Dynamic Watermarking-based Defense of Transportation Cyber-physical Systems

WOO-HYUN KO, BHARADWAJ SATCHIDANANDAN, and P. R. KUMAR,
Texas A&M University

The transportation sector is on the threshold of a revolution as advances in real-time communication, real-time computing, and sensing technologies have brought to fruition the capability to build Transportation Cyber-Physical Systems (TCPS) such as self-driving cars, unmanned aerial vehicles, adaptive cruise control systems, truck platoons, and so on. While there are many benefits that TCPSs have to offer, a major challenge that needs to be addressed to enable their proliferation is their vulnerability to cyber attacks. In this article, we demonstrate, using laboratory prototypes of TCPSs, how the approach of Dynamic Watermarking can secure them from arbitrary sensor attacks. Specifically, we consider two TCPSs of topical interest: (i) an adaptive cruise control system and (ii) a system of self-driving vehicles tracking given trajectories. In each of these systems, we first show how cyber attacks on sensors can compromise safety and cause collisions between vehicles in spite of the presence of a collision avoidance module in the system. We then apply the approach of Dynamic Watermarking and demonstrate that it detects attacks with "low" delay. Once an attack is detected, the controller can take appropriate control actions to prevent collisions, thereby guaranteeing safety in the sense of collision freedom.

CCS Concepts: • **Security and privacy** → *Intrusion/anomaly detection and malware mitigation*; *Intrusion detection systems*;

Additional Key Words and Phrases: Autonomous transportation systems, cybersecurity, driver assist systems, intelligent transportation systems, Dynamic Watermarking, secure control, Cyber-physical systems

ACM Reference format:
Woo-Hyun Ko, Bharadwaj Satchidanandan, and P. R. Kumar. 2019. Dynamic Watermarking-based Defense of Transportation Cyber-physical Systems. *ACM Trans. Cyber-Phys. Syst.* 4, 1, Article 12 (November 2019), 21 pages.
https://doi.org/10.1145/3361700

1 INTRODUCTION

The National Highway Traffic Safety Administration (NHTSA) estimates that, on average, about 94% of road accidents are caused by human errors, and the remaining 6% are caused by all other

This material is based upon work partially supported by the US Army Research Laboratory under Contract No. W911NF-19-2-0033, NSF under Contract Nos. CNS-1646449 and Science & Technology Center Grant CCF-0939370, the U.S. Army Research Office under Contract No. W911NF-15-1-0279, and NPRP grant NPRP 8-1531-2-651 from the Qatar National Research Fund, a member of Qatar Foundation.

Authors' addresses: W.-H. Ko, Room 161, Wisenbaker Engineering Building, Texas A&M University, College Station, Texas 77843; email: whko@tamu.edu; B. Satchidanandan, Room 111, Wisenbaker engineering building, Texas A&M University, College Station, Texas 77843; email: bharadwaj.s1990@gmail.com; P. R. Kumar, Room 331E, Wisenbaker, engineering building, Texas A&M University, College Station, Texas 77843.

2378-962X/2019/11-ART12 $15.00
https://doi.org/10.1145/3361700

factors combined, such as failure of critical components in the vehicle, bad weather, and so on. [1]. Besides the potential for automated transportation systems to reduce road accidents by eliminating human errors, they can also schedule traffic in real time to avoid congestion and increase network throughput. Though the safety and efficiency benefits that automated transportation systems offer appear attractive, their vulnerability to cyber attacks is a major concern that needs to be addressed if their large-scale deployment is to see the light of day. Of particular interest in this article is their vulnerability to sensor attacks. At high levels of autonomy as defined in Reference [7], certain aspects of the navigation are executed by the vehicle's control systems without any interference from the human driver, and in such a situation, malicious attacks on the sensors that provide situational awareness to the vehicles' controllers can lead to application of control inputs that result in collisions. An example of this scenario is an automated vehicle platoon in which the position sensor of the platoon leader is malicious, and it reports wrong position estimates of the platoon leader to the other vehicles in the platoon. Using appropriately designed attack strategies, the malicious sensor can fabricate measurements that cause collisions between vehicles in the platoon, which at high speeds could have catastrophic cascading effects. In fact, such sensor attacks have already been demonstrated on automated transportation systems in the recent past. As an example, a group of researchers has shown that certain off-the-shelf radio-emitting and sound-emitting devices can be used to deceive a Tesla S vehicle's autopilot into perceiving the presence of an obstacle where there exists none, and vice versa [2]. Reference [34] presents experimental results demonstrating the feasibility of a variety of attacks on vehicle sensors.

The problem of securing automated transportation systems lies in the broader realm of security of Cyber-Physical Systems (CPS). A CPS refers to a physical plant controlled using sensors, controllers, and actuators, all of which communicate with one another using an underlying communication network. Figure 1 illustrates such a CPS. Smart grids, process control systems (e.g., SCADA), and advanced manufacturing systems are other examples of cyber-physical systems. The increasing migration of critical industrial and national infrastructure towards interconnected systems using off-the-shelf networking and information technology solutions increases the vulnerability of these cyber-physical systems to security breaches. The Stuxnet attack [22], the Maroochy-Shire incident [8], and the attack on Davis-Besse nuclear power plant [18] demonstrate, among many other attacks, the practical feasibility of cyber attacks on industrial-grade networked control systems. Unless satisfactory security guarantees are provided for cyber-physical systems, there will be resistance to their large-scale implementation, which is needed to address critical societal and infrastructural needs.

In our prior work [38–40, 44], we developed the approach of Dynamic Watermarking to secure a cyber-physical system and established the security guarantees that it provides. Specifically, it is shown in these papers that Dynamic Watermarking can secure, in a very precise sense, many broad classes of linear systems from arbitrary attack strategies. One of the central results of these papers, termed as the fundamental security guarantee of Dynamic Watermarking, was also established in the context of a specific nonlinear dynamical model that describes the motion of a vehicle on a plane [19]. In this article, we apply the technique of Dynamic Watermarking to secure two transportation cyberphysical systems of topical interest, namely, (i) an adaptive cruise control system, or a vehicle platoon, and (ii) a system of self-driving vehicles. The former system consists of a set of vehicles that move "closely together" with some specified headway between them, and a platoon leader that follows some arbitrary trajectory. The vehicles in the platoon periodically share information such as their positions, velocities, and heading angles with one another using an underlying communication network. A control policy is designed to close the loop around this information to determine the acceleration, steering, and braking of each vehicle so the tracking objective is met. These functions are automated mainly to (i) improve safety by eliminating human errors and (ii) remove the latency introduced by human reaction times, which in turn allows for low headway and high traffic throughput. The latter system, viz., a system of self-driving vehicles, consists of a set of vehicles tracking a given position vs. time trajectory in an automated manner without any human intervention. Some of the results pertaining to this system were earlier reported in our conference paper [19]. In each of these systems, we first show how cyber attacks on sensors can compromise safety and cause collisions between vehicles in the system in spite of the presence of a collision avoidance module that is present in them. This is a result of the feeding of faulty sensor measurements to the collision avoidance module. We then implement Dynamic Watermarking in these systems to detect the attacks. Once an attack is detected, the controller can take appropriate control actions to prevent collisions. In our demonstrations, the controller simply halts the vehicles in the system whenever it detects an attack to prevent collisions. However, more general policies can also be used to close the loop around the attack detector's output to guarantee collision freedom.

The rest of this article is organized as follows: Section 2 provides an overview of related work on both the topic of cybersecurity of automated transportation systems as well as the broader topic of CPS security. Section 3 motivates and formulates the problem of CPS security in system-theoretic terms and provides a background on the idea of Dynamic Watermarking by describing it in the context of a simple perfectly observed linear system and stating the security guarantees that it provides. Section 4 contains a description of the laboratory testbed used for our demonstrations and the system model that is used in the subsequent section. To keep the article self-contained, Section 5 first presents a result reported in Reference [19] that extends the fundamental security guarantee of Dynamic Watermarking to the nonlinear system model that describes the transportation system considered in our demonstrations. The essential new aspects of the article are the laboratory demonstrations presented in this section that illustrate the efficacy and practical feasibility of Dynamic Watermarking in securing (i) an adaptive cruise control system and (ii) a system of self-driving vehicles. The theoretical security guarantees presented are contingent upon the attack detector performing two asymptotic tests, and Section 6 describes how these asymptotic tests can be converted into finite-time statistical tests that can be employed in real time with certain false alarm and misdetection rates. Section 7 contains some concluding remarks.

2 RELATED WORK

The security of intelligent transportation systems is a topic that has garnered significant attention from both academia and the industry in recent years. Various attack models and methodologies of attacks on automated vehicles are presented in Reference [33]. The issues of privacy and security brought to the fore by intelligent transportation systems are outlined in Reference [17]. Some potential solutions to guard against the identified security vulnerabilities are also presented. Demonstrations of remote cyber attacks on automobiles have also been conducted [3], [26]. One such demonstration that has received wide coverage in the popular press involves two hackers subverting an automobile and taking control of its steering and braking systems. This incident led to the automobile manufacturer recalling over a million units to patch the identified security vulnerabilities. Another similar incident involved a group of researchers spoofing the sensors of a state-of-the-art automated vehicle, thereby preventing it from detecting obstacles in its path [2]. These demonstrations reveal the scope of possible attacks on intelligent transportation systems and underscore the need for a principled approach to this problem.

A model-based detection scheme to mitigate insider attacks on a platoon controller to cause collisions is presented in Reference [14]. The fundamental idea is for each vehicle to use Dedicated Short-Range Communications (DSRC) to broadcast its state information so each vehicle can predict the behavior of the vehicle in front of it. If the actual behavior is different from what is predicted, then the vehicles switch to non-cooperative ACC mode to mitigate the impact of the attack. However, it is possible to devise attacks that exploit the system model while manipulating the reported observations. A model-based method cannot detect such attacks. Reference [15] considers an attack that modifies the control laws of individual vehicles by varying a single parameter and causes performance degradation of the platoon. Conditions under which the attack causes string instability are also derived. Similarly, Reference [13] presents a malicious attack that modifies the control law to cause vehicles in a platoon to exhibit oscillatory braking and acceleration, resulting in collisions. While these two papers demonstrate how cyber attacks can threaten the safety of a vehicle platoon, they do not address the problem of detecting and mitigating these attacks. Reference [9] illustrates that a connected vehicle platoon controlled by a CACC system is highly susceptible to attacks on the communication channel. It shows via computer simulations how a CACC system can be destabilized by a message falsification attack.

The problem of securing intelligent transportation systems lies in the broader realm of securing cyber-physical systems. The latter has been a topic of intense research in the past decade. Early work on CPS security focused on defining the problem and distinguishing it from the traditional problems of network and information security [12]. Mathematical models for catalogued attacks, such as the Denial-of-Service (DoS) attack and deception attack, are presented in Reference [11]. Fundamental limitations of static, dynamic, and active monitors for both detection and identification of attacks are derived in Reference [32]. A static monitor refers to an algorithm that does not possess knowledge about the system dynamics and uses only the output of the system to detect an attack. A dynamic monitor is assumed to have the knowledge of the system dynamics that generates the output, which it can use to detect and identify attacks. Finally, an active monitor, in addition to being a dynamic monitor, can also modify the system behavior by injecting additional input signals. The approach of Dynamic Watermarking falls under this category.

Reference [42] introduces the notions of the securable and the unsecurable subspaces of a linear system and shows that they have important operational meanings in the context of secure control. These can be thought of as the analogs of the controllable and the unobservable subspaces in a scenario where not all actuators and sensors may be honest. These subspaces, introduced in Reference [42] in the context of deterministic linear systems, are shown in References [41, 43] to have operational meanings also in the context of stochastic systems.

An attack space is defined in Reference [48] based on the knowledge that the adversary is assumed to have. Known attacks such as the replay attack, DoS attack, and deception attack are mapped into the attack space and are analyzed, and these attacks are implemented in a testbed of a quadruple-tank process to observe their consequences.

Several defense techniques have also been proposed to guard against malicious attacks. An algorithm for secure state estimation in the presence of malicious sensors and a technique to improve its resiliency are presented in Reference [16]. Reference [27] presents bounds on the state prediction error in the presence of malicious nodes and also an algorithm that achieves this bound. Zero-dynamics attacks refers to attacks that cannot be detected using input and output data. Reference [49] presents methods that can detect such attacks by perturbing the system parameters.

Many of the above techniques for CPS security can be classified as passive defense techniques. An alternate paradigm for CPS security is active defense, where the actuators expend energy to proactively detect attacks. The approach of Dynamic Watermarking is one such example, where honest actuators actively probe the system by injecting a random signal of arbitrarily small power whose realization is unknown to other nodes in the system. This random signal is known as the "watermark." It evokes a known response from the sensors in accordance with the plant dynamics, provided the sensors are honest. The honest actuators can therefore detect malicious sensors by checking if the measurements reported by the sensors are appropriately correlated with the watermark. In our prior work [38–40], we have shown that by using this approach, and subjecting the reported sequence of measurements to two particular tests of sensor veracity, it can be ensured that the malicious sensors, no matter what attack strategy they employ, do not distort the measurements beyond adding a zero power signal to the noise already entering the system. This result, known as the fundamental security guarantee of Dynamic Watermarking, was established for broad classes of linear systems. To the best of our knowledge, References [30, 31] are among the first papers to explore this technique to secure CPS. It has found applications in other scenarios as well [46, 50]. In this article, we demonstrate the efficacy of this approach in securing two transportation cyberphysical systems of topical interest, viz., the adaptive cruise control system and a system of self-driving vehicles.

3 THE DYNAMIC WATERMARKING APPROACH FOR LINEAR SYSTEMS

While most systems encountered in the real world are non-linear, including the vehicular cyber-physical systems considered in this article, the prior theoretical focus on linear systems is motivated by two reasons: (i) linear systems lend themselves to analytical tractability and (ii) the insights gained from analyzing linear systems often extend to more general nonlinear systems. Indeed, we present in Section 4 the extension of the theory to the nonlinear model describing the vehicular system considered in our demonstrations. Before that, we first provide the relevant background on securing linear systems. We show why an active defense such as watermarking is necessary by showing how a passive system can fail. Then, we show how watermarking can be used to secure linear systems.

Consider a networked control system as shown in Figure 1. At the core of the system is a physical plant with m inputs and n outputs. Suppose that the physical plant is abstracted as a stochastic linear dynamical system

$$\mathbf{y}[t+1] = A\mathbf{y}[t] + B\mathbf{u}[t] + \mathbf{w}[t+1], \tag{1}$$

where $\mathbf{y}[t] \in \mathbb{R}^n$ is the vector of measurements of the sensors at time t, $\mathbf{u}[t] \in \mathbb{R}^m$ is the vector of inputs applied by the actuators at time t, A and B are known matrices of appropriate dimensions describing the system dynamics, and $\mathbf{w}[t] \sim \mathcal{N}(0, \sigma_w^2 I)$ is the process noise assumed to be i.i.d. across time.

Since certain sensors in the system can be subverted, they need not report the measurements that they observe in a true fashion. Hence, we denote by $\mathbf{z}[t] \in \mathbb{R}^n$ the measurements reported by the sensors at time t, which may or may not be equal to the actual measurements $\mathbf{y}[t]$. If sensor i, $i \in \{1, \ldots, n\}$, is honest, then $z_i[t] = y_i[t]$ for all t, where $x_i[t]$ is used to denote the ith entry of vector $\mathbf{x}[t]$. In this article, we assume that all actuators are honest.

In a networked CPS, sensors and actuators exchange information with the help of an underlying communication network, some nodes of which may be malicious. Indicative of this, certain nodes are colored red in Figure 1. We do not assume that the communication network is reliable. We allow for the possibility that it is the communication network that has corrupted or distorted the sensor measurements. The Dynamic Watermarking approach is applicable irrespective of the source of the corruption of sensor measurements, since it detects erroneous measurements regardless of where the corruption takes place.

Our goal is to ensure that the malicious sensors do not affect the performance metric of the system, measured, for example, by a quadratic cost. We begin by first illustrating how malicious sensors can attack a cyberphysical system in the absence of active countermeasures, with the case of an idealized linear plant.

Example. Consider a single input single output (SISO) system described by

$$y[t+1] = 0.5y[t] + u[t] + w[t+1].$$

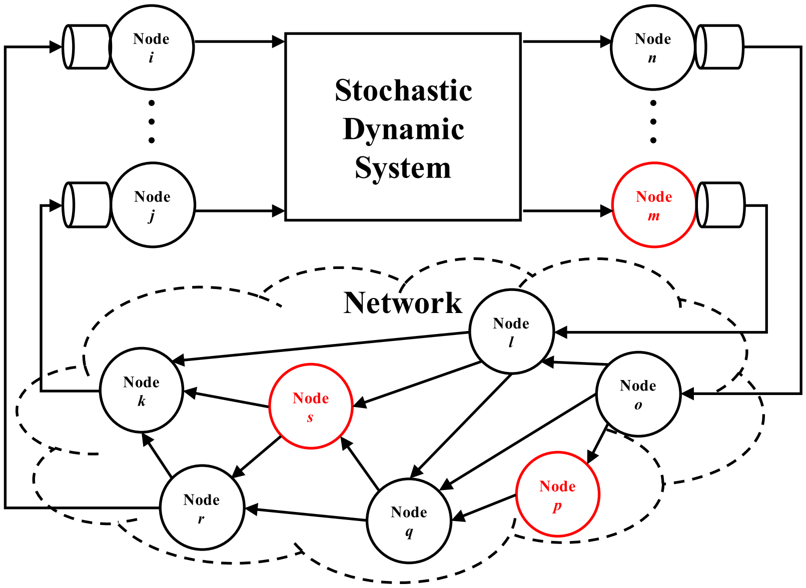

Fig. 1. A Networked Control System.

The minimum variance control law, which minimizes the variance of the output, is $u[t] = -0.5y[t]$ [20]. The actuator can only implement it if the sensor reports the correct measurements $y[t]$. Denote by $z[t]$ the measurements actually reported by the sensor. If the sensor is honest, then $z[t] \equiv y[t]$, but a malicious sensor may report $z[t] \neq y[t]$. The control policy applied by the actuator, $u_t^g(z^t)$, where $z^t := (z[0], z[1], \ldots, z[t])$, is consequently

$$u^g[t] = -0.5z[t].$$

If the sensor is honest, i.e., $z[t] \equiv y[t]$, then the variance of the output is equal to the process noise variance σ_w^2. Now, we show how a malicious sensor can avoid detection while simultaneously reporting false measurements. Suppose that the malicious sensor employs the following attack strategy. It generates the measurement to report at time t as

$$z[t] = 0.5z[t-1] + u^g(z^{t-1}) + w'[t],$$

where $w'[t] \sim \mathcal{N}(0, \sigma_w^2)$ is a random noise sequence generated by the sensor at each time t in an i.i.d. fashion and is independent of the process noise $\{w\}$. Note that this strategy allows the sensor to report measurements without even observing the actual output $\{y\}$. Since the sequence of measurements reported is consistent with the output that would have been observed had the process noise realization been $\{w'\}$ instead of $\{w\}$, and since $\{w'\}$ and $\{w\}$ have the same statistics, no algorithm can detect this attack. Consequently, if the actuator applies the input $u[t] = -0.5z[t]$, the real system evolves as

$$y[t+1] = 0.5(y[t] - z[t]) + w[t+1],$$

whose output variance is $\frac{5}{3}\sigma_w^2$, which is higher than the optimal variance σ_w^2.

The above example illustrates the effect that a malicious sensor can have on the system performance and shows that some active measure has to be taken to assure security. This motivates the employment of "Dynamic Watermarking" described next.

The basic idea of Dynamic Watermarking is for the honest actuators in the system to inject, over and above the control-policy-specified input, a small random noise into the system, which we will call as the "private excitation" of the actuator. We also refer to this private excitation as the "watermark." While the statistics of the watermark are made public so it is known to every sensor and actuator node in the system, including the malicious nodes, the actual realization of the watermark is not revealed by the actuators to any other node in the system.

Why does Watermarking help in detecting malicious sensors? Since the plant model is assumed to be known *a priori*, each actuator node knows exactly how its private excitation is transformed by the plant and appears at the output. Therefore, when a sensor reports the measurements that it observes, the actuators can check if their private excitation "comes back," i.e., if the reported measurements are appropriately correlated with the watermark that they inject. If not, then the actuators can declare that there are malicious sensors present in the system. In what follows, we make this precise.

Denote by $\mathbf{e}[t]$ the private excitation injected by the actuators at time t. Then, it follows from Equation (1) that

$$\mathbf{y}[t+1] = A\mathbf{y}[t] + B\mathbf{u}_t^g(\mathbf{z}^t) + B\mathbf{e}[t] + \mathbf{w}[t+1], \tag{2}$$

where $g_t(\mathbf{z}^t)$ is the control-policy-specified input at time t. Note that the control policy g_t is applied on the reported measurements $\mathbf{z}^t$ rather than on the actual measurements $\mathbf{y}^t$, since the actuators have access to only the reported measurements, which may not be equal to the actual measurements.

Each actuator injects a watermark that is distributed according to $\mathcal{N}(0, \sigma_e^2)$, chosen independently of all random variables in the system realized until that time. Consequently, $\mathbf{e}[t] \sim \mathcal{N}(0, \sigma_e^2 I)$, and we have from Equation (2) that for each t,

$$\mathbf{y}[t+1] - A\mathbf{y}[t] - B\mathbf{u}_t^g(\mathbf{z}^t) \sim \mathcal{N}(0, BB^T\sigma_e^2 + \sigma_w^2 I),$$

and for each $i \in \{1, \ldots, m\}$ and all t,

$$\mathbb{E}[e_i[t](\mathbf{y}[t+1] - A\mathbf{y}[t] - B\mathbf{u}_t^g(\mathbf{z}^t))] = B_{\cdot i}\sigma_e^2.$$

Based on the above observation, each actuator conducts the following tests on the reported measurements and declares the presence of malicious sensors if either of the tests fails.

(1) **Actuator Test 1:** The ith actuator node checks if the reported sequence of measurements satisfies

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} (\mathbf{z}[k+1] - A\mathbf{z}[k] - Bg_k(\mathbf{z}^k))(\mathbf{z}[k+1] - A\mathbf{z}[k] - Bg_k(\mathbf{z}^k))^T = \sigma_e^2 BB^T + \sigma_w^2 I_n. \tag{3}$$

(2) **Actuator Test 2:** The ith actuator node checks if the reported sequence of measurements satisfies

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} e_i[k](\mathbf{z}[k+1] - A\mathbf{z}[k] - Bg_k(\mathbf{z}^k)) = B_{\cdot i}\sigma_e^2, \tag{4}$$

where $B_{\cdot i}$ denotes the ith column of the matrix B.

The basis for the test is that if all sensors are honest, then the reported sequence of measurements will pass the above tests almost surely. The following theorem, termed as the fundamental security guarantee of Dynamic Watermarking, establishes that if the reported sequence of measurements pass the above tests, then any malicious sensor present cannot distort the actual measurements beyond adding a zero power sequence to the process noise.

Definition 1. The *additive distortion* introduced by the malicious sensors at time $t + 1$, denoted by $\mathbf{v}[t + 1]$, is

$$\mathbf{v}[t+1] := \mathbf{z}[t+1] - A\mathbf{z}[t] - Bg_t(\mathbf{z}^t) - B\mathbf{e}[t] - \mathbf{w}[t+1].$$

Note that, in particular, if $\mathbf{z}[t] \equiv \mathbf{y}[t]$, then $\mathbf{v}[t] \equiv 0$.

Theorem 6 of Reference [39]. *Suppose that the matrix B is of rank n. If the reported sequence of measurements passes the tests of Equations (3) and (4), then the additive distortion is of zero power, i.e.,*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} ||\mathbf{v}[k]||^2 = 0. \tag{5}$$

Note that if there were no process noise $\mathbf{w}[t]$, then given the initial condition $\mathbf{y}[0]$, the actuators could compute the output of the system at any time t based on the inputs applied. Hence, the essential "new" information that a sensor provides is the actual noise that enters the system at each time. Now, since $\mathbf{z}[t+1] - A\mathbf{z}[t] - Bg_t(\mathbf{z}^t) - B\mathbf{e}[t] = \mathbf{w}[t+1] + \mathbf{v}[t+1]$, and the LHS of this can be computed by the actuators, $\mathbf{v}[t+1]$ has the physical meaning of the distortion that the malicious sensors add to the process noise sequence, which, including the initial state $\mathbf{x}[0]$, are the basic random variables of the system. For this reason, we call $\{\mathbf{v}\}$ the additive distortion introduced by the sensor. What the above theorem says is that the malicious sensors cannot distort this sequence of basic random variables beyond adding a zero power sequence to it.

In the following sections, we first present the extension of the above result to the nonlinear vehicular cyber-physical systems that we consider in this article, and then demonstrate the efficacy of the approach in securing prototypical transportation cyber-physical systems from sensor attacks.

4 SECURITY OF TRANSPORTATION CYBER-PHYSICAL SYSTEMS

Before describing the vehicular cyber-physical systems that we consider in this article and their security, we first provide a brief account of the experimental setup on which the experiments are conducted.

4.1 Experimental Setup

The cyber-physical systems laboratory at Texas A&M University includes a testbed of a transportation cyber-physical system. It provides the capability to rapidly prototype various control policies for automated transportation systems. The testbed setup is shown in Figure 2.

At the heart of the system is a set of remote-controlled cars that are controlled by computers, marked as low-level car controllers in Figure 2. Ten cameras mounted at various locations in the setup capture the image of the rectangular space in which the vehicles move. These cameras transmit the images that they capture to a vision server, which computes the coordinates of each vehicle in the system as well as their orientation with respect to a reference ray once every 100ms. The cameras along with the vision server, therefore, serve as the position sensors in the system. These position and orientation estimates are fed to the controller, which computes the control input for each vehicle in accordance with a predetermined control law. The vehicles in our testbed directly allow for their translational and the rotational speeds to be controlled. A lower-level controller determines the appropriate linear and angular accelerations that should be applied to achieve the specified speeds. Owing to the vehicles being light in weight, the motors in the vehicles accelerate them to the desired speeds at time scales much shorter than the sampling duration. Consequently, as described in Section 4.2, we use a kinematic model to describe the vehicles' dynamics. The control law that computes the control inputs at each time is determined by the control objective, which in turn is specified by a supervisory control layer. In a system of self-driving cars, for example, the

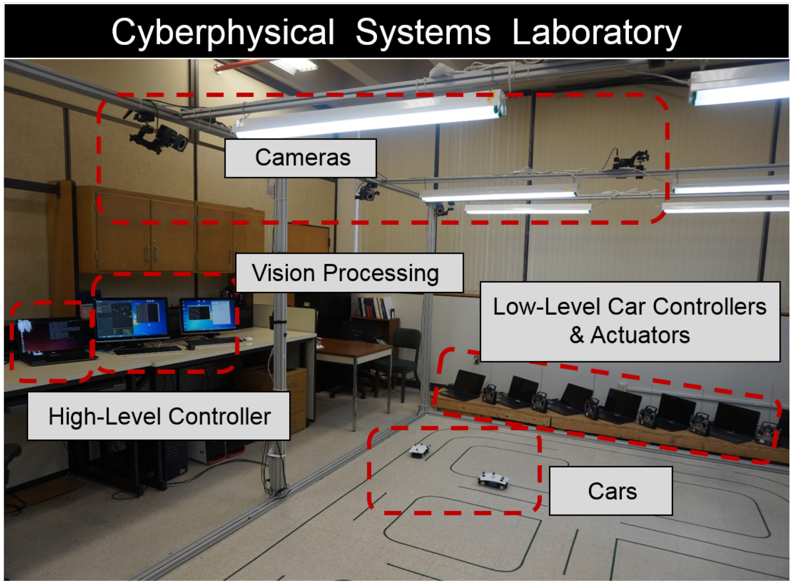
Cyberphysical Systems Laboratory
Cameras
Vision Processing
Low-Level Car Controllers & Actuators
High-Level Controller
Cars

Fig. 2. Testbed facility for transportation CPS.

supervisory control layer computes the position vs. time trajectory that each vehicle in the system should track.

One of the key components of the system that provides provable safety guarantees is the collision avoidance (CA) module integrated into the system. The functioning of this module is as follows: The control inputs computed by the controller, instead of being relayed directly to the actuators, are first screened by the CA module during which phase it checks whether the application of these control inputs would lead to any collision or not. For this purpose, the CA module fetches information regarding the current position of each vehicle from the vision server and, based on its knowledge of the system dynamics, computes the result of applying the specified control inputs. If it detects an impending collision, then it instructs the actuators to halt the vehicles. Otherwise, it relays the control inputs computed by the controller as such to the actuators. It is established in Reference [37] that this system indeed guarantees collision freedom.

We implement on the above testbed two prototypical transportation cyber-physical systems of topical interest: (i) an adaptive cruise control system or a platoon and (ii) a system of self-driving vehicles.

4.2 System Model

In both the vehicular CPSs that we consider, the kinematic equations of motion for the ith vehicle can be expressed as

$$\begin{aligned} x_i[t+1] &= x_i[t] + \tau v_i[t]\cos(\theta_i[t]) + \tau\cos(\theta_i[t])w_i[t+1], \\ y_i[t+1] &= y_i[t] + \tau v_i[t]\sin(\theta_i[t]) + \tau\sin(\theta_i[t])w_i[t+1], \\ \theta_i[t+1] &= \theta_i[t] + \tau\omega_i[t] + \tau w_{i,\theta}[t+1], \end{aligned} \tag{6}$$

where $x_i[t], y_i[t], \theta_i[t]$ denote the x-coordinate, y-coordinate, and the orientation of the ith vehicle at time t, $v_i[t]$, and $\omega_i[t]$ are the *desired* translational speed and rotational speed, respectively, of the ith vehicle at time t, and τ is the sampling time period, fixed to be 100 ms in our experiments. The random variables $w_i[t+1]$ and $w_{i,\theta}[t+1]$ in (6) denote the "process noise" arising from various factors such as air drag, road friction, noise in the applied control inputs, and so on. Since air drag and road friction are not of significant magnitude in our prototype but are in fact key phenomena that affect a real-world transportation CPS, we artificially inject these noises as a part of the control input in our demonstrations. We model all noise in the system as being normally distributed, so for all i, $w_i[t] \sim \mathcal{N}(0, \sigma_x^2)$ and $w_{i,\theta}[t] \sim \mathcal{N}(0, \sigma_\theta^2)$, and all processes are independent and i.i.d. across time. Clearly, (6) describes a nonlinear model with state $[x_i[t] \;\; y_i[t] \;\; \theta_i[t]]^T$. In Section 5, we extend the fundamental security guarantee of dynamic watermarking to this class of nonlinear models.

5 DYNAMIC WATERMARKING FOR TRANSPORTATION CPS

We use notation similar to what was used in Section 3 for linear systems, viz., we denote by $z_{i,x}[t], z_{i,y}[t]$, and $z_{i,\theta}[t]$ the measurements reported by the malicious sensors for $x_i[t], y_i[t]$, and $\theta_i[t]$, respectively. Note that, since honest nodes in the system can exchange the reported measurements to check for consistency, the malicious sensors are constrained to report the same measurements to all honest nodes in the system if they wish to remain undetected. Furthermore, in this article, we restrict attention to the case when only the position sensors are malicious, and not the sensors reporting the orientation of the vehicles. As we show shortly, misreporting just the position is sufficient to compromise safety and cause collisions.

In our laboratory prototype, the control variables in system (6) are the linear and the angular speed of each vehicle, i.e., the quantities $v_i[t]$ and $\omega_i[t]$. The low-level controller employs Model Predictive Control to evaluate the control inputs at each time. Once the controller computes the control-policy-specified input, it superimposes a private excitation on the input before relaying it to the CA module, and ultimately the actuators. We denote by $v_i^g[t]$ and $\omega_i^g[t]$ the control-policy-specified input at time t, and by $e_{i,v}[t]$ and $e_{i,\omega}[t]$ the private excitation superimposed on $v_i^g[t]$ and $\omega_i^g[t]$, respectively. Consequently, the closed-loop system evolves as

$$\begin{aligned}x_i[t+1] &= x_i[t] + \tau v_i^g[t]\cos(\theta_i[t]) + \tau e_{i,v}[t]\cos(\theta_i[t]) + \tau w_i[t+1]\cos(\theta_i[t]),\\ y_i[t+1] &= y_i[t] + \tau v_i^g[t]\sin(\theta_i[t]) + \tau e_{i,v}[t]\sin(\theta_i[t]) + \tau w_i[t+1]\sin(\theta_i[t]),\\ \theta_i[t+1] &= \theta_i[t] + \tau\omega_i^g[t] + \tau e_{i,\omega}[t] + \tau w_{i,\theta}[t+1].\end{aligned} \tag{7}$$

Similar to tests (3) and (4) developed for linear systems, the controller performs the following two tests to detect the presence of malicious sensors in the above system. While the tests below are specified only for $\{z_{i,x}\}$, analogous tests are also performed for $\{z_{i,y}\}$ and $\{z_{i,\theta}\}$.

(1) **Test 1:** The controller checks if

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}(z_{i,x}[k+1] - z_{i,x}[k] - \tau\cos(z_{i,\theta}[k])v_i^g[k] - \tau\cos(z_{i,\theta}[k])e_{i,v}[k])^2 = \widetilde{\sigma}_x^2, \tag{8}$$

where $\widetilde{\sigma}_x^2$ is defined as the limit of the LHS of (8) when $\{z_{i,x}\} \equiv \{x_i\}$ and $\{z_{i,\theta}\} \equiv \{\theta_i\}$.

(2) **Test 2:** The controller checks if

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}(z_{i,x}[k+1] - z_{i,x}[k] - \tau\cos(z_{i,\theta}[k])v_i^g[k])^2 = \sigma_c^2, \tag{9}$$

where σ_c^2 is defined as the limit of the LHS of (9) when $\{z_{i,x}\} \equiv \{x_i\}$ and $\{z_{i,\theta}\} \equiv \{\theta_i\}$. In that case, the LHS of Test 1 becomes

$$\frac{1}{t}\sum_{k=0}^{t-1}(\tau\cos(\theta_i[k])w_i[k+1])^2, \tag{10}$$

and that of Test 2 becomes

$$\frac{1}{t}\sum_{k=0}^{t-1}(\tau\cos(\theta_i[k])e_{i,v}[k]+\tau\cos(\theta_i[k])w_i[k+1])^2. \tag{11}$$

Note that the above are sums of independent but non-identically distributed random variables. For the nominal trajectories required to be followed by the vehicles, we assume that the limits of these two series exist. Hence,

$$\widetilde{\sigma}_x^2 := \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}(\tau\cos(\theta_i[k])w_i[k+1])^2, \tag{12}$$

and

$$\sigma_c^2 := \lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}(\tau\cos(\theta_i[k])(e_{i,v}[k]+w_i[k+1]))^2. \tag{13}$$

These quantities were computed experimentally in our demonstrations.

The following theorem extends the fundamental security guarantee of Dynamic Watermarking to the nonlinear transportation cyber-physical system.

Theorem 5.1. *Define*

$$v_{i,x}[t+1] := z_{i,x}[t+1]-z_{i,x}[t]-\tau\cos(\theta_i[t])v_i^g[t]-\tau\cos(\theta_i[t])e_{i,v}[t]-\tau\cos(\theta_i[k])w_i[t+1]. \tag{14}$$

If the reported sequence of measurements satisfy (8) and (9), then,

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t-1}v_{i,x}^2[k+1]=0. \tag{15}$$

Proof. Since the sequence of reported measurements $\{z_{i,x}\}$ satisfy (8), we have using (14) that

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t}(\tau\cos(\theta_i[k])w_i[k+1]+v_{i,x}[k+1])^2=\widetilde{\sigma}_x^2. \tag{16}$$

Expanding the above and using (12), we get

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t}v_{i,x}^2[k+1]+2\tau\cos(\theta_i[k])w_i[k+1]v_{i,x}[k+1]=0. \tag{17}$$

Since the reported measurements also satisfy (9), it follows that

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t}(\tau\cos(\theta_i[k])w_i[k+1]+\tau\cos(\theta_i[k])e_{i,v}[k]+v_{i,x}[k+1])^2=\sigma_c^2. \tag{18}$$

Expanding and using (13) and (17), we get

$$\lim_{t\to\infty}\frac{1}{t}\sum_{k=0}^{t}\cos(\theta_i[k])e_{i,v}[k]v_{i,x}[k+1]=0. \tag{19}$$

Define the σ−algebra $\mathcal{F}_k := \sigma(x^k, y^{k-1}, \theta^{k-1}, e_{i,v}^{k-2}, z^k)$, $\widehat{w}_i[k] := E[w_i[k] \Big| \mathcal{F}_k]$, and $\widetilde{w}_i[k] := w_i[k] - \widehat{w}_i[k]$. Then, for k such that $\cos(\theta_i[k]) \neq 0$,

$$\begin{aligned}\widehat{w}_{i,x}[k+1] &= \frac{\sigma_w^2 (\tau \cos \theta_i[t])^{-1}}{\sigma_e^2 + \sigma_w^2}(x_i[t+1] - x_i[t] - \tau v_i^g[t] \cos \theta_i[t]) \\ &= \frac{\sigma_w^2}{\sigma_e^2 + \sigma_w^2}(e_{i,v}[k] + w_i[k+1]).\end{aligned}$$

Now,

$$\begin{aligned}\sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) w_i[k+1] v_{i,x}[k+1] &= \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) \widehat{w}_i[k+1] v_{i,x}[k+1] \\ &\quad + \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) \widetilde{w}_i[k+1] v_{i,x}[k+1] \\ &= \beta \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) e_{i,v}[k] v_{i,x}[k+1] \\ &\quad + \beta \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) w_i[k+1] v_{i,x}[k+1] \\ &\quad + \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) \widetilde{w}_i[k+1] v_{i,x}[k+1], \qquad (20)\end{aligned}$$

where $\beta := \frac{\sigma_w^2}{\sigma_e^2 + \sigma_w^2} < 1$. Rearranging the above yields

$$\begin{aligned}\sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) w_i[k+1] v_{i,x}[k+1] &= \frac{\beta}{1-\beta} \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) e_{i,v}[k] v_{i,x}[k+1] \\ &\quad + \frac{1}{1-\beta} \sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) \widetilde{w}_i[k+1] v_{i,x}[k+1]. \qquad (21)\end{aligned}$$

Now, $(w_i[k+1], \theta_i[k]) \in \mathcal{F}_{k+2}$. Furthermore, $\widehat{w}_i[k+1] \in \mathcal{F}_{k+1} \subset \mathcal{F}_{k+2}$. Consequently, $(\cos \theta_i[k] \widetilde{w}_i [k+1] \in \mathcal{F}_{k+2})$. It also follows from the definition of $\widetilde{w}_i[k]$ that $\mathbb{E}[(\cos \theta_i[k+1] \widetilde{w}_i[k+2] | \mathcal{F}_{k+2})] = 0$. It follows that $(\cos \theta_i[k] \widetilde{w}_i[k+1], \mathcal{F}_{k+2})$ is a Martingale Difference Sequence. Also, $v_{i,x}[k+1] \in \mathcal{F}_{k+1}$, since $v_{i,x}[k+1] = z_{i,x}[k+1] - z_{i,x}[k] - (x_i[k+1] - x_i[k])$. So, the Martingale Stability Theorem (MST) [21] applies, and we have

$$\sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} \cos(\theta_i[k]) \widetilde{w}_i[k+1] v_{i,x}[k+1] = o\left(\sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} v_{i,x}^2[k+1] \right) + O(1).$$

Substituting the above in (21), and using (17), we obtain

$$\sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} v_{i,x}^2[k+1] + \frac{2\tau\beta}{1-\beta}\sum_{\substack{k=0,\\ \cos(\theta_i[k])\,\neq\, 0}}^{t-1} \cos(\theta_i[k])e_{i,v}[k]v_{i,x}[k+1] + o\left(\sum_{\substack{k=0,\\ \cos(\theta_i[k])\neq 0}}^{t-1} v_{i,x}^2[k+1]\right) + O(1) + \sum_{\substack{k=0,\\ \cos(\theta_i[k])=0}}^{t-1} v_{i,x}^2[k+1] = o(t). \tag{22}$$

Dividing the above by t, taking the limit as $t \to \infty$, and invoking (19) completes the proof. □

Remark. While the above theorem establishes that the additive distortion $\{v_{i,x}\}$ is of zero power, it follows by an appropriate choice of coordinates that the additive distortion $\{v\}$ is of zero power along any direction in the plane.

While the tests (8) and (9) are specified as asymptotic tests, finite time versions of these tests can be developed using standard approaches, as described in Section 6. In our demonstrations, we use such statistical tests to detect the presence of malicious nodes.

Having established the theoretical guarantees in the context of transportation CPS, we now turn to our laboratory demonstrations demonstrating the practical feasibility of Dynamic Watermarking in securing prototypical transportation systems.

5.1 Adaptive Cruise Control

We first consider the case of an adaptive cruise control system consisting of four vehicles. The control objective in this system is for each vehicle in the system to follow the vehicle ahead of it. Only the vehicle that is in the lead is assigned a trajectory to be followed. The rest of the vehicles are required to follow the vehicle that is ahead of them with a specified headway. The system allows all vehicles to share their state information with all other vehicles using an underlying communication network.

The system consists of two control layers: a supervisory control layer that generates waypoints that each of the vehicles should follow (along with the time instants at which each of these waypoints is to be reached) and a low-level feedback controller that computes in real time the control input that needs to be applied to each of the vehicles to achieve their control objective. The supervisory controller of the platoon leader generates its waypoints based on a user-specified trajectory. The waypoints for the other vehicles are calculated in real time in the following manner based on their respective distances to the vehicle ahead of them.

Let $z_i[t] = [z_{i,x}[t], z_{i,y}[t], z_{i,\theta}[t]]^T$ be the reported state of the ith vehicle at time t. Each vehicle i at time t, based on the history of states of the vehicle ahead of it—say, vehicle $i-1$—makes a prediction $\widehat{z}_{i-1}[t+1]$ of the state of vehicle $i-1$ at the next epoch using a kinematic prediction model. Once the controller computes the prediction at time t, the waypoint $(x_i^d[t+1], y_i^d[t+1])$ for vehicle i at time $t+1$ is computed such that it is at a distance Δ behind vehicle $i-1$ at time $t+1$, i.e.,

$$x_i^d[t+1] = \widehat{z}_{i-1,x}[t+1] - \Delta\cos(\widehat{z}_{i-1,\theta}[t+1]), \tag{23}$$

$$y_i^d[t+1] = \widehat{z}_{i-1,y}[t+1] - \Delta\sin(\widehat{z}_{i-1,\theta}[t+1]), \tag{24}$$

$$\theta_i^d[t+1] = \widehat{z}_{i-1,\theta}[t+1]. \tag{25}$$

Once the desired waypoints are computed, the low-level controller of each vehicle computes the control inputs to minimize the tracking error.

The nominal functioning of the cruise control system in the absence of attacks is shown in the accompanying video [6]. The leader vehicle travels on a rectangular trajectory, and the second, third, and fourth vehicles follow the vehicle that is ahead of them with a constant predetermined headway. The video demonstrates the string stability of the platoon in the absence of attacks.

We then set the position sensor reporting the position of the platoon leader to be malicious. A “sinusoidal attack strategy,” described next, is employed by the sensor while reporting the $x-$coordinate of that vehicle’s position.

Let T_A be the time at which the attack begins. Then, the measurements reported by the malicious sensor are given by

$$z_{1,x}[t] = \begin{cases} x_1[t], \quad \mathit{if} \quad t < T_A, \\ z_{1,x}[t-1] + (x_1[t] - x_1[t-1]) \\ \qquad\qquad + \eta \cos(\frac{2\pi}{p} t), \quad \mathit{if} \quad t \geq T_A, \end{cases}$$

where p, η are arbitrary constants that can be chosen by the malicious sensor. In our demonstration, we have set $\eta = 40$ and $p = 4\text{K}$. These values have been tuned to cause collisions between vehicles. The video [6] shows how this attack leads to string instability and collisions between the vehicles.

We then implement Dynamic Watermarking, so the inputs applied to the system are $v_i[t] = v_i^g[t] + e_{i,v}[t]$, and $\omega_i[t] = \omega_i^g[t] + e_{i,\omega}[t]$ for all $i \in \{1, \dots, 4\}$. The aforementioned attack strategy is then carried out by the malicious sensor. The controller then conducts a finite-time counterpart of the tests (8) and (9) and declares maliciousness if either of these tests fails. Specifically, it computes at each time t the attack indicator signal corresponding to Test 1 as

$$\gamma_1[t] := \frac{1}{t}\sum_{k=0}^{t-1}(z_{i,x}[k+1] - z_{i,x}[k] - \tau\cos(z_{i,\theta}[k])v_i^g[k] - \tau\cos(z_{i,\theta}[k])e_{i,v}[k])^2,$$

and the attack indicator signal corresponding to Test 2 as

$$\gamma_2[t] := \frac{1}{t}\sum_{k=0}^{t-1}(z_{i,x}[k+1] - z_{i,x}[k] - \tau\cos(z_{i,\theta}[k])v_i^g[k])^2.$$

It then compares these signals against predefined thresholds, computed based on the maximum tolerable false alarm rate, and declares maliciousness if the attack indicator signals exceed their respective thresholds. The final segment of the video [6] demonstrates the Dynamic Watermarking–based tests detecting the attack well before a collision takes place and consequently signaling the actuators to halt the vehicles to restore safety.

Figure 3 and Figure 4 show the evolution of the attack indicator signals corresponding to Test 1 and Test 2, respectively, in the presence and absence of attacks. Comparing them shows the efficacy of employing a threshold-based detector to detect the attack.

5.2 Securing Automated Vehicle Platoons from Replay Attacks

The replay attack is one of the most widely studied attacks in the literature [25, 28, 30]. This was also the attack strategy that was employed in Stuxnet, which crippled the centrifuges in Iran’s nuclear facility [22]. The replay attack has also been studied and carried out in the context of a variety of other systems such as communication networks [10], video surveillance systems [36], smart grids [29], and so on (see Reference [24] and references therein). In this section, we report the results of a laboratory demonstration of the replay attack on an automated vehicle platoon. Specifically, we show that an attack as simple as the replay attack, which can be carried out by the adversary without any knowledge of the system parameters such as the plant dynamics, control policy, and so on, and which can in fact be carried out even without subverting the sensors, but

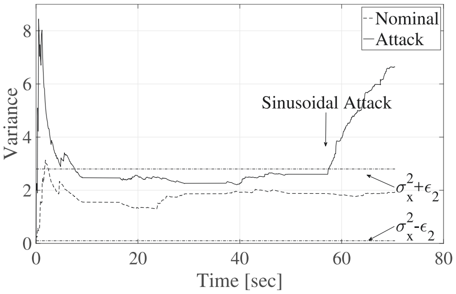

Fig. 3. Attack indicator signal corresponding to Test 1 under the Sinusoidal attack.

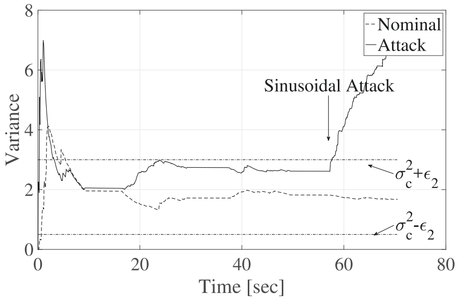

Fig. 4. Attack indicator signal corresponding to Test 2 under the Sinusoidal attack.

by just hijacking the communication network and replaying the packets can lead to collisions among vehicles if appropriately timed. Following this, we implement Dynamic Watermarking in the platoon and show how it can detect the replay attack "quickly," i.e., with a detection delay lower than the headway between any two vehicles in the platoon, and thereby restore safety. Video footage of this demonstration can be found in Reference [5].

Our experimental setup consists of a three-vehicle platoon, with the platoon leader required to follow a specified rectangular trajectory, a second vehicle required to follow the platoon leader with a specified headway, and a third vehicle required to follow the second vehicle with the same headway. As before, the supervisory layer generates the trajectory that the platoon leader is to follow. The sensor measuring the platoon leader's position relays its observations to the vehicle that is following it, which in turn uses a Model Predictive Controller to compute the control inputs that would steer the vehicle to follow the platoon leader with the specified headway. Similarly, the third vehicle uses the position estimates reported by the vehicle ahead of it to compute the appropriate control inputs.

The accompanying video in Reference [5] shows the behavior of the platoon when under a replay attack. The attack involves a malicious sensor recording the measurements that it observes for a certain period of time during nominal operation of the system and replaying those measurements in a constant loop during the attack. We have set the position sensor of the platoon leader to be malicious, so the measurements $\{z_{1,x}\}$ reported by it is

$$z_{1,x}[t] = \begin{cases} x_1[t], & if \quad t < T_A, \\ x_1[t-\tau] & if \quad t \geq T_A, \end{cases}$$

where $\tau > 0$ is a parameter that can be chosen by the adversary, and T_A is the time at which the attack is initiated. In our demonstration, the malicious sensor stores all past measurements and begins replaying them from an appropriate time as soon as the platoon leader halts, so the vehicles that follow it are deluded into estimating that they are following the platoon leader with the specified headway. This leads to a collision between the vehicles, as shown in the accompanying video [5], which the in-built collision avoidance module in the system cannot prevent, owing to it filtering incorrect position information.

The final segment of the video demonstrates a scenario in which the controller of each vehicle implements Dynamic Watermarking and conducts finite-time analogs of the two tests (8) and (9) to detect maliciousness. Specifically, the controller computes, based on the reported measurements, the attack indicator signal corresponding to Test 1 over a moving window of length L as

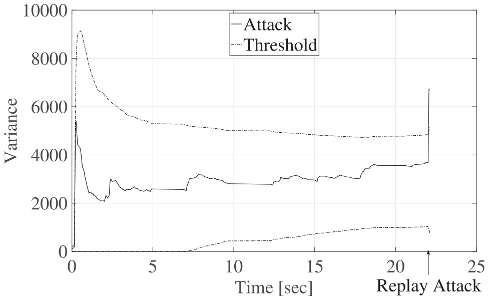

Fig. 5. Evolution of attack indicator signal corresponding to Test 1 under the Replay attack.

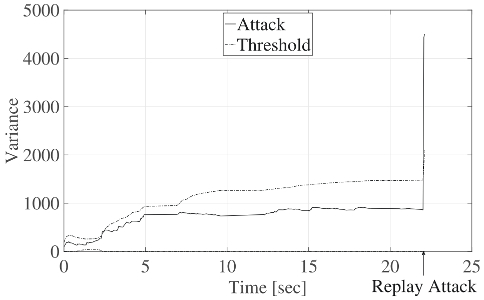

Fig. 6. Evolution of attack indicator signal corresponding to Test 2 under the Replay attack.

$$\delta_1[t] := \frac{1}{L}\sum_{k=t-L+1}^{t}(z_{i,x}[k] - z_{i,x}[k-1] - \tau\cos z_{i,\theta}[k-1]v_i^g[k-1] - \tau\cos(z_{i,\theta}[k-1])e_{i,v}[k-1])^2,$$

and similarly the attack indicator signal corresponding to Test 2 as

$$\delta_2[t] := \frac{1}{L}\sum_{k=t-L+1}^{t}(z_{i,x}[k] - z_{i,x}[k-1] - \tau\cos z_{i,\theta}[k-1]v_i^g[k-1])^2.$$

Also computed offline are the statistics of the sequences $\{\delta_1\}$ and $\{\delta_2\}$ that would be obtained under nominal, i.e., attack-free, operating conditions. Note that these are non-stationary processes, but their statistics at each time t depend only on the L-length sequence $\{\theta_i[t-L+1],\dots,\theta_i[t]\}$, which is known once the vehicle's nominal trajectory is known for that particular time horizon. The attack indicator signals $\delta_1[t]$ and $\delta_2[t]$ that are computed in real time are compared against their (time-varying) mean values, with thresholds chosen according to a tolerable false alarm rate.

The evolution of the attack detection signals $\{\delta_1\}$ and $\{\delta_2\}$, along with their time-varying thresholds, are graphed in Figure 5 and Figure 6, respectively. Note that even such simple finite-time analogs of the asymptotic tests (8) and (9) yield low attack detection delays.

5.3 System of Self-driving Vehicles

The second system that we consider is that of a collection of self-driving vehicles whose control objective is to track a given position vs. time trajectory. A centralized supervisory control layer plans the trajectory that each vehicle should track. In our demonstration, we fix this trajectory to be elliptic and consider a system consisting of two vehicles. Unlike the case of adaptive cruise control that we considered above, in this demonstration, each vehicle uses only its own position to close the loop. As always, the CA module present in the system ensures that no collisions occur between the vehicles when there are no malicious entities in the system. The tracking performance under nominal operation and the efficacy of the CA module is first demonstrated in the accompanying video [4].

To carry out the attack, we subvert the position sensor of one of the vehicles—say vehicle-2—and report distorted measurements of the x-coordinate of that vehicle's position. Specifically, let T_A be the time at which the attack begins. Then, the sensor reports

$$z_{2,x}[t] = \begin{cases} x_2[t], & \textit{if} \quad t < T_A, \\ x_2[t] + \delta, & \textit{if} \quad t = T_A, \\ z_{2,x}[t-1] + \tau\cos(\theta_2[t-1])v_2^g[t-1] + n[t-1], & \textit{if} \quad t > T_A, \end{cases} \tag{26}$$

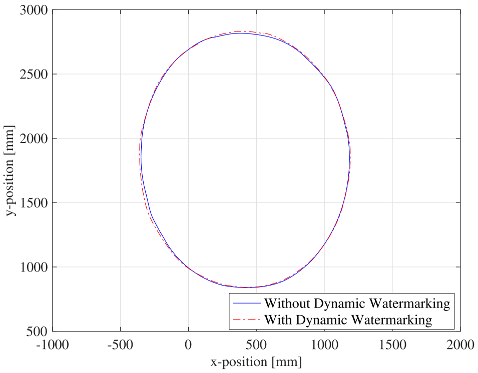

Fig. 7. Trajectory of a vehicle in the presence and absence of Dynamic Watermarking.

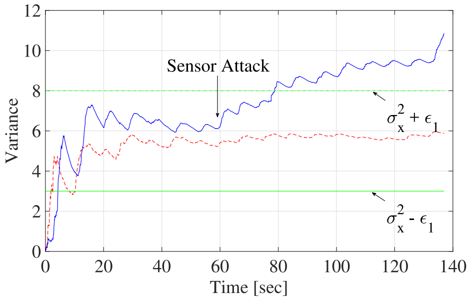

Fig. 8. Evolution of the attack indicator signal corresponding to Test 1.

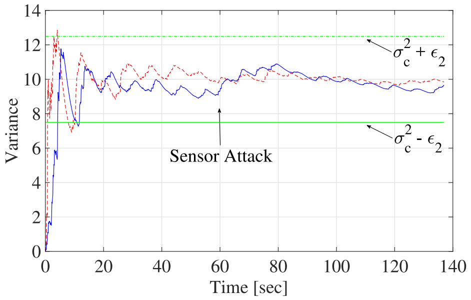

Fig. 9. Evolution of the attack indicator signal corresponding to Test 2.

where δ is a bias that the malicious sensor introduces, $v_2^g[t]$ is the control-policy-specified input, and $n[t] \sim \mathcal{N}(0, \sigma_x^2)$ is i.i.d. across time. Note that this attack strategy leads to a reported noise sequence that has the same statistics as the actual noise sequence $\{w_i[t+1]\}$ and, consequently, *no* detection algorithm can detect this attack. The accompanying video in Reference [4] shows how this attack leads to a collision between the vehicles.

We now implement Dynamic Watermarking and demonstrate how it can aid in detecting this attack. Figure 7 shows the trajectories of the vehicles in the presence and absence of watermarking. Note that the deviation of the vehicles from the reference trajectory due to watermarking is marginal.

The attack strategy employed by the malicious sensors is the same as in (26) except that the malicious sensor now chooses $n[t] \sim \mathcal{N}(0, \tau^2 \cos^2(\theta_i[t])\sigma_e^2 + \sigma_x^2)$ to account for the additional variance introduced by the watermark, thereby passing Test 2 above. The reported measurements are subjected to a finite-time version of the two tests (8) and (9) by the controller. The same finite-time tests described in Section 5.1 are used in this detector. The resulting evolution of the test statistics are plotted in Figure 8 and Figure 9. Once the attack is initiated, the attack indicator signal corresponding to Test 1 crosses a predetermined threshold, which leads to the attack being detected. Once the controller detects the attack, it halts the vehicles as shown in the video [4], thereby preventing collision.

As described above, in our demonstrations, we have used two particular finite-time statistical equivalents of the tests (8) and (9). In the next section, we present a general framework by which the asymptotic tests can be converted into finite-time statistical tests to meet a given objective.

6 FINITE-TIME STATISTICAL TESTS FOR ATTACK DETECTION

We discuss in this section how the tests of Dynamic Watermarking (8) and (9) presented in an asymptotic form can be converted into finite-time statistical tests that can be implemented in real time with certain false alarm and misdetection rates.

A standard framework using which finite-time Dynamic Watermarking–based detectors can be designed is that of quickest change detection. The quickest change detection problem consists of three components [51]:

(1) A stochastic process that can be observed,
(2) A random time τ, referred to as the change time, at which the statistics of the process changes from one distribution to another, and
(3) A change detection algorithm that, based on the observations of the process, declares at each time t whether $t \geq \tau$, i.e., the change has occurred, or $t < \tau$.

A vast amount of literature exists on this problem, which has wide applications in fields such as quality control, network security, manufacturing systems, and so on.

We now elaborate how the problem of designing finite-time statistical tests for Dynamic Watermarking maps to the quickest change detection problem. In the context of Dynamic Watermarking, the stochastic process under observation is the sequence of measurements $\{z\}$ reported by the sensors, which could possibly be malicious. As Theorem 5.1 shows, in the presence of Dynamic Watermarking, the adversary cannot introduce any significant distortion without causing a change in the statistics of the stochastic process that is under test, no matter what attack strategy it chooses to employ. Consequently, the time at which the statistics of the process changes, i.e., the change point, is simply the time at which the attack is initiated. Thus, formulating our problem as one of quickest change detection provides us with a mature framework for developing finite-time statistical tests that can be implemented in real time to optimize a given objective.

Associated with any such change detection algorithm is a trade-off between two important performance metrics, viz., the detection delay and the false alarm rate. A vast literature exists to examine this trade-off (see References [23, 35, 45, 47, 51] and references therein) and design finite-time detection algorithms optimal for given objectives. A detailed discussion of these aspects can be found in the aforementioned references and are beyond the scope of this article.

7 CONCLUSION

In this article, we have addressed the problem of securing transportation cyber-physical systems from cyber attacks. We have considered two TCPSs that are of topical interest: an adaptive cruise control system and a system of self-driving vehicles. We have shown, using laboratory demonstrations, that the safety of these systems can be compromised by subverting sensors in the system and strategically misreporting their measurements. We have then implemented Dynamic Watermarking in these systems and have shown that it serves as a common defense strategy to effectively detect these attacks with low delay and thereby restore safety. We have also presented a framework using which finite-time statistical tests can be developed with certain desired false alarm rates and detection delays. Future work includes extending the theory of Dynamic Watermarking to more general nonlinear models, securing a system against malicious actuators, and demonstrating Dynamic Watermarking on full-fledged autonomous vehicles.

REFERENCES

[1] [n.d.]. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey.* Technical Report. A National Highway Transportation Safety Administration's report. Retrieved from https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115.

[2] [n.d.]. *Hackers Fool Tesla S's Autopilot to hide and Spoof Obstacles*. Wired magazine. Retrieved from https://www.wired.com/2016/08/hackers-fool-tesla-ss-autopilot-hide-spoof-obstacles/.

[3] [n.d.]. *Hackers Remotely Kill a Jeep On the Highway- With Me in it*. Wired magazine. Retrieved from https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[4] [n.d.]. *Secure Control of an Intelligent Transportation System*. Retrieved from https://youtu.be/xapmP2-u6HY.

[5] [n.d.]. *Securing a Prototypical Automated Vehicle Platoon from Replay Attacks Using Dynamic Watermarking*. Retrieved from https://youtu.be/996fg9hAfpw.

[6] [n.d.]. *Securing an Adaptive Cruise Control System from Adversarial Sensors Using Dynamic Watermarking*. Retrieved from https://youtu.be/n4dcaK7uGSo.

[7] [n.d.]. *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*. Society of Automobile Engineers standard. Retrieved from http://standards.sae.org/j3016_201401/.

[8] Marshall Abrams and Joe Weiss. 2008. Malicious Control System Cyber Security Attack Case Study—Maroochy Water Services, Australia. https://www.mitre.org/publications/technical-papers/malicious-control-system-cyber-security-attack-case-study-maroochy-water-services-australia.

[9] M. Amoozadeh, A. Raghuramu, C. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt. 2015. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Commun. Mag.* 53, 6 (June 2015), 126–132. DOI: https://doi.org/10.1109/MCOM.2015.7120028

[10] Tuomas Aura. 1997. Strategies against replay attacks. In *Proceedings of the 10th Computer Security Foundations Workshop*. IEEE, 59–68.

[11] Alvaro A. Cárdenas, Saurabh Amin, and Shankar Sastry. 2008. Research challenges for the security of control systems. In *Proceedings of the 3rd Conference on Hot Topics in Security (HOTSEC'08)*. USENIX Association, Berkeley, CA, Article 6, 6 pages. Retrieved from http://dl.acm.org/citation.cfm?id=1496671.1496677.

[12] Alvaro A. Cardenas, Saurabh Amin, and Shankar Sastry. 2008. Secure control: Towards survivable cyber-physical systems. In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*. IEEE.

[13] Soodeh Dadras, Ryan M. Gerdes, and Rajnikant Sharma. 2015. Vehicular platooning in an adversarial environment. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS'15)*. ACM, New York, NY, 167–178. DOI: https://doi.org/10.1145/2714576.2714619

[14] Bruce DeBruhl, Sean Weerakkody, Bruno Sinopoli, and Patrick Tague. 2015. Is your commute driving you crazy?: A study of misbehavior in vehicular platoons. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec'15)*. ACM, New York, NY, Article 22, 11 pages. DOI: https://doi.org/10.1145/2766498.2766505

[15] D. D. Dunn, S. A. Mitchell, I. Sajjad, R. M. Gerdes, R. Sharma, and M. Li. 2017. Regular: Attacker-induced traffic flow instability in a stream of semi-automated vehicles. In *Proceedings of the 47th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'17)*. 499–510. DOI: https://doi.org/10.1109/DSN.2017.61

[16] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. 2014. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Trans. Automat. Control* 59, 6 (2014), 1454–1467.

[17] J. P. Hubaux, S. Capkun, and Jun Luo. 2004. The security and privacy of smart vehicles. *IEEE Sec. Priv.* 2, 3 (May 2004), 49–55. DOI: https://doi.org/10.1109/MSP.2004.26

[18] Brent Kesler. [n.d.]. The vulnerability of nuclear facilities to cyber attack. http://large.stanford.edu/courses/2015/ph241/holloway1/docs/SI-v10-I1_Kesler.pdf.

[19] Woo-Hyun Ko, B. Satchidanandan, and P. R. Kumar. 2016. Theory and implementation of dynamic watermarking for cybersecurity of advanced transportation systems. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS'16)*. 416–420. DOI: https://doi.org/10.1109/CNS.2016.7860529

[20] P. R. Kumar and Pravin Varaiya. 1986. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice-Hall, Inc., Upper Saddle River, NJ.

[21] Tze Leung Lai and Ching Zong Wei. 1982. Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *The Annals of Statistics* (1982), 154–166. The Institute of Mathematical Statistics.

[22] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Sec. Priv.* 9, 3 (2011), 49–51.

[23] Gary Lorden et al. 1971. Procedures for reacting to a change in distribution. *Ann. Math. Stat.* 42, 6 (1971), 1897–1908.

[24] Sreekanth Malladi, Jim Alves-Foss, and Robert B. Heckendorn. 2002. *On Preventing Replay Attacks on Security Protocols*. Technical Report. Idaho University Moscow, Department of Computer Science.

[25] Fei Miao, Miroslav Pajic, and George J. Pappas. 2013. Stochastic game approach for replay attack detection. In *Proceedings of the IEEE 52nd Conference on Decision and Control (CDC'13)*. IEEE.

[26] Charlie Miller and Chris Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. Black Hat USA. http://illmatics.com/Remote%20Car%20Hacking.pdf.

[27] Shaunak Mishra, Yasser Shoukry, Nikhil Karamchandani, Suhas Diggavi, and Paulo Tabuada. 2015. Secure state estimation: Optimal guarantees against sensor attacks in the presence of noise. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT'15)*. IEEE, 2929–2933.

[28] Yilin Mo, Rohan Chabukswar, and Bruno Sinopoli. 2014. Detecting integrity attacks on SCADA systems. *IEEE Trans. Contr. Syst. Technol.* 22, 4 (2014), 1396–1407.

[29] Yilin Mo, Tiffany Hyun-Jin Kim, Kenneth Brancik, Dona Dickinson, Heejo Lee, Adrian Perrig, and Bruno Sinopoli. 2012. Cyber-physical security of a smart grid infrastructure. *Proc. IEEE* 100, 1 (2012), 195–209.

[30] Yilin Mo and B. Sinopoli. 2009. Secure control against replay attacks. In *Proceedings of the 47th Allerton Conference on Communication, Control, and Computing*. DOI: https://doi.org/10.1109/ALLERTON.2009.5394956

[31] Y. Mo, S. Weerakkody, and B. Sinopoli. 2015. Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Contr. Syst.* 35, 1 (Feb. 2015), 93–109. DOI: https://doi.org/10.1109/MCS.2014.2364724

[32] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. 2013. Attack detection and identification in cyber-physical systems. *IEEE Trans. Automat. Contr.* 58, 11 (2013), 2715–2729.

[33] J. Petit and S. E. Shladover. 2015. Potential cyberattacks on automated vehicles. *IEEE Trans. Intell. Transport. Syst.* 16, 2 (Apr. 2015), 546–556. DOI: https://doi.org/10.1109/TITS.2014.2342271

[34] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote attacks on automated vehicles sensors: Experiments on camera and LIDAR. In *Proceedings of the Black Hat Europe Conference*.

[35] Moshe Pollak. 1985. Optimal detection of a change in distribution. *Ann. Stat.* (1985), 206–227. The Institute of Mathematical Statistics.

[36] Udaya L. N. Puvvadi, Kevin Di Benedetto, Aditya Patil, Kyoung-Don Kang, and Youngjoon Park. 2015. Cost-effective security support in real-time video surveillance. *IEEE Trans. Industr. Inform.* 11, 6 (2015), 1457–1465.

[37] Craig L. Robinson, H.-J. Schutz, Girish Baliga, and P. R. Kumar. 2007. Architecture and algorithm for a laboratory vehicle collision avoidance system. In *Proceedings of the IEEE 22nd International Symposium on Intelligent Control*. IEEE, 23–28.

[38] B. Satchidanandan and P. R. Kumar. 2016. Secure control of networked cyber-physical systems. In *Proceedings of the IEEE 55th Conference on Decision and Control (CDC'16)*. 283–289. DOI: https://doi.org/10.1109/CDC.2016.7798283

[39] B. Satchidanandan and P. R. Kumar. 2017. Dynamic watermarking: Active defense of networked cyber-physical systems. *Proc. IEEE* 105, 2 (Feb. 2017), 219–240. DOI: https://doi.org/10.1109/JPROC.2016.2575064

[40] B. Satchidanandan and P. R. Kumar. 2017. On minimal tests of sensor veracity for dynamic watermarking-based defense of cyber-physical systems. In *Proceedings of the 9th International Conference on Communication Systems and Networks (COMSNETS'17)*. 23–30. DOI: https://doi.org/10.1109/COMSNETS.2017.7945354

[41] B. Satchidanandan and P. R. Kumar. 2017. The securable subspace of a linear stochastic system with malicious sensors and actuators. In *Proceedings of the 55th Allerton Conference on Communication, Control, and Computing*. 911–917. DOI: https://doi.org/10.1109/ALLERTON.2017.8262835

[42] Bharadwaj Satchidanandan and P. R. Kumar. 2018. *Control Systems Under Attack: The Securable and Unsecurable Subspaces of a Linear Stochastic System*. Springer International Publishing, Cham, 217–228. DOI: https://doi.org/10.1007/978-3-319-67068-3_16

[43] B. Satchidanandan and P. R. Kumar. 2018. On the operational significance of the securable subspace for partially observed linear stochastic systems. In *Proceedings of the IEEE Conference on Decision and Control (CDC'18)*. 2068–2073. DOI: https://doi.org/10.1109/CDC.2018.8619407

[44] B. Satchidanandan and P. R. Kumar. 2020. On the design of security-guaranteeing dynamic watermarks. *IEEE Contr. Syst. Lett.* 4, 2 (Apr. 2020), 307–312. DOI: https://doi.org/10.1109/LCSYS.2019.2925278

[45] Albert N. Shiryaev. 1963. On optimum methods in quickest detection problems. *Theor. Prob. Its Appl.* 8, 1 (1963), 22–46.

[46] Yasser Shoukry, Paul Martin, Yair Yona, Suhas Diggavi, and Mani Srivastava. 2015. PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*. ACM, New York, NY, 1004–1015. DOI: https://doi.org/10.1145/2810103.2813679

[47] Alexander G. Tartakovsky and Venugopal V. Veeravalli. 2005. General asymptotic Bayesian theory of quickest change detection. *Theor. Prob. Its Appl.* 49, 3 (2005), 458–497.

[48] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. 2012. Attack models and scenarios for networked control systems. In *Proceedings of the 1st International Conference on High Confidence Networked Systems*. ACM, 55–64.

[49] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. 2012. Revealing stealthy attacks in control systems. In *Proceedings of the 50th Allerton Conference on Communication, Control, and Computing*. 1806–1813. DOI: https://doi.org/10.1109/Allerton.2012.6483441

[50] Junia Valente and Alvaro A. Cárdenas. 2015. Using visual challenges to verify the integrity of security cameras. In *Proceedings of the 31st Computer Security Applications Conference (ACSAC'15)*. ACM, New York, NY, 141–150. DOI: https://doi.org/10.1145/2818000.2818045

[51] Venugopal V. Veeravalli and Taposh Banerjee. 2014. Quickest change detection. In *Academic Press Library in Signal Processing*, Vol. 3. Elsevier, 209–255.

Received December 2017; revised July 2019; accepted August 2019