Secure Traffic Lights: Replay Attack Detection for Model-based Smart Traffic Controllers

Pratham Oza*
Virginia Tech
prathamo@vt.edu

Mahsa Foruhandeh*
Virginia Tech
mfhd@vt.edu

Ryan Gerdes
Virginia Tech
rgerdes@vt.edu

Thidapat Chantem
Virginia Tech
tchantem@vt.edu

ABSTRACT

Rapid urbanization calls for smart traffic management solutions that incorporate sensors, distributed traffic controllers and V2X communication technologies to provide fine-grained traffic control to mitigate congestion. As in many other cyber-physical systems, smart traffic management systems typically lack security measures. This allows numerous opportunities for adversarial entities to craft attacks on the sensor networks, wireless data sharing and/or the distributed traffic controllers. We show that such vulnerabilities can be exploited to disrupt mobility in a large urban area and cause unsafe conditions for drivers and the pedestrians on the roads. Specifically, in this paper, we look into vulnerabilities in model-based traffic controllers and show that, even with state-of-the-art attack detectors in place, false-data injection can be used to hamper mobility. We demonstrate a replay attack by modeling an isolated intersection in VISSIM, a popular traffic simulator and also discuss countermeasures to thwart such attacks.

KEYWORDS

Traffic Controllers, False-data Injection, Model-based CPS.

1 INTRODUCTION

More people are moving into urban areas for better economic and social prospects [26]. This is accompanied by increasing demands on urban transportation infrastructure which is typically at capacity and often cannot cope with this surge in demand. As a result, city-dwellers often face traffic congestion. On an average, commuters

*Both authors contributed equally to this research.

face about 42 hours of delays in a year due to traffic congestion [2]. Such delays cause wasted time, negative impacts to the socioeconomic growth of the area, increased pollution and fuel expenses [9].

Smart cities provide a promising framework to tackle issues pertaining to urban resources [28]. Many traffic control strategies have been proposed that rely on sensors, on-road detectors, cameras and/or other road-side units (RSUs) to measure real-time traffic flow characteristics such as queue lengths, vehicle speeds, traffic density, etc [27]. The measured data is then communicated to a traffic controller that adaptively manages the traffic flow as per the deployed control strategy [23, 30]. However, most of these techniques are deployed on top of the legacy traffic infrastructure [3] which was built without security in mind. This makes the intelligent traffic management system, like many other cyber-physical systems, vulnerable to attacks [11, 15]. Adversarial entities can exploit the weaknesses of these systems using, for example, replay attacks (i.e., provide controllers with spurious sensing information [6]) or disrupt measurements and communications (i.e., denial-of-service (DoS) attacks [16]) to cause phantom traffic jams or otherwise degrade the quality of (transportation) service [18].

Model-based traffic control approaches have been fairly popular in estimating traffic queues and controlling the traffic flow. Traditional approaches such as UTOPIA [19], PRODYN [13], and RHODES [20], etc., as well as newly developed MPC-based traffic control algorithms [17] rely on traffic models and state-space estimations to predict traffic flow and hence establish a finer control over the traffic. Such model-based controllers not only provide real-time traffic control from received information about vehicle density and speed but also provide self-adaptive control strategy that combines current measurements as well as future estimations of the traffic characteristics [7, 30]. Since such systems still rely on measurements from insecure sensors, they need to accompanied with attack detectors that monitor the estimates made by the model. These detectors can detect malicious activities if the decisions do not corroborate the observations [25]. Our key contributions are:

- We propose a replay attack, which is a subset of false-data injection attacks, on the inductive loop detector used in transportation systems to estimate vehicle flow.
- We explore the shortcomings of the existing model-based attack detectors.
- We provide a threshold-based countermeasure and validate its performance with VISSIM, a real-world traffic simulator.

In this work, we discuss the system model as a four-way traffic intersection and provide a state-space model in Sec. 2. Then, in

Sec. 3 we design a replay attack on the inductive loop detectors. We demonstrate the shortcoming of model-based detectors in detecting our attack in Sec. 4.1. We propose a threshold-based approach as a countermeasure in Sec. 4.2 and then conclude in Sec. 5.

2 SYSTEM MODEL

This section delves into the traffic parameters and state-space models to mimic a four-way intersection traffic controller.

2.1 Traffic, Intersection and Infrastructure

Most traffic controllers in the U.S. deploy inductive loop detectors to collect traffic-related information [12]. Occupancy sensor readings obtained from these inductive loop detectors are then used to establish control over the traffic lights [1]. In this paper, we consider an isolated four-way intersection as shown in Fig. 1a. Traffic lights are situated at the end of all four lanes downstream and are controlled using our model-based traffic controller. This traffic controller acquires real-time measurements from inductive loop detectors located upstream. For simplicity, we consider that the vehicles use the intersection only to go straight (no left/right turns). Adding turning lanes and vehicles simply requires extending the system model with multiple traffic lights. Incoming traffic within each lane has the following characteristics:

- Intensity (I_t)[v/h]: Average rate of vehicles entering a given lane at a time instant t.
- Saturation Flow (S)[v/h]: The maximum number of vehicles per hour that can flow through the intersection depending on the lane capacity.
- Green Light Duration (z)[s]: The time for which the vehicles from a lane can access the intersection.

The traffic controller decides the green-yellow-red timing schedule for the traffic lights located at the intersection. The controller follows a *cycle time (T_c[s])* that defines the time window within which each traffic light turns green at least once. At the beginning of each T_c, the controller decides the schedule for the next cycle window as per the control algorithm and the process repeats at the end of the cycle.

A vehicle entering the lane activates the inductive loop installed on the road that generates a signal variation whose duration depends on how long a vehicle stays over the detector. The amount of time the detector remains activated within a given sample period determines the *occupancy rate (O[%])* of the sensor. When a vehicle passes over the detector at high speed, the occupancy rate reported would be a lower value since the vehicles spend less time over the detector. This value increases as the speed of the passing vehicle decreases, and eventually saturates at 100% when the vehicle is stationary on top of the detector.

While multiple loop detectors may be located on the road to measure different traffic characteristics, in practice, only the one located farthest from the intersection is used to measure the flow of the traffic as it has the least probability of saturation [8]. Therefore, our traffic model considers the loop detectors situated at the farthest end of our incoming lanes, as shown in Fig. 1a for traffic estimation. Our model is communication-agnostic and the measurements may be reported using Dedicated Short Range Communication (DSRC), CV2X, etc., while assuming that the communication channel is secure. We now provide a state-space model that closely represents our adaptive traffic control system.

2.2 Model-based Self-Adaptive Traffic Controller

Among the various application-specific models that have been developed to analyze traffic characteristics, the model presented in [4] provides a state-space model to estimate the parameters of the traffic model based off inductive loop measurements [30]. We use the model in [4] to adjust the green light duration at the intersection.

Traffic control within each lanes of an intersection can be characterized by estimating the development of vehicle queues in each lane over time. By utilizing the occupancy rate measurements provided by the sensors (also known as the measurable quantities) the states of the system (X_t) at a given time instant t need to be formulated to describe how the queues evolve with the measurements and the control decisions for the green light. The state-space equations are defined as

$$X_t = [Q_t, O_t]^T, \tag{1}$$

where Q_t is the current estimated queue length in a given lane and O_t is the occupancy of the inductive loop detector at time t.

Similarly, y_t defines the observable quantities in the system that influence the evolution of the queues in all lanes and is given by

$$y_t = [Y_t, O_t]^T, \tag{2}$$

where Y_t is the number of vehicles being dispatched within a given time window of T_c due to the controlled green light.

The queue length within each lane also depends on the saturation flow of the lane. Saturation flow (S [veh/h]), as explained earlier, provides the maximum number of vehicles that can enter into the lane due to the physical capacity of the lanes and intersection. Hence, to relate the queue development with the saturation flow, a parameter δ is introduced. δ is set to zero when the incoming vehicle flow is less than the saturation flow and is set to unity, otherwise. The value of δ is updated at the start of each traffic cycle window and ensures that the incoming flow never exceeds the saturation flow.

The objective of the controller is to collectively minimize the vehicle queues in each incoming lane of our intersection. Eq. 3 describes how queue lengths can be estimated from current queues in the system as well as the current sensor measurements

$$Q_{t+1} = \delta_t Q_t - [\delta_t S + (1 - \delta_t) I_t] z_t + I_t, \tag{3a}$$
$$O_{t+1} = k_t Q_t + \beta_t O_t + \lambda_t, \tag{3b}$$

where λ, k and β are environment-dependent parameters that describe the traffic flow within each lane and have to be estimated real-time as the traffic flow intensity changes. Here, we estimate the values of λ, k and β using our traffic flow simulator, VISSIM [5] by fitting a linear regression model to each arm of the intersection, as shown in Fig. 1b and c.

The observable variables of the system such as the vehicles being dispatched at each green light and the occupancy rate measurements are given by Eq. 4 as

$$Y_{t+1} = \Delta Q_{t+1} + I_t, \tag{4a}$$
$$O_{t+1} = O_{t+1}. \tag{4b}$$
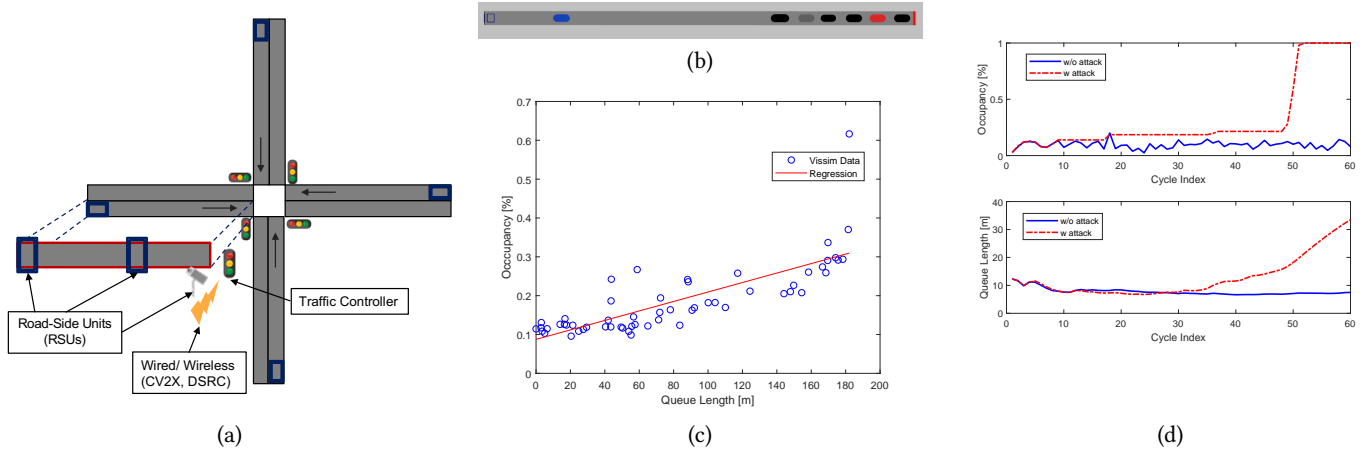
Figure 1: (a) A typical smart traffic management system for an isolated four-way intersection, (b) VISSIM queue simulation, (c) Linear regression model to estimate λ, k and β using queue formation from VISSIM simulation, (d) Observables of the system with/without attack, Occupancy and Queue Length (the value of the scaling parameter f is set to 0.7 in this graph).

And finally, the control law for updating the green lights with changing occupancy rate measurements is

$$z_{t+1} = z_t \ (1 + f \times \Delta O), \qquad (5)$$

where, the green light time, z_{t+1} for the next traffic cycle depends on the current green light time z_t and the change in occupancy rate measurements within the last traffic cycle. With a constant traffic flow, the green light timing settles to a value at which the occupancy rates do not change and the queues developed are constant. The parameter f is a scaling factor determined experimentally.

Since turning vehicles are disregarded in our model, even though there are four incoming flows into the intersection, every flow has a complementing non-conflicting flow. That is, when the traffic lights turn green, vehicles from two lanes can access the intersection at once without causing collisions. Thus, the controller can calculate the green times as per the maximum flow intensity of the two non-conflicting flows. This will ensure that the flow with less intensity will also be served properly if the timings are decided as per the dominant flow [10]. Hence, the traffic lights of the non-conflicting flows can be paired and controlled at once to improve the efficiency of the system. Thus, our traffic controller for a four-way intersection controls the timing schedule for four traffic lights or two traffic light pairs within a cycle window T_c.

The equations described above can be extended for a four-way intersection traffic controller with two traffic light pairs. To ensure the robustness of our model, measurement noise (v) and process noise (p) are added. The complete system model equations are given below. This traffic controller model is then implemented on MAT-LAB with VISSIM simulation-in-loop to verify its performance. A short video of VISSIM simulation with the adaptive traffic controller is available [24].

$$\begin{bmatrix} Q_{1,t+1} \\ Q_{2,t+1} \\ O_{1,t+1} \\ O_{2,t+1} \end{bmatrix} = A \times \begin{bmatrix} Q_{1,t+1} \\ Q_{2,t+1} \\ O_{1,t+1} \\ O_{2,t+1} \end{bmatrix} + B \times \begin{bmatrix} z_{1,t} \\ z_{2,t} \end{bmatrix} + F + p_t, \qquad (6)$$

$$\begin{bmatrix} Y_{1,t+1} \\ Y_{2,t+1} \\ O_{1,t+1} \\ O_{2,t+1} \end{bmatrix} = C \times \begin{bmatrix} Q_{1,t+1} \\ Q_{2,t+1} \\ O_{1,t+1} \\ O_{2,t+1} \end{bmatrix} + H + v_{t+1}, \qquad (7)$$

where

$$A = \begin{bmatrix} \delta_{1,t} & 0 & 0 & 0 \\ 0 & \delta_{2,t} & 0 & 0 \\ k_{1,t} & 0 & \beta_{1,t} & 0 \\ 0 & k_{2,t} & 0 & \beta_{2,t} \end{bmatrix},$$

$$B = \begin{bmatrix} -\delta S_{1,t}/dt - (1-\delta)I_{1,t}/dt & 0 \\ 0 & -\delta S_{2,t}/dt - (1-\delta)I_{2,t}/dt \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$F = \begin{bmatrix} I_{1,t}dt \\ I_{2,t}dt \\ \lambda_1 \\ \lambda_2 \end{bmatrix}, C = \begin{bmatrix} 0 & -\alpha & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = -C \times \begin{bmatrix} Q_{1,t} + I_{1,t}dt \\ Q_{2,t} + I_{2,t}dt \\ 0 \\ 0 \end{bmatrix},$$

where dt shows the simulation step time.

$$\begin{bmatrix} Z_{1,t+1} \\ Z_{2,t+1} \end{bmatrix} = \begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} \times \left(1 + f \times \begin{bmatrix} \Delta O_1 \\ \Delta O_2 \end{bmatrix} \right); Z_{1,t+1} + Z_{2,t+1} <= T_c.$$

$$(8)$$

In Eq. 7, α represents the ratio of left turning vehicles in a given incoming traffic stream. To extend the model for turning lanes, α values can be found through various data collection points and traffic flow estimations. For simplicity, we set α to zero.

3 THREAT MODEL AND ATTACK STRATEGY

In a typical CPS, adversaries can attack either the analog stimulus that excites the sensor, the analog output of the sensors or the digital output of the sensor system that is fed into the control system. This paper focuses on the digital output of the inductive loop detector as a possible foothold for a replay attack.

3.1 Attack Goal and Assumptions

Traffic congestion can have network level effects and a large-scale impact on the safety of everyone involved. The attacker's objective is maximizing the queue length to eventually disrupt the transportation network. Hence, our attacker's goal is to decrease the efficiency of traffic lights by manipulating the traffic controller.

Since we focus on an isolated traffic intersection, this paper assumes that the traffic has not been affected by prior events such as impending queues, traffic congestion etc. Also, the vehicles arrival pattern in each lane is modeled by Poisson distribution [22] where the length of each vehicle is the same. All vehicles are assumed to follow the rules of the traffic light.

The attacker has the ability to view and affect the perceived occupancy rate. Once the occupancy sensors perform the measurements, the attacker is able to record the output of the sensor and sending it later at an opportune moment. The attacker may do so by attaching an additional off-the-shelf microcontroller to read the sensor values and/or an inexpensive transmitter antenna to send out the falsified data, which summarizes our replay attack. Additionally, the attacker has sufficient resources to monitor the system (intersection). They are also able to wait until the proper conditions, from their perspective, are present for the attack to take place. Critically, the attacker does not have the ability to tamper with the sensor data and can only replay the existing data.

3.2 Attack Strategy

Since the control law for the traffic lights relies on the difference in occupancy rate measurements, a negative occupancy rate difference would mean reduction in traffic and would require reduction in the green light time allocation. Hence, the attacker can constantly capture the sensor data by monitoring the system and induce an attack by replaying a low occupancy measurement in place of the actual reported. By judiciously choosing the right data to send, the attacker can constantly introduce negative difference in occupancy rate measurement and disrupt the lights by eventually bringing the green light time to zero. In the state space model, we target the

---

Algorithm 1 Replay attack

1: z: green light duration
2: O: occupancy of the inductive loop detector
3: f: environment dependent parameter
4: ras: replay attack signal
5: **procedure** REPLAY ATTACK (O_t, z_t, f)
6:     ras ← Inf
7:     **for** each t in T_C **do**
8:         **if** $O_t - O_{t-1} <$ ras $- O_{t-1}$ **then**
9:             ras ← O_t ▷ ras: the smallest value of the sensor over time
10:             $z_t \leftarrow z_{t-1}\left(1 + f\left(\text{ras} - O_{t-1}\right)\right)$
11:         **else**
12:             $z_t \leftarrow z_{t-1}\left(1 + f\left(O_t - O_{t-1}\right)\right)$

---

green light control law given at Eq. 4 to craft a replay attack. This attack is performed by changing the green light equation as,

$$z_{t+1} = z_t\left(1 + f\left(\text{ras} - O_t\right)\right), \tag{9}$$

where "ras" is the replay attack signal, which holds the smallest value of O up until the current time instant, and gets updated if $O_{t+1} - O_t <$ ras $- O_t$. This is summarized in Alg. 1. Eventually this drives the green light allocation to be less that sufficient and creates a long queue in at least one arm.

The effect of the attack on a detector placed in one of the lanes in the system is given in Fig. 1d in terms of occupancy and queue length with 700 v/h intensity as the dominant traffic flow when the attack is launched at 420 seconds. It is clear that when the system is under attack, the occupancy difference injected by the attacker makes the green light allocation smaller. Due to this, the queue length increases drastically, the occupancy sensors saturate and thus, disrupt the traffic network.

We next evaluate the performance of our attack in presence of the state-of-the-art attack detectors.

4 COUNTERMEASURES

Algorithm 2 Threshold-based attack detection

1: z: green light duration
2: I: intensity of flow of vehicles into an intersection
3: O: occupancy of strategic loop detector
4: thrsh: offline value defined on the queue length in each arm
5: ζ: scaling factor
6: **procedure** DEFINE AN OFFLINE THRESHOLD (O_t, z_t)
7:     thrsh ← $-\infty$
8:     **for** each t in T_C **do**
9:         **if** $(O_t - O_{t-1}) >$ thrsh **then**
10:             thrsh ← avg$\{$thrsh$, O_t\}$
11: **procedure** COUNTER MEASURE (O_t, z_t, f, I_t)
12:     **for** each dt in T_C **do**
13:         **if** $(O_t - O_{t-1}) <$ thrsh **then**
14:             attack detected
15:             $\zeta \leftarrow \frac{T_c}{\sum_{arm} I}$
16:             $z_t \leftarrow z_{t-1}\left(\zeta \times I_t\right)$
17:         **else**
18:             $z_t \leftarrow z_{t-1}\left(1 + f \times \Delta O\right)$
19:             update offline threshold

In this section, by using an approximate model for estimating the occupancy previously presented in Sec. 2, we show that the state-of-the-art model-based detectors fail to detect our replay attack, and then we provide countermeasures to modify such detectors.

4.1 Model-based attack detectors

In practice, an attack detector is based off an approximate model of the system, as it must closely represent the system without relying on having an exact knowledge of it. The model given in Eq. 10 and Eq. 11 holds the occupancy values as the main observable in the system. The simplicity of the model, and the fact that these occupancy measurements are the main foothold for the attacker, makes the approximate model a better candidate for our analysis hereafter. The state space equations of the approximate model are

$$\begin{bmatrix} O_{1,t+1} \\ O_{2,t+1} \end{bmatrix} = A' \times \begin{bmatrix} O_{1,t} \\ O_{2,t} \end{bmatrix} + B'\left(dt\right) \times \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} + \begin{bmatrix} p_{1,t} \\ p_{2,t} \end{bmatrix}, \tag{10}$$

(a) Noise power: 0.015% of signal power (b) Noise power: 0.172% of signal power (c) Noise power: 0.394% of signal power
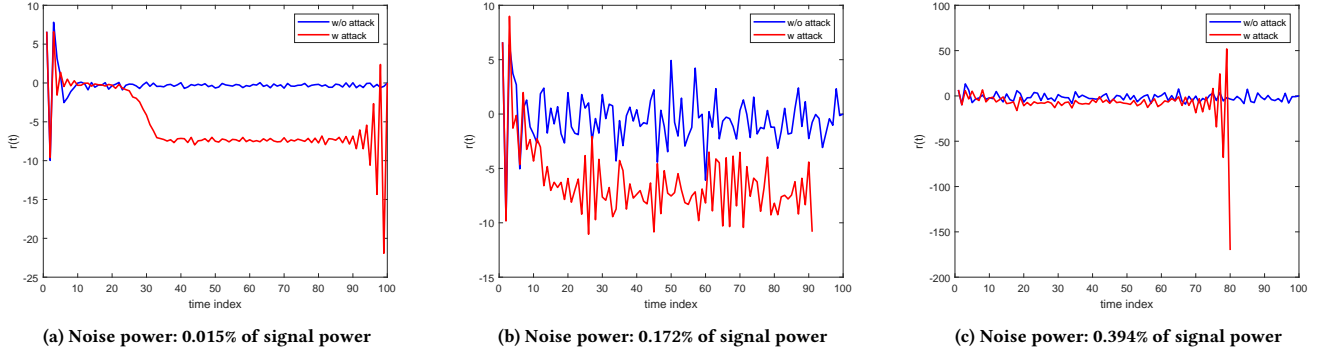
Figure 2: The performance of the state-of-the-art model-based detector [25] under different measurement noise power values.
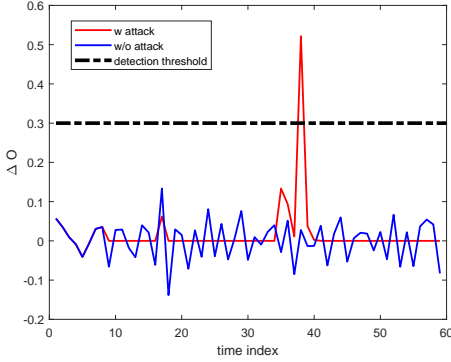

Figure 3: Setting a threshold to detect an attack using the occupancy sensor values.

$$\begin{bmatrix} O_{1,t+1} \\ O_{2,t+1} \end{bmatrix} = C' \times \begin{bmatrix} O_{1,t} \\ O_{2,t} \end{bmatrix} + D'\,(dt) \times \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} + \begin{bmatrix} v_{1,t} \\ v_{2,t} \end{bmatrix}, \qquad (11)$$

where

$$A' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B' = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, C' = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix}, D' = D,$$

and u shows the input approximation as per vehicles entering and leaving and u_{t+1} is given as

$$u_{t+1} = i \times dt - S \times z_{t+1}. \qquad (12)$$

Next, we explore the existing methods for detecting our replay attack on the approximate model. Model-based attack detectors are extensively covered in the state-of-the-art as strong candidates for detection of false-data injection in control systems [25]. Here we demonstrate shortcomings of the model-based approach by Pasqualetti et al. [25] in detecting our replay attack, and then provide our threshold-based countermeasure which successfully detects malicious activities when our attack is in progress. Our results are then validated through VISSIM.

Model-based attack (anomaly) detectors for CPS are those that make an estimate of how the system should be observed to evolve and compare it with the measured value of the state (how the

system actually evolves), shown by $r\,(t)$ at Eq. 13. In absence of an attacker $r\,(t)$ should be close to zero. Pasqualetti et al [25] propose a monitor $\omega\,(t)$ for this purpose which is defined as

$$E\,\dot{\omega}\,(t) = \left(A' + G\,C'\right)\omega\,(t) - G\,y\,(t),$$
$$r\,(t) = C'\,\omega\,(t) - y\,(t), \qquad (13)$$

where $E = 1$ in our approximate model, $\omega\,(0) = x\,(0)$, and G is the output injection matrix which should be selected so that the pair $(E, A' + G\,C')$ is regular and Hurwitz [14].

Fig. 2 illustrates the robustness of the Pasqualetti's detector to the measurement noise power when our attack is taking place. With low measurement and process noise, an abrupt change will appear in the error calculations which makes it possible to perform anomaly based attack detection shown in Fig. 2a. As the measurement noise power increases while the estimation noise is kept consistent, the performance of Pasqualetti's detector degrades (Fig. 2b) and eventually breaks (Fig. 2c). This shows that our replay attack is not detectable by the model-based detector of the state-of-the-art if noise power exceeds 0.394% of the original signal power.

4.2 Threshold-based attack detectors

Having shown the shortcomings of the model-based attack detectors to detect our replay attack, we now introduce a countermeasure in this section. The conventional and most common countermeasure against false-data injection attacks on control systems is Kalman Filter (KF) estimation of the states of the system followed by a simple comparison [4, 21]. Even though KF has high accuracy of estimation over the other available filters, it is designed on the basis of the not necessarily correct assumption that the parameters are Gaussian distributed in a linear system. Other filters such as particle filtering [29] overcome this shortcoming by addressing the general non-Gaussian non-linear systems at the cost of higher complexity. To avoid the complexity-performance trade-off, a general solution to this problem could be using the timestamps of the inductive loop detector output. However, this method depends on another sensor output and there is no guarantee that it is secure. Also, a sophisticated attacker can forge time-stamps by crafting a more complicated attack than a replay attack.

We address these shortcomings by proposing a threshold based countermeasure described in detail in Alg. 2. The threshold is defined in an offline training process to distinguish between benign and malicious activities by looking into the changes in the occupancy sensors readings in two consecutive time instants (given by ΔO). In this case, we adjust the threshold in a way that it detects the abrupt change generated in the reported ΔO when there are ongoing malicious activities. Anytime the change in ΔO goes above this threshold an attack is said to be detected. This threshold changes overtime, and needs to be updated using secure data. Note that upon detection of an attack the green light values get updated by a new control law

$$z_t = z_{t-1}\left(\zeta \times I_t\right), \tag{14}$$

where ζ is a normalizing factor given by $\zeta = \frac{T_c}{\sum_{arm} I}$, such that the green time is proportionally distributed as per the incoming traffic flow rate, while assuming the I_t measurement is secure.

Fig. 3 shows the change of occupancy over time for one arm with/without attack, where the threshold is defined to create separability between the spike in the graph caused by the attack versus the case with no attack.

4.3 Time Complexity

The threshold-based detection mechanism is provided in Alg. 2, where the training of threshold is performed offline and hence does not add to the time complexity of the approach. The detection mechanism however, includes a comparison operation which results in time complexity of $O(n)$.

5 CONCLUSION

In this paper, we first introduce a replay attack on the inductive loop sensors which are used for traffic control in an isolated four-way intersection. More specifically, the occupancy sensor measurements are used as a feedback to adjust the green light timings for the traffic lights. By introducing a replay attack, a type of false-data injection attack, our attacker minimizes the efficiency of the traffic controller by affecting the green light allocation and thereby maximizing the queue lengths in at least one of the lanes of the intersection. By showing that the existing model-based attack detectors fail in detecting our attack, we put forth an alternative threshold-based detection algorithm. This detector observes the occupancy sensor data over time and raises a flag if the changes in the occupancy sensor measurements are above an offline defined threshold. Assessment of the current threshold-based detector under a smarter attacker capable of evasion attacks is the future direction of our work. We are also interested in extending this study to analyse the feasibility of our approach for broader ranges of traffic models that rely on occupancy sensor measurements.

6 ACKNOWLEDGEMENT

REFERENCES

[1] M Burger, M Van Den Berg, Andreas Hegyi, B De Schutter, and J Hellendoorn. 2013. Considerations for model-based traffic control. *Transportation Research Part C: Emerging Technologies* 35 (2013), 1–19.

[2] Graham Cookson and Bob Pishue. 2017. INRIX Global Traffic Scorecard– Appendices. *INRIX research* (2017).

[3] Christopher M Day, Rob Helt, Daniel Sines, and AM Tahsin Emtenan. 2019. Leveraging Sensor-Based Vehicle Position and Speed Information in Traffic Signal Control with Existing Infrastructure. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE.

[4] Jindrich Dunik, Pavla Pecherkova, and Miroslav Flidr. 2006. State space model of traffic system and its estimation using derivative-free methods. *UTIA AVCR* (2006).

[5] Martin Fellendorf and Peter Vortisch. 2010. Microscopic traffic flow simulator VISSIM. In *Fundamentals of traffic simulation*. Springer.

[6] Yiheng Feng, Shihong Huang, Qi Alfred Chen, Henry X Liu, and Z Morley Mao. 2018. Vulnerability of traffic control system under cyber-attacks using falsified data. In *97th Annual Meeting of the Transportation Research Board*.

[7] Alberto Fernández-Isabel and Rubén Fuentes-Fernández. 2015. Analysis of intelligent transportation systems using model-driven simulations. *Sensors* 15, 6 (2015).

[8] FHWA. 2006. Traffic Detector Handbook, Chapter 4. https://www.fhwa.dot.gov/publications/research/operations/ITS/06108/04.cfm. [Online; accessed 24-December-2019].

[9] FHWA. 2018. Intersection Safety, Data and analysis. https://highways.dot.gov/research-programs/safety/intersection-safety. [Online; accessed 18-Jan-2019].

[10] FHWA-USDoT. 2019. Signalized intersections: Informational guide. https://www.fhwa.dot.gov/publications/research/safety/04091/07.cfm.

[11] Branden Ghena, William Beyer, Allen Hillaker, Jonathan Pevarnek, and J Alex Halderman. 2014. Green lights forever: Analyzing the security of traffic infrastructure. In *8th {USENIX} Workshop on Offensive Technologies ({WOOT})*.

[12] Juan Guerrero-Ibáñez, Sherali Zeadally, and Juan Contreras-Castillo. 2018. Sensor technologies for intelligent transportation systems. *Sensors* 18 (2018).

[13] Jean-Jacques Henry, Jean Loup Farges, and J Tuffal. 1984. The PRODYN real time traffic algorithm. In *Control in Transportation Systems*. Elsevier, 305–310.

[14] Adolf Hurwitz. 1898. Ueber die Composition der quadratischen Formen von belibig vielen Variablen. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1898 (1898), 309–316.

[15] Kaveh Bakhsh Kelarestaghi, Mahsa Foruhandeh, Kevin Heaslip, and Ryan Gerdes. 2019. Intelligent transportation system security: impact-oriented risk assessment of in-vehicle networks. *IEEE Intelligent Transp. Systems Magazine* 10 (2019).

[16] Zhiyi Li, Dong Jin, Christopher Hannon, Mohammad Shahidehpour, and Jianhui Wang. 2016. Assessing and mitigating cybersecurity risks of traffic light systems in smart cities. *IET CPS: Theory & Applications* (2016).

[17] Shu Lin, Bart De Schutter, Yugeng Xi, and J Hellendoorn. 2009. A simplified macroscopic urban traffic network model for model-based predictive control. *IFAC Proceedings Volumes* 42 (2009), 286–291.

[18] Yue Liu and Gang-Len Chang. 2011. An arterial signal optimization model for intersections experiencing queue spillback and lane blockage. *Transportation research part C: emerging technologies* 19, 1 (2011), 130–144.

[19] Vito Mauro and C Di Taranto. 1990. Utopia. *IFAC Proceedings Volumes* (1990).

[20] P. Mirchandani and Fei-Yue Wang. 2005. RHODES to intelligent transportation systems. *IEEE Intelligent Systems* (2005), 10–15.

[21] Yilin Mo and Bruno Sinopoli. 2010. False data injection attacks in control systems. In *The 1st workshop on Secure Control Systems*. 1–6.

[22] Alan V Oppenheim. 1999. *Disc-time signal processing*. Pearson.

[23] Pratham Oza and Tam Chantem. 2019. A Real-Time Server Based Approach for Safe and Timely Intersection Crossings. In *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 1–12.

[24] Pratham Oza and Mahsa Foruhandeh. 2020. VISSIM simulation with replay attack. https://youtu.be/CnZGagb-s_A. [Online; accessed January 2, 2020].

[25] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. 2013. Attack detection and identification in cyber-physical systems. *IEEE transactions on automatic control* 58, 11 (2013), 2715–2729.

[26] Statista. 2017. US Urbanization. https://www.statista.com/statistics/269967/urbanization-in-the-united-states/. [Online; accessed 24-December-2019].

[27] Ozan K Tonguz and Rusheng Zhang. 2019. Harnessing vehicular broadcast communications: Dsrc-actuated traffic control. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[28] Norbert Varga, László Bokor, András Takács, József Kovács, and László Virág. 2017. An architecture proposal for V2X communication-centric traffic light controller systems. In *15th Int. Conf. on ITS Telecomm. (ITST)*. IEEE, 1–7.

[29] Darren B Ward, Eric A Lehmann, and Robert C Williamson. 2003. Particle filtering algorithms for tracking an acoustic source in a reverberant environment. *IEEE Transactions on speech and audio processing* 11, 6 (2003).

[30] Luana Chetcuti Zammit, Simon G Fabri, and Kenneth Scerri. 2019. Real-time parametric modeling and estimation of urban traffic junctions. *IEEE Transactions on Intelligent Transportation Systems* (2019).