Department of computer and software engineering
(2) Department of civil, geological and mining engineering
École Polytechnique de Montréal, Montreal, Canada
E-mail:
marielba-margarita.urdaneta-velasquez@polymtl.ca, antoine.lemay@polymtl.ca,
nicolas.saunier@polymtl.ca, jose.fernandez@polymtl.ca

Abstract: Efficient and safe transportation of people and goods is a key requirement for the economy to prosper. Traffic control systems are installed at complex intersections to ensure the safe and efficient flow of traffic. However, what if an adversary were to take advantage of the existing security flaws in traffic control systems to create a cyber attack? In this paper, we present a co-simulation framework for cyber-physical systems that allows researchers to reproduce computer-based attacks targeting traffic control systems and measure the impact of those attacks on road traffic. This solution inte7grates an emulated Supervisory Control and Data Acquisition (SCADA) system with a microscopic traffic simulation tool to provide the functions of a traffic signal control system. The impact of the cyber attacks on road traffic can be measured from the outputs provided by the traffic simulation. Experimental results for a corridor of six coordinated signalized intersections are presented, where the impact is measured in terms of vehicle travel time and queue length. 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 in such a co-simulation framework.

Keywords: Computer security; cyber-physical systems; road traffic control; control process networks.

1. INTRODUCTION

Traffic congestion is a growing problem and road safety remains an issue in many cities around the world [1]. Traffic congestion not only impacts the economy and the environment of cities, but also the quality of life and health of their inhabitants. To mitigate congestion, cities are constantly looking for measures to improve and expand their traffic infrastructure and public transportation systems. Traffic infrastructure not only comprises road networks, but also traffic control devices, such as signs, markings and traffic signals, which regulate and control traffic at intersections. Traffic signals and sensors can be connected to centralized systems responsible for collecting real-time traffic data, analyzing this data and implementing subsequent control strategies. These control strategies seek to optimize traffic conditions, increase network capacity and user safety. Moreover, they intend to reduce delays, stops, fuel consumption and pollutant emissions originating from traffic lights operations.

Current traffic signal control systems are typically integrated using traffic light controllers, sensors, communication networks and a computer-based central system controlling traffic signals and monitoring traffic conditions and equipment status [2]. However, as newer technology is introduced, the system is exposed to more cyber risks. For example, recent trends show that wireless technology is being increasingly used in both communication networks and traffic detection, due to its low maintenance costs and high scalability potential [3], [4].

Despite its benefits, wireless technology introduces some security risks that make traffic signal control systems vulnerable to cyber attacks. In particular, wireless communication networks are remotely accessible. Once the communication network is accessed, the control network is exposed and vulnerable to be hacked, as demonstrated by Cerrudo [5] and Ghena *et al.* [6]. They detected vulnerabilities related to the lack of authentication (or poor authentication mechanisms) while accessing both wireless network components and traffic light controllers, and the lack of

communication encryption. Due to these conditions, the researchers could control traffic signals by capturing and modifying wireless communication, sending fake data and commands to traffic light controllers, and connecting to controllers in order to alter their programming.

Imagine that an adversary takes advantage of existing security flaws in traffic control systems to create a cyber attack. How would the attack impact road congestion? What would be the economic, environmental and social consequences of such an attack? By having an experimental environment that faithfully reproduces computer-based attacks on traffic control systems and its effects on road traffic, municipal authorities could measure the impact of this kind of attack in a controlled and safe way prior to the occurrence of real attacks. As a result, they could effectively plan defense strategies to improve security in both communication and process networks and establish adequate measures to mitigate the physical impact of the attacks. Furthermore, it would help authorities determine and implement the best mitigation strategies, according to the impact of the attacks, thus facilitating adequate decision making during actual occurrences of these attacks.

To enable this capability, we developed a co-simulation framework that allows researchers to experimentally reproduce cyber attacks targeting traffic signal control systems, and to evaluate how they impact road traffic in cities. Our approach integrates a microscopic traffic simulation tool and an emulated Supervisory Control and Data Acquisition (SCADA) system to provide the functionalities of a traffic control system. The main purpose of this work is to offer an experimental mechanism to conduct computer security tests in the application domain of road traffic control and that allows quantification of the environmental, economic and social impact of the attacks. It is, to the best of our knowledge, the first cyber-physical test bed based on a co-simulation framework created to conduct computer security research in the road traffic control domain.

This article is organized as follows. We start by reviewing the background on traffic control. Next, in Section 3 we present previous research related to computer security of both process control

and traffic control systems as well as the usage of co-simulation-based frameworks to assess the impacts of cyber attacks in control process systems. The functional requirements and the architecture of the proposed test bed are explained in Sections 4 and 5 respectively. Section 6 describes the validation setup we developed and the experimental setup we used to execute the attacks. The experimental results are shown in Section 7. Finally, we present our conclusions and some insights for future work in Section 8.

2. BACKGROUND ON TRAFFIC CONTROL

This section provides some key notions of traffic control, collected from *Advance traffic management systems in the Ontario traffic manual* [2], *Traffic control systems handbook* [7], *Traffic signal timing manual 2008* [8] and *Traffic signals 101* [9].

Traffic is composed of pedestrians, cyclists, vehicles, trucks and on-road public transport that share public roads concurrently. They form traffic movements (or traffic flows) when they move together in the same way and direction. At intersections, two or more traffic movements are considered in conflict if their trajectories cross each other at the same level. In that case, it is necessary to establish which traffic flow has priority over the other (in yield or stop controlled intersections) or when each movement is allowed in the intersection. This assignment is called priority or right-of-way.

Traffic signals are equipped with controllers, which are responsible for switching the lights that indicate to 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 Centre (TMC) that monitors and controls road traffic conditions and equipment status in the intersection.

Traffic signal controllers follow a set of rules that establishes the order in which the right-of-way is assigned to the different traffic movements. In addition, the rules establish the green light time duration for each movement. The element that contains all those rules is called the *timing plan*, whose design and use constitutes the technique most commonly used by traffic engineers to regulate traffic. Timing plans contain control parameters, such as cycle length, phases, splits and intervals. A cycle is a complete sequence of phases, in which the right-of-way has been given to all movements, and the time required to complete that sequence is the cycle length. A phase represents the part of the cycle assigned to a traffic movement, or to several non-conflictual traffic movements simultaneously. The part of the cycle assigned to each phase is the split, and the portion of the cycle during which the lights do not change is an interval. An attacker that would have the capability to alter the configuration of the controllers, i.e. the timing plan, could significantly hamper the flow of traffic.

Traffic signals can operate either as isolated nodes or as part of a coordinated system. While working in coordination with other signalized intersections, the time (or offset) between the beginning of the cycles of each successive signalized intersection is computed so that vehicles do not stop at intermediary intersections. Isolated traffic signals are not coordinated and do not consider how neighboring intersections are configured.

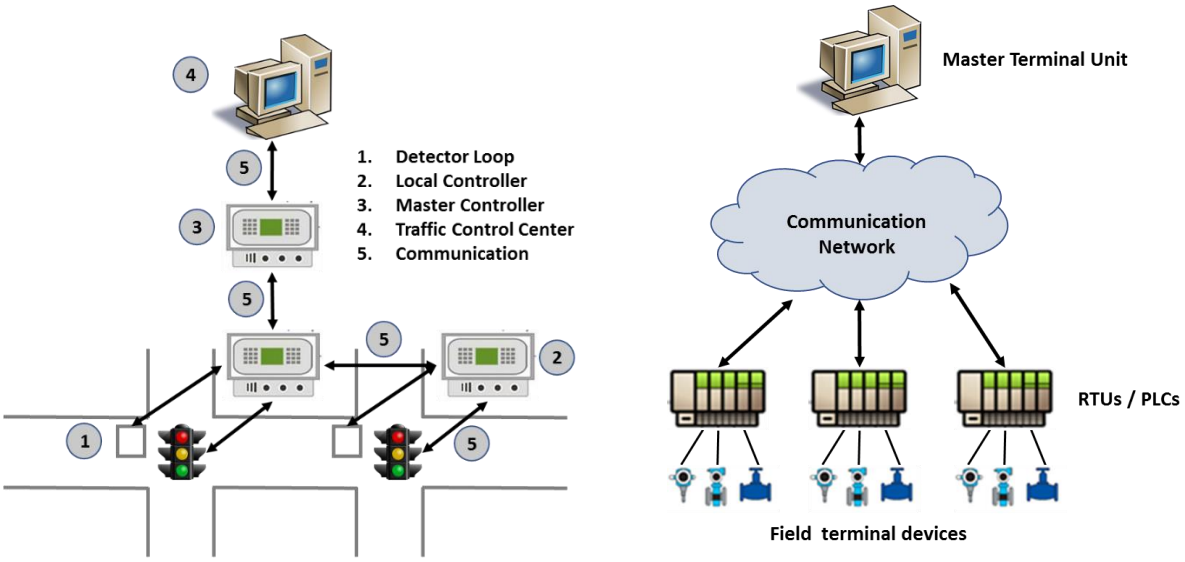
Traffic regulation at isolated intersections can be done using *pre-timed control*, *actuated control* or a combination of both. Pre-timed traffic lights use pre-elaborated timing plans in which the number, sequence and duration of phases are fixed. Pre-elaborated plans are calculated using historic traffic conditions at intersections. Actuated traffic lights use traffic condition information, collected by sensors, to activate phases if the presence of vehicles or pedestrians is detected.

Figure 1a shows the typical hardware components and architecture of a traffic signal control system. It comprises detectors, local controllers, on-street master controllers, a TMC and

communication networks. Detectors are used to determine vehicle presence or pulse duration, needed to calculate vehicle volume, occupancy, speed, etc. Local controllers are responsible for switching head lights at intersections using stored timing plans and schedules previously provided by operators. They receive traffic data from detectors, pre-process it into volume and occupancy parameters, and send it to on-street master controllers. Master controllers are located at intersections and are connected to all local controllers belonging to the same control area to facilitate communication between them and the TMC. They are responsible for selecting traffic responsive timing plans, processing and storing the data collected by the detectors, and monitoring the equipment status at intersections. They communicate with the TMC in the case of critical alarms, on a regular predetermined basis, or when requested by operators. The main function of the TMC is to gather and display information about traffic conditions and intersections equipment status. In addition, it calculates timing plans and the schedules for their selection. Once the timing plans and the selection schedules are generated, they can be downloaded to the on-street master controllers. Furthermore, operators at the TMC can issue commands to master controllers, for example to set the time, or upload information saved in the master controllers.

The above described system has the same distributed architecture, control and monitoring elements as a SCADA network. Basically, a SCADA network controlling an industrial process (depicted in Figure 1b) comprises a central station, or Master Terminal Unit (MTU) at the highest control level. It processes the data collected from the field devices, saves it and displays it in the Human-Machine-Interface (HMI) such that the operators can monitor and control the process. MTU are connected to Remote Terminal Units (RTU) or Programmable Logic Controller (PLC). Both RTU and PLC are data acquisition and control devices that are connected to the measurement and control points in the field. They collect the measurement data, convert it and send it to the

MTU. Additionally, they process the commands sent from the MTU to the field devices. Finally, the communication network provides the connectivity and the data exchange in the network.



(a) (b)

Figure 1: a) Elements of a traffic signal control system (adapted from [8]**) and b) SCADA network components and architecture**

3. RELATED WORK

3.1. Computer security vulnerabilities of process control and traffic control systems

To demonstrate how vulnerable control systems are to cyber attacks, Luallen asked a group of cyber security students to study a process control system governing a physical process in order to find its known vulnerabilities and exploit them [10]. To accomplish this task, students used the Internet to search for information related to the computer security flaws of the chosen system. Next, they hacked it using a commercial cybersecurity training kit. This work demonstrated that nowadays attackers do not require vast knowledge in computer security or any specific skill set to conduct successful attacks against such cyber-physical systems, but also shows that they can readily find on the Internet most of the information they need about their targets. As Luallen's students did, anyone can use Open Source Intelligence (OSINT) tools, and consult existing ICS-

CERT[1] reports, vendor websites and control systems user forums to gather enough information about the target system. Valuable information, such as the system components and architecture, the communication protocols it uses and possible exploits, can be found in this way. Moreover, there are commercial products that can be used to exploit vulnerabilities in the different existing systems.

In the field of computer security of traffic control systems, Cerrudo [5] and Ghena *et al.* [6] detected several security flaws in currently deployed systems in the United States. Even though they studied different systems, they got similar findings which are 1) lack of authentication or poor authentication mechanisms to access traffic light controllers, 2) lack of data encryption, 3) usage of default credentials supplied by the vendors to access traffic light controllers and communication network devices, such as switches, access points or repeaters, and 4) some of the authentication credentials were published in the vendor's website and were not modifiable. In both cases, Cerrudo and Ghena's team could successfully gain access to some of the system components and alter the traffic light state on command.

In their talk, Hack Like a Movie Star [11], Krotofil *et al.* explained that a successful attack to a cyber-physical system requires the execution of five fundamental steps: 1) gain access to the system, 2) discover the system, 3) take control of the system, 4) cause damage or disruption in the physical process, and 5) clean-up all the evidences pointing to a cyber attack.

To illustrate how this works, they created an experimental cyber-physical test bed that reproduced a system controlling traffic lights in a four-way intersection. They got a commercial control system and a cybersecurity training kit, and used them to assemble the test bed. They could easily obtain access to the system, using the credentials provided by the vendor. Once the system

[1] The ICS-CERT is a Computer Emergency Response Team (CERT) created by the US Department of Homeland Security (DHS) specifically to address cyber security issues in Industrial Control Systems.

was accessed, they could learn the system configuration and the system behavior by analyzing the information found in the different system tools available on the machine for diagnosis, development and visualization. Additionally, they applied reverse engineering to some captured binary files and communication messages to deduce the link between the information in the monitoring system and its corresponding elements in the physical process. Then, they successfully manipulated traffic lights state and operation. Finally, for the attack to remain undetectable, they manipulated the system data so that the operator could not notice the changes in traffic conditions during the attacks. Even though they succeeded in hacking the system, authors emphasized that an attacker must have enough knowledge about how the targeted physical system works to identify the attack that better fulfils the attacker's objectives.

These previous works aimed at demonstrating that existing security flaws in currently deployed traffic signal systems could be exploited by adversaries for successfully hacking traffic signals. However, none of those works measured the impacts of the attacks on traffic congestion or traffic safety.

3.2. Usage of experimental scenarios to assess security risks in cyber-physical systems

Experimental setups based on co-simulation frameworks have been used to assess computer security in different cyber-physical systems. In the work of Huang *et al*. [12], it was used to evaluate the impact of computer-based attacks in a process control system governing a chemical reactor. The main objective of this work was to measure the impact of the attacks in the physical process being controlled. Thus, while conducting different types of attacks, the system reaction was modelled and monitored so that the researchers could determine the attack vectors that impacted the physical process the most. They found that, in the steady-state condition of the system, attacks like denial-of-service (DoS) had minor impact on the physical process whereas the

combination of DoS attacks with integrity attacks could lead to important damages to the physical system. Furthermore, the authors determined that the operating costs of the system varied depending on the controllers and sensors targeted during the attacks. Krotofil and Larsen [13] developed an open-source framework to control a chemical plant, based on two realistic models: the Tennessee Eastmann (TE) and the Vinyl Acetate Monomer (VAC) chemical plant models. They redesigned prior Matlab models to produce Simulink models of both plants. First, they used the framework alone to conduct cyber attacks targeting sensors and actuators in the physical process. Then, they coupled it to an industrial control network and conducted cyber attacks aimed to capture and modify the data exchanged between the cyber system and the physical system. Another co-simulation framework was used in [14] to evaluate the impacts of cyber attacks on the monitoring elements of a control process system governing a water supply system. This time, Bernieri *et al*. used the online Fault detection Approach for Critical Infrastructures (FACIES) [15], which is based on a fault diagnosis and intrusion detection architecture, and conducted integrity and availability attacks to evaluate the performance of the fault diagnosis system in effectively detecting them. The tests demonstrated that the fault diagnosis system performed well in detecting replay attacks and attacks targeting actuators state. However, it performed poorly in identifying flooding attacks and attacks targeting sensor information. Results also pointed out that a mediocre performance of the fault diagnosis system, in detecting and identifying the attacks, could induce operators to make unnecessary or erroneous decisions, which could have negative impacts on the physical process. Finally, Lemay, Fernandez and Knight [16] used co-simulation to develop a test bed to evaluate the effects of attacks in both cyber and physical components of an Industrial Control System (ICS) network governing an electric power grid. They used the proven virtualized cluster approach that emulates an IT network with high fidelity described by Calvet *et al*. in [17], and interfaced it with an electrical power flow simulator to reproduce an ICS network controlling an electrical grid. This

test bed has proven to be suitable to reproduce network attacks, such as DoS, data falsification (or injection) and malware infection. Moreover, it was efficient to evaluate the impact of the attacks in both the control network and the power grid.

As we can see, test beds based on co-simulation frameworks have been widely used to conduct experimentation in computer security of cyber-physical systems in different domains. However, to the best of our knowledge, none has been built to assess computer security in road traffic control systems.

3.3. Threat assessment of traffic control system components

A different approach was adopted by Ernst and Michaels in [18]. They presented a threat assessment framework to evaluate the impact of cyber vulnerabilities providing access to field elements of a traffic control system. More specifically, they distinguished the following four access levels whose security flaws can be exploited by an attacker: 1) vehicle detector, 2) corridor synchronization, 3) traditional Internet, and 4) physical access.

They used the Simulation of Urban Mobility package (SUMO) [19] to simulate a road network consisting of a corridor with six signalized intersection which were either coordinated or isolated, depending on the attack type. Then, they conducted simulated tests to reproduce attacks at access levels 1, 2 and 3. Accordingly, they measured possible effects of the tests considering different scenarios of traffic demand.

In this case, the traffic simulation was used to reproduce cyber attacks targeting traffic control system elements, and measure the impacts of those attacks on road congestion. However, this simulation-only approach does not include the cyber component of the traffic signals control system. As such, the simulation must rely on broad assumptions of the impact of cyber attacks, and cannot be used to test network defenses.

4. FUNCTIONAL REQUIREMENTS OF THE TEST BED

Our goal is to design an experimental setup that allows computer security researchers to reproduce cyber attacks targeting traffic control systems and evaluate the impact of those attacks on road traffic in real time. For that, we decided to develop a cyber-physical scenario based on a co-simulation framework to reproduce a two-level distributed control system controlling an urban road network.

One way to accomplish that goal is coupling a monitoring and control system (e.g. a SCADA system) with a microscopic road traffic simulation. On the one hand, the SCADA system provides the required functions to monitor and control in real time large-scale physical processes, such as road networks. On the other hand, traffic simulation is commonly used to reproduce road networks and traffic conditions to plan road traffic control strategies. Additionally, microscopic traffic simulation provides the required information about the different existing entities in road networks, such as pedestrians, vehicles, public transport and traffic lights at a suitable level of granularity.

Moreover, traffic simulation must provide the adequate outputs to measure the economic, environmental and social effects of road congestion resulting from cyber attacking road networks. Some examples of outputs that can be used to measure the impacts are: fuel consumption, greenhouse gases emissions, pollutant emissions, noise emissions, vehicle density, vehicle travel time, and vehicle waiting time. All that information can be provided by the microscopic traffic simulation.

Finally, it is necessary to incorporate a mechanism to properly couple both the cyber and the physical components of the system. This mechanism will allow us to handle the time difference between the supervisory and control system sampling time, and the traffic simulation step time (if any). Additionally, it will permit the data exchange between the control system and the road traffic simulation.

5. TEST BED ARCHITECTURE

With the aim to support the computer security research community with an available, reusable and adaptable platform to test new ideas, we combined different open source software applications to construct our test bed. Figure 2 illustrates this architecture. We describe its details in the following sections.

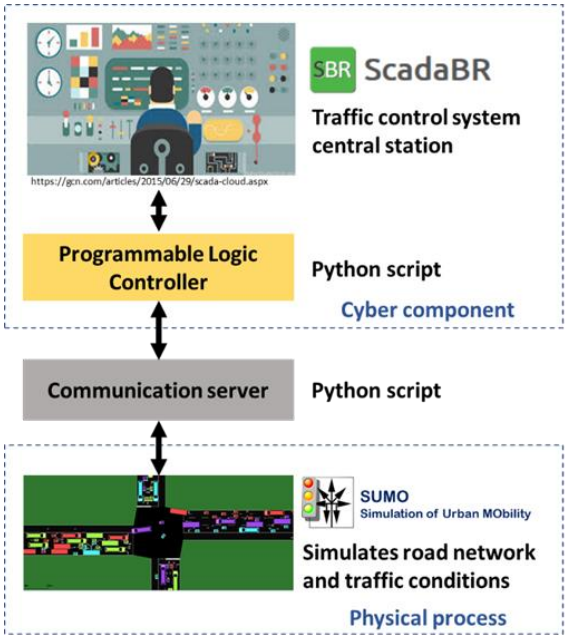


Figure 2: General architecture and components of the proposed test bed

5.1. Monitoring and control system

The high-level control component of the system has been reproduced by using the open source SCADA software ScadaBR 1.0 CE [20]. It is a browser-based SCADA system that 1) provides the monitoring and control functions of a MTU, 2) displays and saves the information about traffic conditions and traffic light states received from the low-level control, 3) enables operators to send commands to change traffic light operation modes (NORMAL/DISABLE), and 4) runs a Modbus client to communicate with each control and data acquisition device in the low-

level control. As such, ScadaBr can be configured to accomplish the functions of a TMC to monitor and control several traffic lights.

The low-level control component was implemented using Python scripts that emulate the functions of the PLCs. They read both the MTU commands and the road network data, convert this information and transmit it to the required level. Moreover, they execute the logic to control the traffic signals in the network which means that they act as traffic light controllers. Each PLC is designed to control all traffic signals at one intersection, and it is possible to replicate as many PLCs as there are signalized intersections in the network. Every PLC script runs a Modbus/TCP server to communicate upstream with ScadaBR, using the Modbus/TCP server functionality available in the Modbus TK Python library [21]. In addition, each PLC runs a TCP client to communicate downstream with the road traffic simulation, through a communication server.

5.2. Road traffic simulation

In our approach, the physical process to be controlled is the road traffic. To reproduce it, we adopted the open-source microscopic traffic simulation package Simulation of Urban Mobility (SUMO), developed by the German Aerospace Center [19]. SUMO offers the flexibility of creating large-scale road networks from common formats, such as shapefiles or Open Street Map files. Road networks in SUMO include the identification of each signalized intersection and traffic light plans. Additionally, Origin/Destination matrices (OD-matrix) can be converted to single vehicle trips to be loaded in the SUMO simulation.

At each time step SUMO generates outputs giving information about all the simulation elements in the network, such as vehicles, intersections, roads, lanes, traffic lights and inductive loops, among others. This level of granularity is necessary to generate the data that will be measured by the monitoring component. Also, it generates noise emission, pollutant emission and

fuel consumption outputs required to quantify the economic, environmental and social effects of road congestion.

SUMO has a Python Traffic Control Interface (TraCI) to interface it with an external application via a TCP socket connection. It permits SUMO to connect to other systems, such as the monitoring and control system. In addition, the TraCI interface allows users to set and modify simulation conditions at any time. For instance, users can change vehicle speeds, driver behavior, road priority or traffic light state as well as force vehicles to change lanes. This is used to enforce state changes dictated by the control component.

SUMO performs a time-discrete simulation, with adjustable step duration from 1 ms and upwards. It also offers two alternatives for the simulation: 1) without visualization, and 2) with visualization through a graphical interface.

5.3. Communication server

To properly couple the monitoring and control system with the physical process, we developed a Python TCP multithreaded communication server. Multithreading enables the server to handle and serve multiple concurrent incoming client requests at the same time. Moreover, it allows us to solve any communication synchronization problem related to the difference in time between the PLC sampling interval and SUMO's simulation time step.

At every simulation step, the server receives data and requests from SUMO and the PLCs. On one hand, the data received from SUMO contains the identification of the signalized intersections and the traffic lights states gathered from the simulation. It is stored in a table that matches each signalized intersection with its controlling PLC. Then, the data is sent to the corresponding PLC when requested. On the other hand, the data received from the PLCs contains the identification of the signalized intersection and the traffic light state to be set during the

simulation. This data is stored in another table that matches each PLC with its corresponding controlled signalized intersection. Then, it is sent to SUMO when requested.

Using the SUMO TraCI interface, we created a script running a TCP client that at each simulation step transmits the simulation results to the server and requests from it new commands from the PLCs. Then, SUMO adjusts the state of the traffic lights according to the information received from the server.

6. VALIDATION AND EXPERIMENTAL SETUP

6.1. Initial validation

For configuration and testing purposes, we built a preliminary setup in which we connected all the components of our proposed co-simulation framework. Then, we used it to validate: 1) the proper integration of all the components, 2) the proper system operation, and 3) the correct conversion/transmission of the data from the MTU to the traffic simulation, and vice versa.

For the first simulation scenario, the road network contained three signalized intersections, spaced 100 m each and running in pre-timed or semi-actuated mode. To control the traffic lights, we reproduced the control logic described by Krotofil *et al*. [11], which is based on a finite-state machine that uses eight possible states and nine transition conditions to commute the traffic lights. It also uses four control signals: AUTO, DISABLE, MAIN ROAD and SIDE ROAD. These signals are used to set the traffic lights operation mode from the MTU. When the operation mode is set to AUTO, the traffic lights commute automatically following the finite state machine programming. In this condition, the traffic lights operate in a pre-timed control mode with fixed control parameters. Moreover, timing plans can be changed by modifying the timing conditions and the state sequence in the finite state machine programming. If the operation mode is set to DISABLE, the lights are set to yellow for all directions at the intersection, and will remain in this condition

until the DISABLE signal is no longer set. When either the MAIN ROAD or the SIDE ROAD signal is set, the traffic lights operate in a semi-actuated control mode. It means that the green light is assigned to the corresponding road (MAIN or SIDE) until vehicles in the opposite road are detected.

All the system components were installed and configured in a desktop computer running the Windows 10 operating system. SUMO, the TraCI simulation update script and the communication server were running directly in the computer. ScadaBr and the PLCs were running in virtual machines. More specifically, ScadaBR and PLC 1 ran in WindowsXP virtual machines, and PLC 2 and PLC 3 ran in Ubuntu Linux virtual machines. All virtual machines were created using VMWare Workstation software. The interconnection of all these elements is shown in Figure 3.

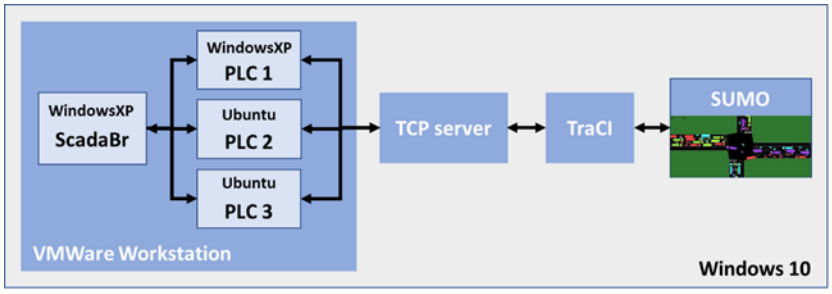


Figure 3: System used in the validation

After validating the integration and proper operation of our preliminary setup, we executed computer-based attacks to evaluate the veracity of the proposed test bed. For that purpose, we configured a Kali-Linux virtual machine connected to the same network to which the PLCs and ScadaBr were connected. Using this Kali-Linux machine as the attacker's machine, we conducted man-in-the-middle (MITM) packet capture attacks and packet injection attacks. Our scenario assumed that an attacker had gained access to the communication network, and intercepted the data exchanged between de MTU and the controller. Since the Modbus communication protocol does not use any authentication nor encryption mechanism, attackers can inject control packets on the

network that will be accepted by the traffic controller. Furthermore, using information available on the Internet, it is easy to reproduce the content of Modbus messages to generate arbitrary control messages and send them to the controller.

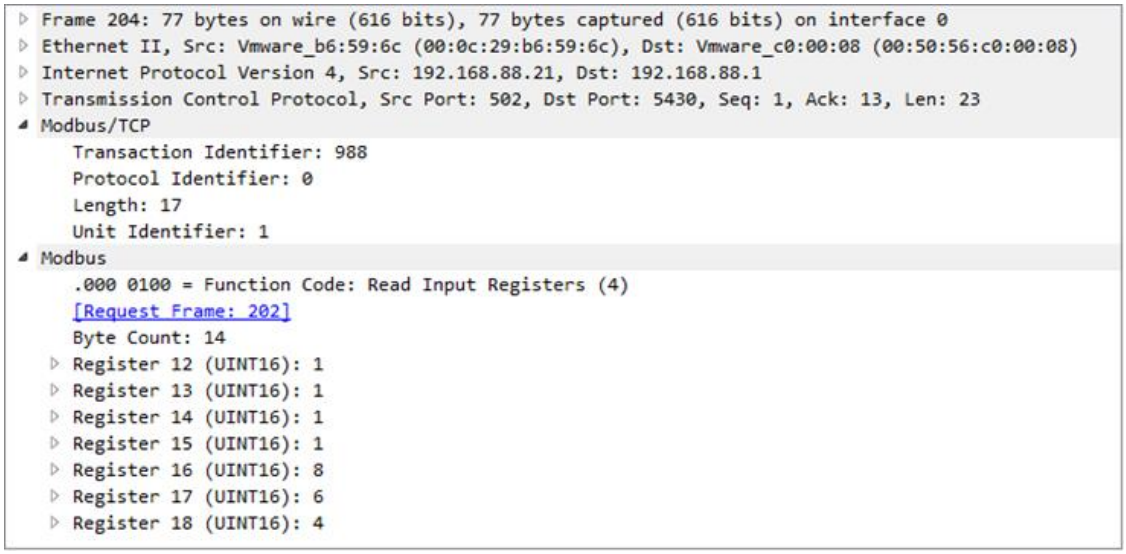
The MITM packet capture attacks were conducted using a Python script which performed an address resolution protocol (ARP) cache poisoning that targeted ScadaBr and PLC 1. This attack let the adversary impersonate both machines and intercept the messages exchanged by them. Figures 4a and 4b show a request and response generated by ScadaBr and PLC 1 before the ARP cache poisoning. Figure 4c shows a request generated by ScadaBr and intercepted by the attacker impersonating PLC 1. Figure 4d shows the corresponding response generated by PLC 1 and intercepted by the attacker impersonating ScadaBr.

For the packet injection attacks, we used another Python script to send Modbus commands from the attacker's machine to PLC 1. Figure 5a shows one request generated by the attacker to set to DISABLE the operation mode of the traffic light (function code *Write Single Coil* and database point reference number 3). Figure 5b shows the response generated by PLC 1 confirming the setting of the database point value.

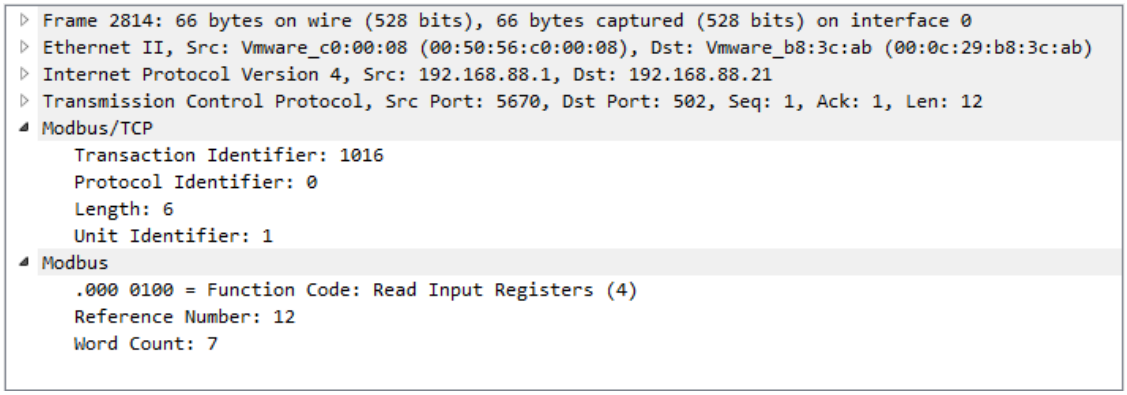
(a)

```
▷ Frame 202: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_b6:59:6c (00:0c:29:b6:59:6c)
▷ Internet Protocol Version 4, Src: 192.168.88.1, Dst: 192.168.88.21
▷ Transmission Control Protocol, Src Port: 5430, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
◢ Modbus/TCP
    Transaction Identifier: 988
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 1
◢ Modbus
    .000 0100 = Function Code: Read Input Registers (4)
    Reference Number: 12
    Word Count: 7
```

(b)



(c)



(d)

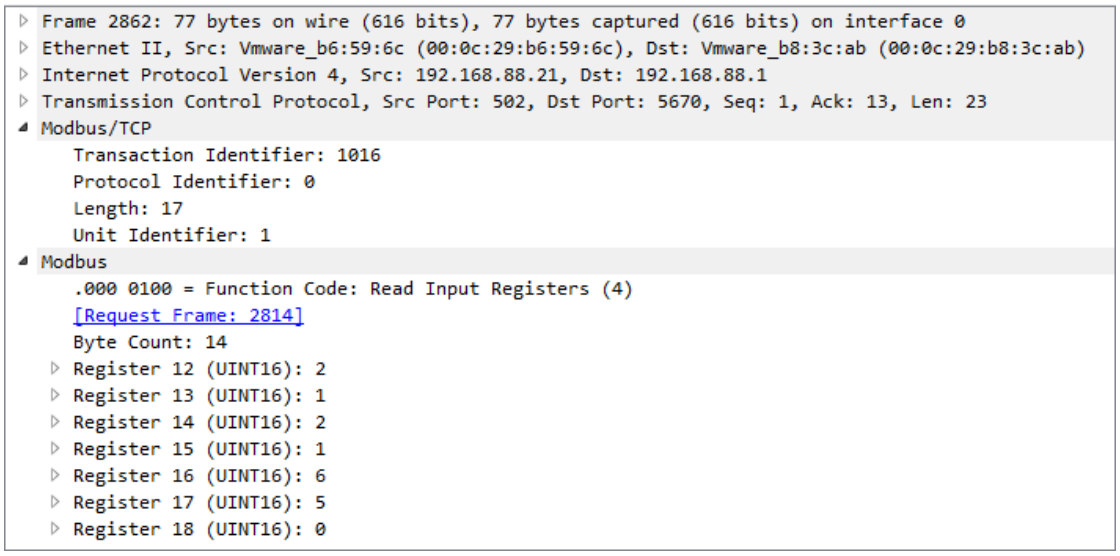


Figure 4: (a) Request sent by ScadaBr to PLC 1 in normal conditions; (b) response from PLC 1 to ScadaBr in normal conditions; (c) request sent by ScadaBr and intercepted by the attacker (mac address 00:0c:29:b8:3c:ab) impersonating PLC 1 during the MITM attack; (d) response sent by PLC 1 and intercepted by the attacker impersonating ScadaBr during the MITM attack

(a)

```
Frame 946: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
Ethernet II, Src: Vmware_b8:3c:ab (00:0c:29:b8:3c:ab), Dst: Vmware_b6:59:6c (00:0c:29:b6:59:6c)
Internet Protocol Version 4, Src: 192.168.88.20, Dst: 192.168.88.21
Transmission Control Protocol, Src Port: 55178, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 1
Modbus
    .000 0101 = Function Code: Write Single Coil (5)
    Reference Number: 3
    Data: ff00
    Padding: 0x00
```

(b)

```
Frame 947: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
Ethernet II, Src: Vmware_b6:59:6c (00:0c:29:b6:59:6c), Dst: Vmware_b8:3c:ab (00:0c:29:b8:3c:ab)
Internet Protocol Version 4, Src: 192.168.88.21, Dst: 192.168.88.20
Transmission Control Protocol, Src Port: 502, Dst Port: 55178, Seq: 1, Ack: 13, Len: 12
Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 1
Modbus
    .000 0101 = Function Code: Write Single Coil (5)
    [Request Frame: 946]
    Reference Number: 3
    Data: ff00
    Padding: 0x00
```

Figure 5: Messages exchanged during the packet injection attack: (a) Request sent by the attacker to PLC 1 to set to DISABLE the operation mode of the traffic light; (b) response sent by PLC 1 confirming the setting

6.2. Experimental setup

Going further, we decided to reproduce a cyber attack targeting a coordinated traffic light system. To do that, we recreated the same road corridor used by Ernst and Michaels [18] (Figure 6). It is composed of six coordinated signalized intersections, spaced 100 m each. An additional intersection was placed at 2,000 m from the east entry of the corridor to generate vehicle platoons. As in Ernst and Michaels' network, no turns are allowed and there is only one lane in each direction on each road in order to keep the model simple. Nonetheless, it is complex enough to demonstrate the impacts of the attacks on a corridor of signalized intersections. The corridor was coordinated to favor the eastbound flow using the simulation parameters shown in Table 1. The traffic lights operation was configured with the timing plan parameters shown in Table 2.

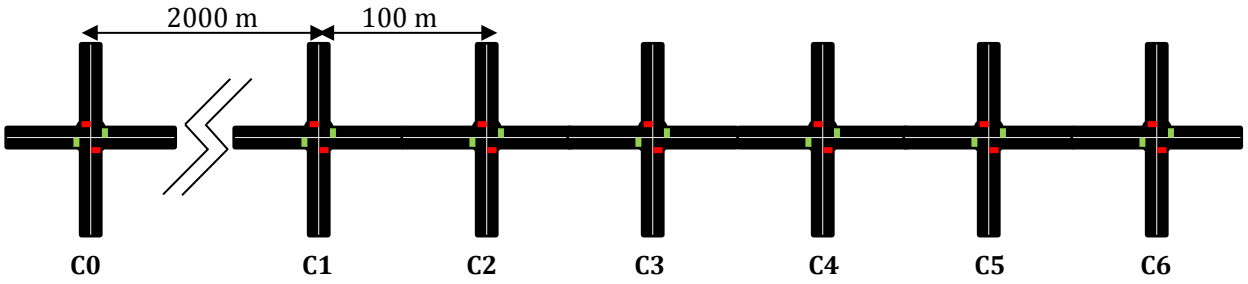


Figure 6: Road network used in the experimental setup

Table 1: Traffic simulation parameters for the different flows

Parameter	Eastbound flow	Westbound flow	Southbound flow	Northbound flow
Max Speed	16.67 m/s	16.67 m/s	11.11 m/s	11.11 m/s
Acceleration	4.5 m/s^2	4.5 m/s^2	4.5 m/s^2	4.5 m/s^2
Deceleration	0.8 m/s^2	0.8 m/s^2	0.8 m/s^2	0.8 m/s^2
Length	5 m	5 m	5 m	5 m
Min Gap	2.5 m	2.5 m	2.5 m	2.5 m
Sigma	0.5	0.5	0.5	0.5
Demand	1000 veh/h	500 veh/h	200 veh/h	200 veh/h
Car following model	Krauss	Krauss	Krauss	Krauss

Table 2: Timing plan parameters for coordinated corridor

Cycle length	Main road green duration	Side road green duration	Yellow duration	All red duration
98 s	60 s	20 s	6 s	3 s

In order to achieve coordination in the corridor, intersection C1 was chosen as the master intersection of the system, and 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. Then, we configured one PLC to control intersection C1 and another PLC to control intersection C5 which was the target of the attacks. In this experiment, the control logic for the four remaining intersections (C2, C3, C4 and C6) were implemented by using the corresponding functionalities within SUMO rather than a simulated PLC. This decision not to use fine-grained emulation for those intersections was only made in order to limit the computer resources required for this experiment. This is without loss of generality, as nothing, other than computational power, would prevent virtualize them all if required.

After configuring the corridor, we executed packet injection attacks targeting signalized intersection C5. Using a Kali Linux machine and the script we used in the validation setup, we sent Modbus/TCP messages to change the programming of traffic lights at the intersection. More specifically, the main green time was changed to 22 s and the side green time to 10 s.

7. EXPERIMENTAL RESULTS

Attacks impacts were measured in terms of travel time and queue length. First, we recorded each vehicle's travel time for the main corridor in the eastbound direction (going through all intersections). Then, we plotted it as a function of the vehicle number (in the order of their generation at the network entrance). Queue lengths were measured at each simulation step, and reported for each intersection. Travel time results are presented for two simulations, along with the queue lengths over time for four intersections for a given simulation, in Figures 7 and 8.

As we see, travel time increases two to threefold under attack. Queue lengths increase even more: they are almost non-existent under normal conditions (up to two vehicles for most intersections) and increase four to fivefold (up to 11 vehicles). The effect on queue length is larger for intersections in the middle of the corridor, with queue spillback from downstream intersections. The evaluation of the impact of this simple attack demonstrates that our co-simulation approach can be used to evaluate the physical impact of real cyber attacks (no need to rely on assumptions of the effect of cyber attacks on control components as in the work by Ernst and Michaels [18]).

8. CONCLUSION

We have developed an experimental scenario that successfully integrates a microscopic traffic simulation (SUMO) with an emulated SCADA system (ScadaBr) to reproduce a traffic signal control system for a coordinated corridor of signalized intersections. This test bed has proven to be suitable to evaluate the impact of cyber attacks on traffic control systems. The impact can be

measured using traffic performance measures such as travel time and queue length rather than IT performance metrics. For example, in our simple attack scenario, travel time is increased two to threefold and queue length four to fivefold. Moreover, we observed that the physical impact of compromising a single node could be felt at other intersections in the network. Such a result highlights the importance of understanding the larger impacts of cyber attacks targeting road networks. This type of emergent result could have only been observed in a co-simulation framework (even for our simple scenario) like ours.

The need to use such an approach in more complex road networks is important as it paves the way to more precise quantitative evaluation of the social, economic and environmental impacts of cyber attacks. This can then provide guidance to policy makers to prioritize cyber security efforts. For example, the co-simulation could be used to identify the intersections with the highest impact on traffic (if attacked) to prioritize monitoring efforts.

Our next step is to use our framework to evaluate the resilience of existing highly complex road networks to cyber attacks. This includes implementing and evaluating the impact of more advanced cyber attacks targeting the centrally-controlled traffic control systems, where the impact of the attack could not be evaluated by using traffic simulators alone.

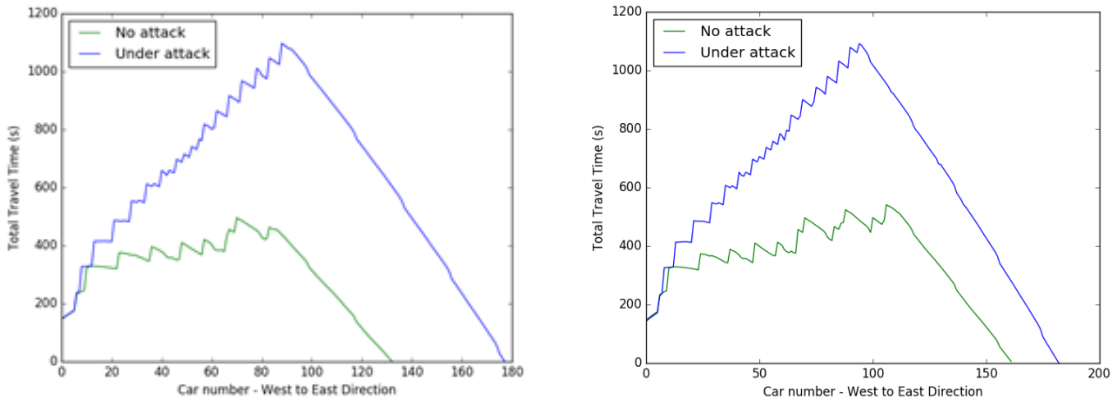


Figure 7: Eastbound vehicle travel time for each vehicle for two simulation runs

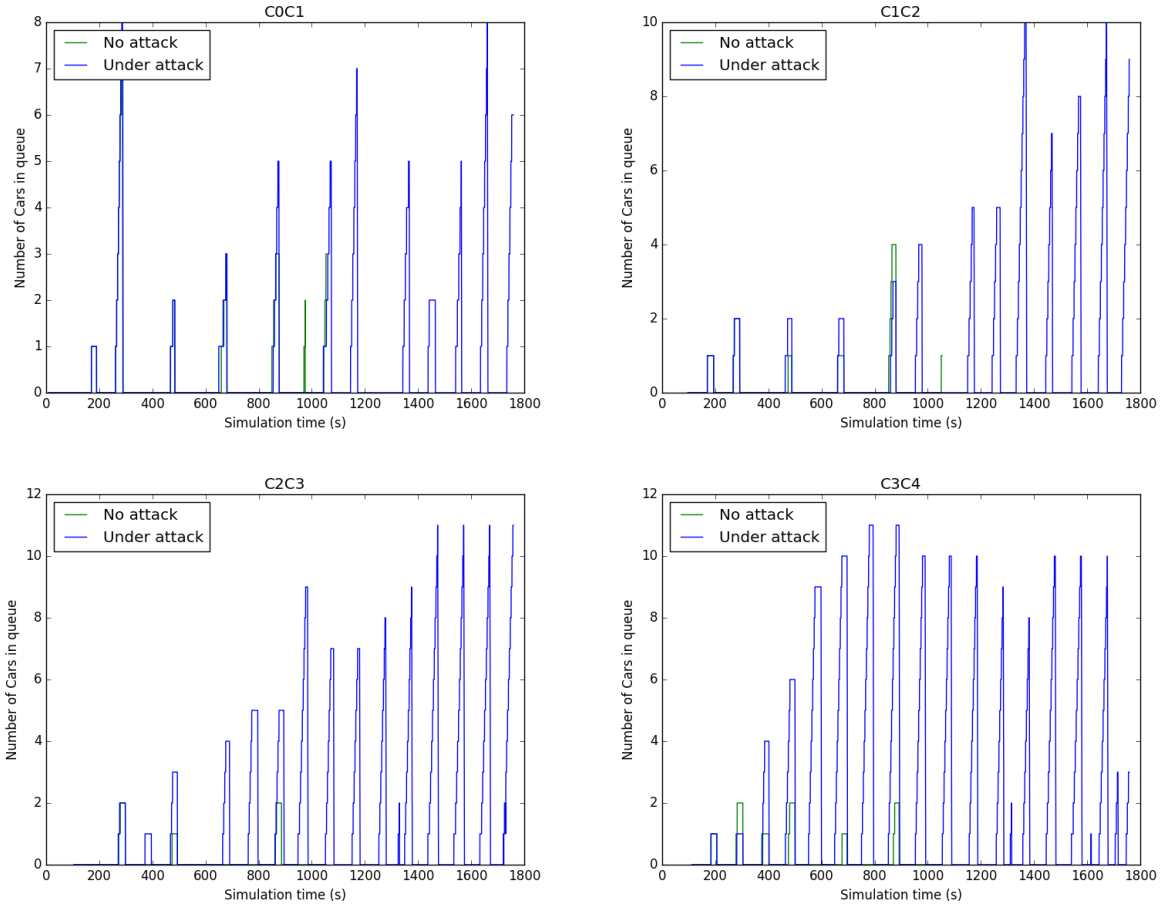


Figure 8: Queue length as a function of time for 4 eastbound approaches of the 6 intersections. Results under normal conditions (no attack) are plotted in green, while results under attack are plotted in blue

9. REFERENCES

[1] INRIX, "INRIX Global Traffic Scorecard," INRIX, 2017.

[2] Ontario Traffic Council, "Advance Traffic Management Systems," in *Ontrario Traffic Manual*, Toronto, Canada, 2007.

[3] M. Tubaishat, Y. Shang and H. Shi, "Adaptive Traffic Light Control with Wireless Sensor Networks," in *Proceedings of IEEE consumer communications and networking conference*, Las Vegas, USA, 2007.

[4] S. Faye, "Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil," Télécom ParisTech, Paris, France, 2012.

[5] C. Cerrudo, "Hacking US (and UK, Australia, France, etc.) Traffic Control Systems," 30 04 2014. [Online]. Available: http://blog.ioactive.com/2014/04/hacking-us-and-uk-australia-france-etc.html. [Accessed 15 04 2017].

[6] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek and J. A. Halderman, "Green Lights Forever: Analyzing the Security of Traffic Infrastructure," in *8th USENIX Workshop on Offensive Technologies*, San Diego, USA, 2014.

[7] Federal Highway Administration, "Traffic control systems handbook," Federal Highway Administration, Washington, USA, 2005.

[8] U.S. Department of Transportation, "Traffic signal timing manual," Federal Highway Administration, McLean, USA, 2008.

[9] Minnesota Department of Transportation, "Traffic Signals 101", Minneapolis, USA: Minessota Department of Transportation, 2015.

[10] M. E. Luallen, "Critical Control System Vulnerabilities Demonstrated - And What to Do About Them," SANS Institute, InfoSec Reading Room, Chicago, USA, 2011.

[11] M. Krotofil and A. D, "Hack Like a Movie Star," 2015. [Online]. Available: http://2015.zeronights.org/assets/files/12-Krotofil.pdf. [Accessed 19 05 2017].

[12] Y.-L. Huang, A. A. Cárdenas, S. Amin, Z.-S. Lin, H.-Y. Tsai and S. Sastry, "Understanding the physical and economic consequences of attacks on control systems," *International Journal of Critical Infrastructure Protection*, vol. 2, no. 3, pp. 73-83, 2009.

[13] M. Krotofil and J. Larsen, "Rocking the pocket book: Hacking chemical plants for competition and extortion," in *DEF CON 23*, Las Vegas, USA, 2015.

[14] G. Bernieri, E. E. Miciolino, F. Pascucci and R. Setola, "Monitoring system reaction in cyber-physical testbed under cyber-attacks," *Computers & Electrical Engineering*, 2017.

[15] C. Heracleous, E. Etchevés M., R. Setola, F. Pascucci, D. Eliades, G. Ellinas and M. Panayiotou, "Critical Infrastructure Online Fault Detection: Application in Water Supply Systems," in *9th International Conference, Critical Information Infrastructures Security (CITRIS 2014)*, Limassol, Cyprus, 2014.

[16] A. Lemay, J. Fernandez and S. Knight, "An isolated virtual cluster for SCADA network security research," in *1st International Symposium for ICS & SCADA Cyber Security Research*, Leicester, UK, 2013.

[17] J. Calvet, C. R. Davis, J. M. Fernandez, W. Guizani, M. Kaczmarek, J.-Y. Marion and P.-L. St-Onge, "Isolated Virtualised Clusters: Testbeds for High-Risk Security Experimentation and Training," in *3rd Workshop on Cyber Security Experimentation and Test (CSET'10)*, Washington, USA, 2010.

[18] J. M. Ernst and A. J. Michaels, "A Framework for Evaluating the Severity of a Traffic Cabinet Cyber Vulnerability," Washington DC, USA, 2017.

[19] D. Krajzewicz, G. Hertkorn, C. Rössel and P. Wagner, "SUMO (Simulation of Urban MObility)-an open-source traffic simulation," in *4th Middle Eastern Symposium on Simulation and Modelling (MESM2002)*, Dubai, 2002.

[20] "ScadaBR," [Online]. Available: http://www.scadabr.com.br/. [Accessed 15 11 2016].

[21] Python, "Package Index > modbus_tk > 0.5.7," Phyton Software Foundation (US), [Online]. Available: https://pypi.python.org/pypi/modbus_tk. [Accessed 19 05 2017].