A Recursive Watermark Method for Hard Real-Time Industrial Control System Cyber-Resilience Enhancement

Zhen Song, Antun Skuric, Kun Ji

A Recursive Watermark Method for Hard Real-Time Industrial Control Systems Cyber Resilience Enhancement

Zhen Song[1], Antun Skuric [2], Kun Ji [1]

***Abstract*—cybersecurity is of vital importance to Industrial Control Systems (ICSes), such as ship automation, manufacturing, building, and energy automation systems. Many control applications requires hard real-time channels, where the delay and jitter are in the levels of milliseconds or less. To our best knowledge, no encryption algorithm is fast enough for hard real-time channels of existing industrial fieldbuses, therefore made mission-critical applications vulnerable to cyber-attacks, e.g., delay and data injection attacks. In this paper, we propose a novel Recursive Watermark (RWM) algorithm for hard real-time control systems data integrity validation. Using a watermark key, a transmitter applies watermark noise to hard real-time signals and sent through the unencrypted hard real-time channel. The same key is transferred to the receiver via the encrypted non-real time channel. With the same key, the receiver can detect if the data has been modified by attackers and take action to prevent catastrophic damages. We provide analysis and methods to design proper watermark keys to ensure reliable attack detection. We use a ship propulsion control system for simulation-based case study, where our algorithm smoothly shuts down the system after attacks. We also evaluated the algorithm speed on a Siemens S7-1500 Programmable Logic Controller (PLC). This hardware experiment demonstrated that the RWM algorithm takes about 2.8μs to add or validate the watermark noise on one sample data point. As a comparison, common cryptic hashing algorithms can hardly process a small data set under 100ms. The proposed RWM is about 32 to 1375 times faster than the standard approaches.**

Note to Practitioners: ***Abstract*—It is widely believed that the emerging Internet of Things (IoT) technologies will seamlessly connect countless smart devices, profoundly change the industry. Traditionally, field devices within the feedback control loops are isolated from the Internet via secure gateways. In future, field devices will connect to the Internet in more direct manners. To our best knowledge, no encryption algorithm is fast enough for hard real-time channels of existing industrial fieldbuses, therefore made mission-critical applications vulnerable to cyber-attacks. We propose a novel Recursive Watermark (RWM) algorithm for hard real-time control systems data integrity validation. This paper serves industry practitioners in three manners. First, it is a timely caution to Industrial IoT (IIoT) pilot users on the security challenges in real-time channels. Once a compromised edge device is connected to a field device, attackers have unlimited means to jeopardize valuable assets. In this paper, we gave an example where attackers may damage shipboard assets by introducing millisecond-level delays. Second, since hard real-time encryption is not available, we propose a detour solution. With the proposed algorithm, even attackers may read the content in real-time channel, they cannot change it without being detected. We implemented the real-time RWM algorithms in Structured Control Language (SCL), and tested on a Siemens S7-1500 PLC. Third, we provide theoretical analysis as design guidelines for practitioners to setup or customize the RWM algorithm per their specific applications.**

***Index Terms*—Industrial control system, Internet of things, cyber security, cyber resilience, delay attack, watermark.**

NOMENCLATURE

- x: Vector. x: Scalar. X: Matrix. $f(\ )$: Function. $\mathcal{K}$: Set. s_w: Variable (lower letter subscript). T_S: A block in the system diagram (capital letters in the name and the subscript).
- $\mathcal{N}(\mu, \sigma^2)$: Normal distribution.
- A: Generic data integrity cyber attack event. A_I: Data injection attack event. A_R Replay attack event. A_D: Delay attack event. A is a set of all possible data integrity attacks, i.e. $A \in \{A_I, A_S, A_D, \cdots\}$. $\overline{A}$: The events without attacks.
- k_A: The time instance when the attack happens.
- $r[k]$: Reference signal (scalar or vector) at the k-th time instance.
- d: Discrete time delay, where d is an integer.
- $u[k]$: Control signal.
- $y[k]$: Output signal.
- $v[k]$: Sensor noise of normal distributions, where $v[k] \sim \mathcal{N}(0, \sigma_v^2)$.
- W: The sliding window length to calculate $g[k]$, $g_s[k]$ or $g_c[k]$.
- $w[k]$: Actuator noise of normal distributions, where $w[k] \sim \mathcal{N}(0, \sigma_w^2)$.
- $s[k]$: Raw sensor measurement. $s[k] = y[k] + v[k]$.
- $n_s[k]$: Watermark noise.
- $s_w[k]$: Watermarked sensor signal sent by the transmitter (T_S or T_C), where $s_w[k] = s[k] + n_s[k]$.
- $s_h[k]$: Potentially hacked signal at the receiver side.
- $s_r[k]$: The recovered sensor signal at the receiver side, i.e., if there is no attack, we should have $s_r[k] = s[k]$.
- $g[k]$, $g_s[k]$, $g_c[k]$: The generic detection function, the sensor channel detection function, and the control channel detection function, respectively, where k is the time instance; $g[k] \in \{g_s[k], g_c[k]\}$ and $g[k] \in \mathbb{R}$.
- $g_d[k]$, $g_{ds}[k]$, $g_{dc}[k]$: The Boolean detection variable for the generic channel, the sensor channel, and the control channel, respectively.
- τ: The threshold for the detection function, where $\tau > 0, \tau \in \mathbb{R}$.
- D: The bound of the second order derivative of the sensor or actuator noise.
- d: Discrete time delay introduced by attacks.
- $\mathcal{K}$: The watermark key $\mathcal{K}$ is a set of configurable parameters for hashing functions. In one example, $\mathcal{K} = \{\alpha_1, \alpha_2, w, n_p, b, m, a_{m-1}, \cdots, a_0\}$, where α_1 and α_2 are parameters for watermark noise generation; n_p is the RWM signal resetting duration; w is the window length for the watermark detector; $b, m, a_{m-1}, \cdots, a_0$ are configuration parameters for pseudo random number generation.
- $h_n(k; \mathcal{K})$, $h_n(k)$: A natural number cryptic hashing function to map a positive integer number k to a real number, i.e., $h_n(k; \mathcal{K}) \in$ [-0.5, 0.5], $k \in \mathbb{N}$, $y \in [0, 1], y \in \mathbb{R}$. Similar to h_r, we may denote it as h_n or $h_n(k)$ under the context when $\mathcal{K}$ is known.
- $h_r(x; \mathcal{K})$, $h_r(x)$: A real number cryptic hashing function to map a real number x to a real number, i.e., $h_r(x; \mathcal{K}) \in$ [-0.5, 0.5], and $x \in \mathbb{R}$. When there is no ambiguity, we use $h_r(x)$ for simplicity.
- $h_{rw}(k-1; \mathcal{K})$: A short form for $h_r(s_h[k-1]; \mathcal{K})$

I. INTRODUCTION

A. Motivation

[1] Siemens Corporation zhensong@ieee.org, ji.kun@Siemens.com. [2] INRIA Bordeaux, Sud-Ouest antun.skuric@inria.fr.

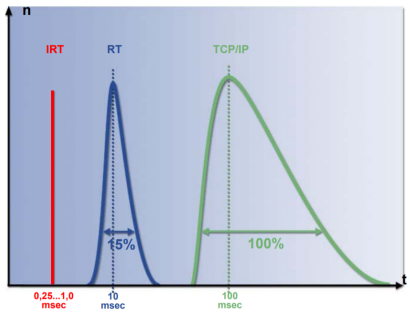


Fig. 1. Distribution of Profinet delays and jitters [7].

RECENTLY, cybersecurity issues of Industrial Control Systems (ICSes) has attracted much attention [1], [2]. An Industry Control System (ICS) is a set of industrial controllers and devices, such as Programmable Logic Control (PLC), Input/Output module (IO), Human Machine Interface (HMI), etc. Today, ICSes are facing different cybersecurity challenges [3] as comparing to traditional Information Technology (IT) systems. For example, data integrity attack is of vital importance to the security of an ICS, where attackers manipulate the communication data of the timing of the data to jeopardize high value targets. The Stuxnet worm issued a data injection attack [4], [5] on Programmable Logic Controllers (PLCs) and damaged over 1000 centrifuges of a nuclear facility. The Maroochy water breach [6] is another example, where attacker intruded the ICS network and released one million liters of untreated sewage water into a storm-water drain over the course of three months and eventually polluted the local waterways.

In this paper, we focus on the data integrity issues for the ICS hard real-time channels. To our best knowledge, no encryption algorithm or hardware is fast enough for hard real-time channels of existing industrial fieldbuses, therefore made some critical control applications vulnerable to different data integrity attacks, e.g., replay and data injection attacks. In this paper, hard real-time channels are those communication channels with delays and jitters in milliseconds or less, such as the Real Time (RT) and Isochronous Real Time (IRT) channels of the Profinet protocol [7]. The notion of jitter is the range between the maximal and minimal delays. Many industry fieldbus protocols, such as EtherCAT, CAN bus, Modbus, etc, support hard real-time communications with different technical solutions and different performances. These concepts are illustrated in Fig. I-A, where the delay of the standard TCP/UDP protocols is about 100 ms, the jitter range is about 100 ms or more. The Profinet RT channel has a stochastic delay of 10 ms and the jitter of 10 ms. The IRT channel of the Profinet is deterministic, therefore has the shortest communication jitters among the three and the delay is under 1 ms. Keep in mind that the definition of real-time, in other domains may have different definitions. For instance, delays and jitters for "real-time" video stream can be in the ranges of seconds to hundreds of milliseconds [8], respectively. Delays and jitters in this level may introduce unstable issues for some critical control systems.

A Profinet-based ICS secure network topology is shown on the left of Fig. 2, where the field level real-time communication among devices are through the unencrypted RT or IRT channels. The secure gateways can forward the real-time data within the RT and IRT channels to TCP/IP packets for long distance communications via the Internet. However, the real-time property is lost once the data is passed the security gateways. Although equipped with special encryption chips,

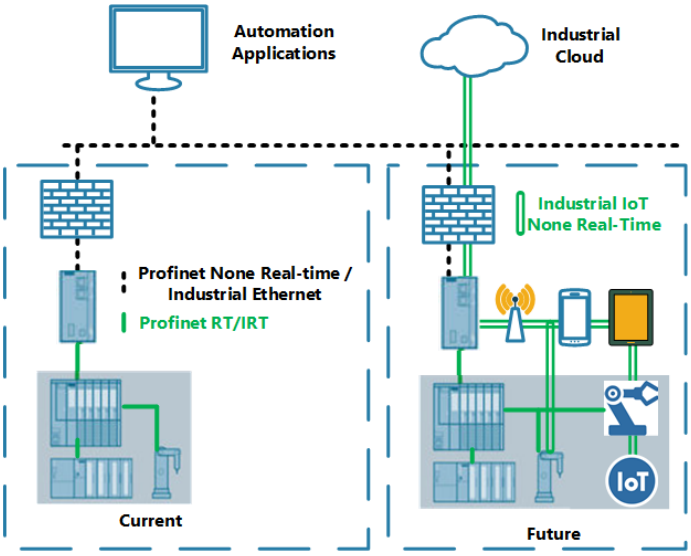


Fig. 2. Current and future ICS secure network topology.

the SCALANCE gateway cannot encrypt or decrypt data in the RT or IRT channels. This is not a serious issue in the past several decades, because there seemed to be not enough motivations to connect field devices to the Internet in the past. With the emerging IIoT technologies, there are unlimited new use cases to connect field devices to the Cloud, which expose the unencrypted real-time channels to potential future attacks. As shown on the right side of Fig. 2, operators may bring smart phones, tablets, mobile devices to the field and connect to PLCs or HMIs directly. Robots and Computer Numeric Control (CNC) machines may connect to the AI algorithms on the Cloud and continuously learn from each others. Smart sensors may be connected to the Cloud for big data analysis. If we do not get them carefully isolated, once a field device is compromised, hackers may read and write the unencrypted RT or IRT channels at will. Noticed that the network topology in Fig. 2 is applicable to generic industrial automation domains, including manufacturing, shipboard control system, energy automation, building automation, etc.

B. Potential Attacks on Shipboard Systems

Although proposed recursive watermark framework is applicable to ICS in general, we focus on the shipboard use case in this paper. Shipboard control networks are not only high value targets for cyberattacks, but also sensitive to small delays or jitters [9]. Future Navy ships [9] and civilian ships will be electrical driven [9]–[11], and involve agent-based distributed controls [12]–[14]. In our case study, we will demonstrate a case where millisecond level delays can unstable the ship generator control loop and potentially damage the propulsion motor or other critical equipment. If this attack happens, human operator cannot observe the abnormal signal, because the delay is too short. In shipboard systems, the dynamics fall within the range of milliseconds to seconds [15], [16], where small delays under human sensible level may have catastrophic damages. In order to illustrate the impacts of potential delay attack, we simulate the effects using a shipboard power control system model described in the Appendix A. Without delay attacks, the system step response is shown in Fig. 3, where the ω and i_{qs} are the angular speed and current of a shipboard electric propulsion motor, respectively. In our simulation-based study, we introduced small delays in the sensor signal channels of a shipboard system described in the Appendix A. As shown

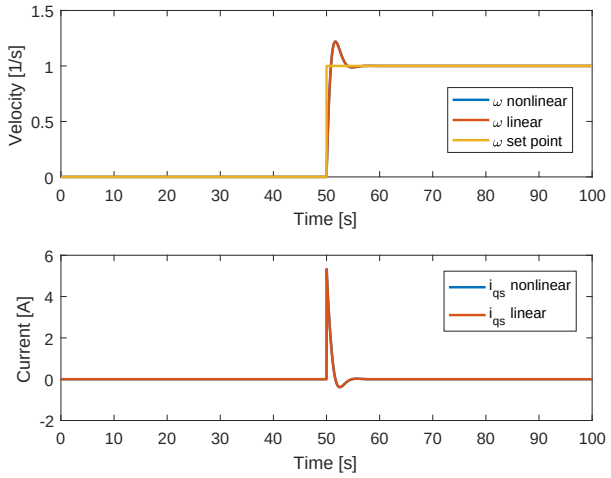


Fig. 3. Simulated shipboard response without delay.

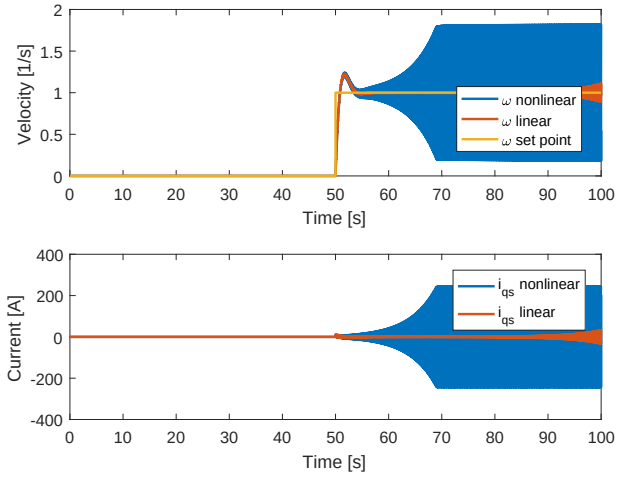


Fig. 4. Simulated shipboard response with $12.8ms$ delay of i_{qs}.

in Fig. 4, a short 12.8ms delay attack is sufficient to make the system unstable and potentially damage the propulsion motor.

C. Literature Review

Cyberattacks on ICSes can be classified in several groups [17], [18]:

- Denial of Service (DoS) attacks [19], [20].
- Replay and delay attacks [21]–[23].
- Deception attacks [24].
- Control [25] and sensor [1] signal data injection attacks.

Except the DoS attacks, we consider other attacks as data integrity attacks. In replay attacks [22], [23], the attackers record and replay commands to trigger dangerous actions. Deception attacker can manipulate sensors and other physical components of the system, or inject malicious program logic [24]. Data injection attacks are performed by adding false data to the information channel with purpose to impair the performance of the system [1], [2]. In this paper, we focus on data integrity attacks, because they raise unique challenges for many fieldbus-based ICSes. We don't distinguish *delay attacks* and *replay attacks*. Although the engineering implementation methods of the attacks are different, the underline mathematical attack models [23] are the same, which we will present in Sec.II-B. One approach to protect against delay attack is based on the system identification theory [26], where the receiver side can estimate the delay and trigger emergency actions to enhance control system resilience [27]. This model-based approach requires knowledge on the system model structure [28] and demands sufficient excitation signals [26] to identify parameters of non-linear systems. We consider it a passive method, because it does not require adding control signals to the control system. Another approach is active protection, where we can add special signals for better intrusion detection [29]. The watermark methods belong to this approach.

Traditionally, digital watermark methods are used for hiding information in digital contents, such as photos [30], audio [31], or videos [32], etc. The concept was introduced to control system domain recently [5], [27], [29], [33]. Although using the same "watermark" analogy, the underline mathematical formulations and objectives of the digital content watermarks (video, audio, image) and control system watermarks are significantly different. For example, it is desirable for a copyright watermark on a photo to be robust against photo editing, such that attackers cannot remove the copyright information. The watermark algorithms are often designed based on frequency domain analysis, and compared against robustness metrics, such as Peak Noise Signal Ratio (PNSR), Structural Similarity Measure (SSIM) etc. [34]–[36]. For control systems, the ideal watermark should be very *fragile*, such that any tiny adjustment will destroy the watermark. In control systems, copying the sensor signal is not the major concern. Instead, the major threat is modification on the signal, since it does not introduce catastrophic result. In this regard, the objective of control system watermark is similar to cryptic hashing methods [37], such as MD5 or SHA algorithms. However, *standard hashing methods do not consider the timing*, which is of vital importance to control systems. Another unique challenge for control system watermark is due to its nature to involve dynamic system physical models. From the engineering perspective, modeling physical systems are expansive and effort intensive. From the academia perspective, it is important to provide quantitative analysis, based on physics, to define the capability of the control system watermark methods.

To our best knowledge, existing control system watermark algorithms are all relies on physics models, mostly linear time invariant (LTI) system state space models. After adding different pseudo random Gaussian noises [27], [38] or psuedo random Bernoulli package dropout [39] on top of sensor or control noises, it is possible to detect the data integrity attacks with Kalman filter and/or delay estimation techniques [27], [38]. After all, the watermark noise or packet dropout are not completely random. To add even more barrier to the attackers and improve detection performance, some hybrid methods combined packet dropout and watermark noises [39] together. Similarly, other work merged several watermark signals together [19], with different offsets and multiplexed in time, rather than one statistical zero mean noise. Another method to challenge attackers is to add pseudo system dynamics in watermark signals [33]. Some work focus on cyber resilience enhancement, therefore the control system will try to operate after the attack by rejecting the attack signals as much as possible [27]. Also to our best knowledge, none of the existing work addressed the unencrypted hard real-time channels in industrial fieldbuses. We have not seen report on hardware experiments on mainstream industrial controllers. As a comparison, our proposed RWM method is more aligned with industrial practices. In order to avoid the intensive engineering efforts to build

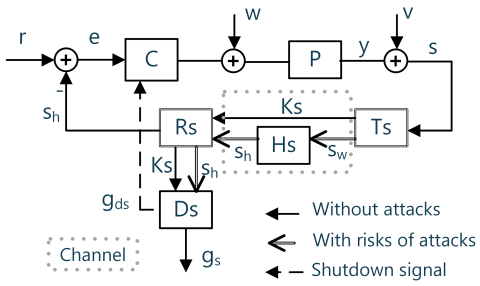


Fig. 5. Profinet ICS system diagram with RWM in the sensor channel.

accurate dynamic models for industrial non-linear systems (cf. Appendix A), we introduce internal dynamics through the recursive loop within the RWM algorithm. Since the internal dynamics is much faster than standard plant dynamics, it is safe to allocate the RWM algorithm to different positions in the control system without knowing detailed plant model. As Sec. II-G shows, as far as the max 2nd order derivative of the plant output is less than a threshold, our RWM method is applicable. We employed the gradient $|\Delta^2 v|$ as a metric for the fragile level.

The rest of the paper is organized as the followings. Sec. II introduces system block diagram and problem formulations. We then explain the watermark adding, removing, and detection algorithms. Sec. II-G provides detailed analysis on the detection algorithm. In Sec. III, we implemented the algorithms on Siemens PLC for performance testing. We also simulated a shipboard power system to validate the detection rate for common attacks. Finally, we conclude the paper in Sec. IV. For readers who want to repeat our simulation, we provide detailed shipboard system power system model in the Appendix A.

II. Problem Formulation and Solution

A. Control System Communication Block Diagram

In this section, we firstly describe the system with attack only in the sensor signal channel, as shown in Fig. 5. Secondly, we introduce attacks in both the sensor and control signal channels, which is plotted in Fig. 6. Since our RWM algorithm does not rely on physical system dynamics, we can easily deploy the RWM algorithm to any one channel or all the channels.

In Fig. 5, C is the controller; P is the plant; S is the sensors; H_S is the hacker; T_S is the transmitter in the sensor loop and R_S is the receiver in the sensor loop; D_S is the detector. The r, e, u, y, w, v are the reference signal, error signal, control system input, output, actuator and sensor noise, respectively. All these signals are time variant, e.g. e is $e[t]$, etc. The RWM key $\mathcal{K}_s$ is transferred from the sensor channel *transmitter*, T_S, to R_S via the encrypted non-real-time channel. Once R_S receives the key, T_S can send watermarked data, s_w, to the *receiver*, R_S, via the real-time channels, i.e., RT or IRT channels. Attackers may issue data integrity attacks on s_w within the H_S block and the output is s_h. Since the watermark noise is very small, R_S can feed the watermarked signal s_h directly the controller C. In order to check the data integrity, R_S also forwards $\mathcal{K}_s$ and s_h to the *detector* D_S to validate the data integrity, where some delay due the computation is not a concern, because it is out of the control loop. The float number g_s is the attack indicator and g_{ds} is the boolean variable to trigger shutdown process for the control C. If any data integrity attack is detected by D_S, the whole controller will shutdown smoothly.

Since the RWM algorithm is decoupled from the plant's dynamics, it is simple to embed another RWM block within the control signal channel, as shown in Fig. 6. The dot line box includes the control and sensor signal channels, where T_C, R_C, H_C, $\mathcal{K}_c$, D_C, g_c, g_{dc}, and w are the control channel transmitter, receiver, attacker, RWM key, detector, detector signal, detector decision signal, and actuator noise, respectively.

The name recursive is due to the unique *internal dynamics* design of the RWM algorithm, as shown in Fig. 7, where z^{-1} is the delay to introduce internal state and internal dynamics of the RWM algorithm; k is the discrete time instance; y, v, s_w are the system output, sensor noise, watermarked signal, respectively. The block in Fig. 7 is either the T_s or T_c in Fig. 5 or Fig. 6. The motivation for the internal dynamics is to make the RWM detector independent from the plant dynamics, therefore engineers can easily place the RWM algorithm in either the control channel or the sensor channel without the effort to accurately model the physics of the plant. It is a non-trivial task to model a real plant such as a ship power chain (as shown in the Appendix) or a manufacturing process, etc. While some works [5] employ Kalman filter to estimate the physical states in the plant, our proposed RWM detector estimates the internal state in the RWM algorithm. Since the sampling rate of the Profinet RT/IRT channels are very high, the induced delay of the RWM algorithm is short. Therefore, the internal state of the RWM algorithm changes much faster than typical physical variables in the plant. They are decoupled and placed at different positions on the system.

B. Attack Models

To issue data injection attacks, attackers must be able to read and write data. In generic data injection attacks, the attacker can inject any false data $u_a[k]$ to the channel.

Definition 1 (data injection attack A_I). Given k_1 and k_2 the beginning and ending discrete time instance of the attack; s_h is the received signal; s_w is the watermarked signal; the data injection attack function f_I is defined as:

$$s_h[k] = f_I(s_w[k]) = \begin{cases} u_a[k], & \text{if } k_1 < k < k_2 \\ s_w[k], & \text{else} \end{cases}$$

where $u_a[k]$ can be any signal defined by the attacker and $u_a[k] \neq s_w[k]$.

Now, we define shift attack, which is a subset of the data injection attack, i.e., $A_S \in A_I$. Instead of injecting random signal, the attacker can shift the original signal by a value $h_a[k]$.

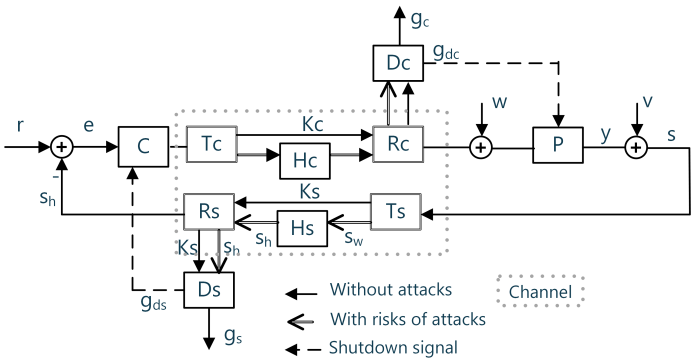


Fig. 6. Profinet ICS system diagram with RWM in channels.

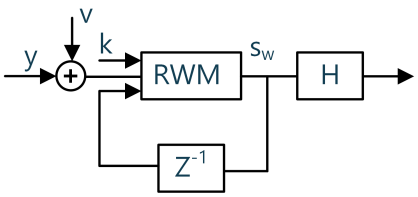


Fig. 7. RWM algorithm internal dynamics.

Definition 2 (data shift attack A_S). Given k_1, k_2, s_h, s_w in Definition 1, the data shift attack function f_S is defined as:

$$s_h[k] = f_S(s_w[k]) = \begin{cases} s_w[k] + h_a[k], & \text{if } k_1 < k < k_2 \\ s_w[k], & \text{else} \end{cases}$$

where $h_a[k]$ is a small real value defined by the attacker.

Even if the communication channel is encrypted, attacker can store the signal from a network device, such as a switch, and replay the same signal after certain delay. Some control systems are sensitive to even millisecond-level delays. The delay attack A_D is formulated as:

Definition 3 (delay attack A_D). Given k_1, k_2, s_h, s_w in Definition 1 and d is the delay, the delay attack function f_D is defined as

$$s_h[k] = f_D(s_w[k]) = \begin{cases} s_w[k-d], & \text{if } k_1 < k < k_2 \\ s_w[k], & \text{else} \end{cases}$$

The delay d can be either a constant or a function of time.

In this paper, we use $\overline{A}$ as the event without attack, i.e. when $s_h[k] = s_w[k]$. We use A to represent the attack event, where $A \in \{A_I, A_S, A_D, \cdots\}$. From the math definitions, A_I is the most generic set, i.e., $A_S \in A_I$, $A_D \in A_I$. Of course, the software implementation of different attacks are different. Therefore, we discuss them separately in verbal discussions, yet often aggregate the cases to one form during algorithm discussions.

C. Hashing Functions

In Appendix B, we present a short tutorial on an existing pseudo random integer sequence generation function $h(k)$, which generates integers between 0 and $b-1$.

Definition 4 (pseudo random number generator $h(k)$). Given set of configuration parameters $\{b, m, a_0, a_1, \cdots, a_{m-1}\} \in \mathbb{Z}$, $h(k)$ is a pseudo random number and $h(k) \in [0, b-1]$, where $k \in \mathbb{N}$.

Based on $h(k)$, we construct an integer hashing function $h_n(k)$ with normalized real output.

Definition 5 (integer hashing $h_n(k)$). Given a natural number k, h_n maps k to a real number between -0.5 and 0.5: $h_n(k) \in [-0.5, 0.5], k \in \mathbb{Z}$.

Similarly, we can construct a hashing function $h_r(x)$, which maps real number x to another pseudo random number between -0.5 and 0.5.

Definition 6 (real hashing $h_r(x)$). Given a real number x, h_r maps x to real number between -0.5 and 0.5: $h_r(x) \in [-0.5, 0.5], x \in \mathbb{R}$.

We can easily construct h_n and h_r functions based on the h function, so we don't present the details due to the limited space. In Appendix B, we provide a brief summary on the h function. The pseudo random number generation process cannot be easily reversed [40], and we found it fast enough for hard real-time control purposes. There are many potential choices for h_n and h_r functions, and they should meet the following criterion for the RWM algorithm.

- Not reversible: Ideal hashing function are cryptic functions, i.e, one cannot derive k from the value of $h_n(k)$.
- Fast: The forward mapping calculation must be fast enough to meet the real time performance requirement.

D. Recursive Watermark Generation Algorithm

While the concept of watermark for control system has been discussed before [21], the proposed RWM method introduced two or more channels: One non-real-time encrypted channel and one or more real-time yet unencrypted channel(s). As aforementioned, mainstream industry fieldbuses, such as Profinet, offer these multi-channel settings. The RWM generation algorithm resides at the sender side with two communication channels connected to the receiver side. Different from watermark approaches, which use one public channel only, the proposed RWM utilizes the encrypted channel to transfer the watermark *key* for the unencrypted channels. With this advantage, one novel feature of the proposed RWM method is the ability to identify the exact time of the attack event due to the new recursive watermarking algorithm, which will be presented in the following subsections.

The motivation of the watermark generation algorithm is to add the watermark noise on the original signal. With the same set of parameters that the sender used, i.e., the key, the receiver can easily validate the self-consistency. The attacker does not have the key, and cannot validate the self-consistency. If attacker modifies the watermarked signal at certain time, the self-consistence breaks since that time. Thanks to the recursive feature, any attack on the watermarked signal at one time instance will propagate to the following samples.

There are unlimited methods to define the recursive watermark function h_{rw}. For example, it can be a high order recursive function as $h_{rw} = 0.7s_w[k-1] + 0.2s_w[k-2] + 0.1s_w[k-3]$. In this paper, we only discuss 1st order RWM due to its simplicity, where $h_{rw} = h_r(s_w[k-1])$:

Definition 7 (1st order recursive watermark signal). Given the key $\mathcal{K}$ a set of constant parameters, the 1st order watermarked signal $s_w[k]$ is defined as:

$$s_w[k] := \begin{cases} s[k], \text{if } k \bmod n_p = 0 \\ s[k] + \alpha_1 h_n(k; \mathcal{K}) + \alpha_2 h_r(s_w[k-1]; \mathcal{K}), & \text{else} \\ \{\alpha_1, \alpha_2\} \geq 0. \end{cases}$$

For simplicity, it is also denoted as:

$$s_w[k] := s[k] + \alpha_1 h_n(k) + \alpha_2 h_{rw}(k-1), . \tag{1}$$

For presentation purpose, in Eq. 1, we refer $h_r(s_w[k-1])$ as $h_{rw}(k-1)$. Also, we refer the difference between $s_w[k]$ and $s[k]$ the watermark noise $n_s[k]$. For the 1st order RWM, we have $n_s[k] := \alpha_1 h_n(k) + \alpha_2 h_{rw}(k-1)$. The definition is illustrated in Fig. 8, where h_n and h_{rw} are reset to 0 after every n_p samples, when $s[k] = s_w[k]$.

For the engineering implementation, we have the choice to either calculate pseudo random numbers offline or online. The 1st method requires more memory yet demands less computations. The 2nd method is opposite. Hereby we define the 1st and 2nd method Algorithm 1 and Algorithm 2, respectively. In the algorithms, the mod function is modulus operation; The $\lfloor x \rfloor$ is the floor operator, i.e., the largest integer no larger than x.

Data: Input data is $\mathcal{K}$, $s[k]$, $s_w[k-1]$, s_{max}, s_{min}
Result: Output is $s_w[k]$
Offline: during the initialization phase
From $\mathcal{K}$, generate two lookup table T_1 and T_2, where T_1 has n_n elements and $T1[k] = h_n(k;\mathcal{K}), k \in [0, n_n]$; T_2 has n_r elements and $T2[k] = h_r(k;\mathcal{K}), k \in [0, n_p]$.
Online: Given $s[k]$
$n = k \bmod n_p$
if n = *0* **then**
  $s_w[k] = s[k]$
end
else
  $h_n(k) = T_1[n]$
  $i = \left\lfloor \dfrac{s_w[k-1] - s_{min}}{s_{max} - s_{min}} \right\rfloor$
  $h_{rw}(k-1) = T_2[i]$
  $s_w[k] = s[k] + \alpha_1 h_n[n] + \alpha_2 h_{rw}(k-1)$
end
return $s_w[k]$;
Algorithm 1: Offline Watermark Generation Algorithm

E. Watermark Removing Algorithm

Since the transmitter intentionally added noises of very small magnitude, it is acceptable to directly feed the watermarked signal to the actuator. However, if users do not want the small performance degradation due to the additional noise, there is an optional to remove the watermark, which can be conducted by the D_S block in Fig. 5. As shown in Fig. 8, $s_w[k] = s[k]$ and n_p is a periodic reset event, when the receiver can start the watermark removing process.

Data: Input data is $\mathcal{K}$, $s[k]$, $s_w[k-1]$,s_{max}, s_{min}
Result: Output is $s_w[k]$
Online: T_1, T_2 are arrays. Given $s[k]$
n = k mod n_p
if n = *0* **then**
  $s_w[k]$ = $s[k]$
end
else
  $T_1[k]$ = $h_n(k;\mathcal{K})$
  $h_n(k)$ = $T_1[k]$
  $i = \left\lfloor \dfrac{s_w[k-1] - s_{min}}{s_{max} - s_{min}} \right\rfloor$
  $T_2[i]$ = $h_r(i;\mathcal{K})$
  $h_{rw}(k-1)$ = $T_2[i]$
  $s_w[k] = s[k] + \alpha_1 h_n[n] + \alpha_2 h_{rw}(k-1)$
end
return $s_w[k]$
Algorithm 2: Online Watermark Generation Algorithm

Definition 8 (recovered signal for the 1st order recursive watermark signal).

$$s_r[k] = \begin{cases} s_h[k], \text{if } k \bmod n_p = 0\ , \\ s_h[k] - \alpha_1 h_n[k](n;\mathcal{K}) - \alpha_2 h_r\Big(s_h[k-1];\mathcal{K}\Big), \text{else.} \\ \alpha_1 + \alpha_2 = 1, \{\alpha_1, \alpha_2\} \in [0,1]. \end{cases}$$

The signals s_r and s_h are the recovered and received signal, respectively. The process is written in Algorithm 3.

F. Watermark Validation Algorithm

The receiver validates if the received signal has the proper watermark signal or not. If yes, the receiver believes there is no cyber attack. If not, the receiver will raise an alarm and trigger the shutdown process. In order to validate the watermark signal, the receiver needs to compare if the noise in the watermarked signal follows the distribution of the watermarked signal based on expected pseudo random noise. Since the sender already told the receiver key parameters, $\mathcal{K}$ of the watermark noise, the receiver can employ the χ^2 test to check if the received watermarked signal $s_w[k]$ follows the expected.

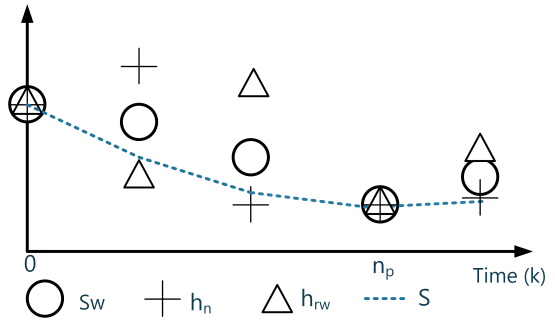


Fig. 8. RWM generation method.

Data: $\mathcal{K}$, $s_h[0], s_h[1], \cdots, s_h[k]$, time k
Result: $s_r[k]$
$n = k \bmod n_p$ **if** $n = 0$ **then**
  ▶ *It is the reset event*
  $s_r[k] = s_h[k]$
else
  ▶ *Not the reset event*
  $s_r[k] = s_h[k] - \alpha_1 h_n(n;\mathcal{K}) - \alpha_2 h_r(s_h[k-1];\mathcal{K})$
end
return $s_r[k]$
Algorithm 3: Watermark Removal Algorithm

The objective of the detection algorithm is illustrated in Fig. 9, where the x-axis is time and the y-axis is the detector signal $g[k]$ can be either $g_s[k]$ or $g_c[k]$ in Fig. 5 or Fig. 6. Before attack, i.e., $\bar{A}$, the detector signal $g[k]$ is smaller than that after the attack. Due to the unavoidable sensor or actuator noises, v or w, $g[k]$ is not completely 0 before the attack. With properly design watermark signal, we can setup a threshold τ, such that $\mathbb{E}\{g_k|\bar{A}\} < \tau < \mathbb{E}\{g_k|A\}$, where $\mathbb{E}\{g_k|\bar{A}\}$ and $\mathbb{E}\{g_k|A\}$ are the expectation of $g[k]$ without and with attacks, respectively. The scenario is shown in Fig. 9.

The receiver validates the received signal, $s_h[k]$ by using χ^2 test to check if the estimated noise $\hat{v}[k]$ is still subject to the known normal distribution. The χ^2 test is widely used for system malfunction analysis [41] and validation [19], [21].

$$e[k] = s_r[k] - \hat{s}_r[k|k-1]$$

where $e[k]$ is the forecast error based on $s_r[k-1]$; The vector

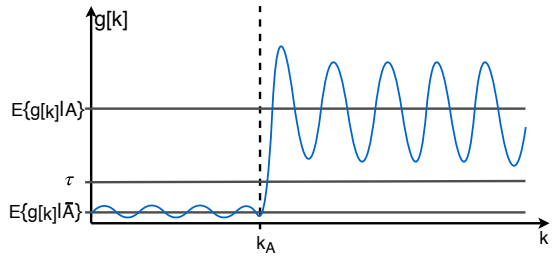


Fig. 9. Notations for χ^2 detector outputs.

$\hat{s}[k|k-1]$ is the forecast on $s_r[k]$

$$\begin{aligned}\hat{s}_r[k|k-1] &\approx s_r[k-1] + \Delta s_r[k-1] \\ &= 2s_r[k-1] - s_r[k-2] \\ e[k] &= s_r[k] - 2s_r[k-1] + s_r[k-2]\end{aligned} \tag{2}$$

In the detector D_S conducts a null hypothesis test assuming there is not attack, $\overline{A}$, therefore $s_r[k] = s_w[k]$ and $s_r[k] = s[k] = y[k] + v[k]$. Since $v[k]$ is subject to normal distribution $v[k] \sim \mathcal{N}(0, \sigma_v^2)$, we can apply the χ^2 test with W sample points:

$$g[k] := \frac{1}{W} \sum_{i=k-W+1}^{k} e[i]\sigma_v^{-1}e[i]. \tag{3}$$

We have $\mathbb{E}\{g_k\} = \lim_{W \to +\infty} g[k]$.

If $g[k]$ is not bigger than a threshold τ, we have good confidence that there is no attack. If $g[k]$ is too big, the detector will send the shut down signal to the controller. We have $g_d[k]$ the Boolean shut down signal defined as:

$$g_d[k] = \begin{cases} 1, g[k] \geq \tau \\ 0, g[k] < \tau \end{cases}$$

Data: $\mathcal{K}$, $s_r[0], s_r[1], \cdots, s_r[k]$
Result: $g[k], g_d[k]$
$g[k] = 0$
for $i = k - W + 1$, $i \leq k$ **do**
  $e[i] = s_r[i] - 2s_r[i-1] + s_r[i-2]$
  $g[i] = g[i] + e[i]\sigma_v^{-1}e[i]$ ▶ *Calculation χ^2 detector*
  $i = i + 1$
end
if $g[k] \geq \tau$ **then**
  $g_d[k] = 1$; ▶ *Shut down control*
else
  $g_d[k] = 0$; ▶ *Continue*
end

Algorithm 4: Watermark Validation Algorithm.

G. Analysis

We need to prove two fundamental properties for the sensor channel RWM algorithm: (1) Without attack and ignoring the sensor noise v, the original signal can be recovered, e.g., given the RWM key, $\mathcal{K}$, we can recover sensor signal s from the received sensor signal s_r. (2) When the absolute value of the second derivative of the sensor noise is no larger than a value D, it is possible to find a threshold τ to discriminate the attack and no attack event. Due to the limited space, we will not prove the similar properties for the control channel. We can easily replace sensor noise v by actuator noise w and the proof is immediate. Property 1 shows that the RWM algorithm has no information loss when there is no sensor noise. In engineering practices, system noises (sensor and actuator noises) are unavoidable, therefore property 2 gives a practical guideline on the capability of the RWM algorithm. As aforementioned, we intentionally introduce fast internal dynamics for the RWM algorithm to be decoupled from the physical system dynamics. The sensor noises, depends on hardware, may have high frequency component and overwhelm the watermark noise n_s. For engineering practices, we need to quantify metric to measure the noise and ensure it will not jeopardize the watermark noise, otherwise the $\mathbb{E}\{g_k|A\}$ and $\mathbb{E}\{g_k|\overline{A}\}$ in Fig. 9 are not significantly different.

1) Recover Sensor Signal:

Theorem 1. Given $s_w[k]$ defined in Definition 7 and $s_r[k]$ defined in Definition 8, when there is no attack, we can recover the sensor signal, i.e., $s_r[k] = s[k]$.

Proof. Under the no attack condition, i.e., $\overline{A}$, $s_h[k] = s_w[k]$. At the time $k = 0$, based on the definitions, it is trivial to have

$$s_r[0] = s_h[0] = s_w[0] = s[0].$$

When $k > 0, k < n_p$, since k mod $n_p \neq 0$, we have

$$\begin{aligned}s_h[1] &= s_w[1] = s[1] + \alpha_1 h_n(1;\mathcal{K}) + \alpha_2 h_r(s[0];\mathcal{K}) \\ s_r[1] &= s_h[1] - \alpha_1 h_n(1;\mathcal{K}) - \alpha_2 h_r(s[0];\mathcal{K}) \\ s_r[1] &= s[1] \\ &\vdots \\ s_r[k] &= s[k], k < n_p\end{aligned}$$

When $k = n_p$, the watermark signal is reset. Therefore,

$$\begin{aligned}s_r[n_p] &= s_h[n_p] = s_w[n_p] = s[n_p] \\ s_r[n_p] &= s[n_p].\end{aligned}$$

Similarly, it is easy to show that

$$\begin{aligned}s_r[n_p+1] &= s_h[n_p+1] - \alpha_1 h_n(1;\mathcal{K}) - \alpha_2 h_r(s_h[n_p];\mathcal{K}) \\ s_r[n_p+1] &= s_h[n_p+1] - \alpha_1 h_n(1;\mathcal{K}) - \alpha_2 h_r(s[n_p];\mathcal{K}) \\ s_r[n_p+1] &= s[n_p+1] + \alpha_1 h_n(1;\mathcal{K}) + \alpha_2 h_r(s[n_p];\mathcal{K}) \\ &\quad - \alpha_1 h_n(1;\mathcal{K}) - \alpha_2 h_r(s[n_p];\mathcal{K}) \\ s_r[n_p+1] &= s[n_p+1] \\ &\vdots \\ s_r[n_p+k] &= s[n_p+k],\ k < n_p\end{aligned}$$

In summary, when $\overline{A}$ we have $s_r[k] = s[r]$ for $k \geq 0$. □

2) Expected Value without Attack: We will derive the expectations of detection $\mathbb{E}\{g_k\}$ when there is no attack, then we compare those expectations with attacks. If they are significantly different, the detector can distinguish them give sufficient number of samples.

Theorem 2. Given sensor noise $v \sim \mathcal{N}(0, \sigma_v^2)$, D is the bound of the max absolute value of the second order derivative of sensor signal, i.e., $|\Delta^2 y| \leq D$, then the expectation of the χ^2 testing is g_k and $\mathbb{E}\{g_k|\overline{A}\} = 6\sigma_v$.

Proof. As proved in Theorem 1, without attack $s_r[k] = s[k]$ and $s[k] = y[k] + v[k]$ by definition, the watermarking residue, the forecast error can be written as:

$$\begin{aligned}e[k] &= y[k] - 2y[k-1] + y[k] + v[k] - 2v[k-1] + v[k-2] \\ e[k] &= \Delta^2 y[k] + \Delta^2 v[k].\end{aligned}$$

System noise par of the detector $\Delta^2 v[k]$ follows $\Delta^2 v[k] \sim \mathcal{N}(0, 6\sigma_v^2)$, where Δ^2 is the second order discrete derivation operator and is defined as:

$$\Delta^2 y[k] = y[k] - 2y[k-1] + y[k-2] = y''[k] \cdot T_s^2.$$

The T_s is a sampling time of the channel, and $y''[k]$ is the second order gradient for the continuous time system. When no attack scenario is considered, the expectation of the $g[k]$ follows:

$$\begin{aligned}\mathbb{E}\{g_k|\overline{A}\} &= \frac{1}{\sigma_v}\mathbb{E}\{e^2[k]\}, \\ &= \frac{1}{\sigma_v}\Big(\mathbb{E}\big\{\big[\Delta^2 y[k]\big]^2\big\} + \mathbb{E}\big\{\big[\Delta^2 v[k]\big]^2\big\}\Big),\end{aligned}$$

where signals y and v are independent. The expectation of the second derivation of the signal y naturally tends to be zero: $\mathbb{E}\{\Delta^2 y[k]\} = 0$.

With the assumption that the system noise v is white, different time instances of the noise $v[k], v[k-1], v[k-2]$ are independent as well and their cumulative expectations becomes:

$$\begin{aligned}\mathbb{E}\big\{\big[\Delta^2 v[k]\big]^2\big\} &= \mathbb{E}\big\{v^2[k]\big\} + 4\mathbb{E}\big\{v^2[k-1]\big\} + \mathbb{E}\big\{v^2[k-2]\big\} \\ &= 6\sigma_v^2\end{aligned}$$

Therefore the expectation of no attack χ^2 detector is:

$$\mathbb{E}\{g_k|\overline{A}\} = \frac{1}{\sigma_v}(6\sigma_v^2) = 6\sigma_v$$

□

Comments: This theorem quantifies the acceptable sensor noise for the RWM algorithm. While the magnitude of noise is important to robustness metrics for digital content watermarks [34]–[36], both frequency and magnitude are related to noise rejection capability of the RWM algorithm. Too much high frequency components in the sensor noise leads to larger $\Delta^2 v[k]$, which demands larger threshold for the χ^2 detector. Since $\Delta^2 v[k] \sim \mathcal{N}(0, 6\sigma_v^2)$, the value of the noise component $\Delta^2 v[k]$ of the detector g_k follows Gaussian normal distribution and its maximal value V can be determined by satisfying the equation:

$$Pr(\big|\Delta^2 v[k]\big| < V) = \Phi(V) - \Phi(-V) = \gamma$$

where $Pr()$ is the probability function of the Gaussian distribution and $\Phi()$ its cumulative distribution function. The desired probability is γ with a recommended value of 0.95. The scalar V is the maximum value of $\Delta^2 v[k]$, which can be calculated from sensor measurement for a given γ. Finally, with calculated V and known limit D the threshold value τ can be set as:

$$\tau \geq \max\big\{g_k|\overline{A}\big\} = \frac{1}{\sigma_v}(D+V)^2$$

3) Expected Value under Data Injection Attacks: If the attacker injects signal $u_a[k]$ as shown in Definition 1, we can estimate the expected value of the χ^2 testing as well.

Theorem 3. Under the data injection attack in Definition 1, the expected χ^2 testing result is

$$\mathbb{E}\{g_k|A_I\} = \frac{1}{\sigma_v}\mathbb{E}\{[\Delta^2 u_a]^2\} + \frac{1}{\sigma_v}\Big(6\mu_n + 6\mu_{rw}\Big),$$

where $\mu_n = \alpha_1^2\mathbb{E}\{h_n^2(k)\}, \mu_{rw} = \alpha_2^2\mathbb{E}\{h_{rw}^2(k)\}$.

Proof. Let's start from the simple case when $k < n_p$ and attack event happens before n_p. Based on Definition 8, and Definition 1 the recovered signal under A_I is:

$$\begin{aligned}s_h[k] &= u_a[k] \\ s_r[k] &= u_a[k] - \alpha_1 h_n(k) - \alpha_2 h_{rw}(u_a[k-1]) \\ e[k] &= \Delta^2 s_r[k]\end{aligned} \tag{4}$$

The expectation of the second derivation of the u_a is unknown because it is introduced from the attacker. On the other hand, h_n and h_{rw} have well known distributions and variances and their expectations can easily be calculated.

$$\begin{aligned}\mathbb{E}\{e^2[k]\} &= \mathbb{E}\{[\Delta^2 u_a[k]]^2\} \\ &\quad + \alpha_1^2\mathbb{E}\{[\Delta^2 h_n(k)]^2\} + \alpha_2^2\mathbb{E}\{[\Delta^2 h_{rw}(u_a[k-1])]^2\} \\ &= \mathbb{E}\{[\Delta^2 u_a[k]]^2\} + 6\mu_n + 6\mu_{rw},\end{aligned}$$

Therefore the expectation of the χ^2 detector during the data injection attack is:

$$\mathbb{E}\{g_k|A_I\} = \frac{1}{\sigma_v}\mathbb{E}\{[\Delta^2 u_a]^2\} + \frac{1}{\sigma_v}\Big(6\mu_n + 6\mu_{rw}\Big)$$

□

4) Expected Value under Delay Attacks: Due to the limited space, we study one delay attack where the delay is of one discrete time instance, $d = 1$. Other scenarios can be analyzed with the same technique.

Theorem 4. Given delay attacks in Definition 3 and the delay is 1,

$$\mathbb{E}\{g_k|A_D\} = \frac{1}{\sigma_v}\Big(6\sigma_v^2 + 20\mu_n\Big).$$

Proof. In the case of one sample delay attack, the received signal becomes:

$$\begin{aligned}s_h[k] &= s_w[k-1] \\ s_r[k] &= s_w[k-1] - \alpha_1 h_n(k) - \alpha_2 h_{rw}(s_w[k-2]) \\ e[k] &= s_r[k-1] - 2s_r[k-2] + s_r[k-3] \\ &\quad + \alpha_1\big(-h_n[k] + 3h_n[k-1] - 3h_n[k-2] + h_n[k-3]\big)\end{aligned}$$

Therefore the expectation of the watermarking residue $e[k]$ follows:

$$\begin{aligned}\mathbb{E}\{e^2[k]\} &= \mathbb{E}\{[\Delta^2 y[k-1]]^2\} + \mathbb{E}\{[\Delta^2 v[k-1]]^2\} \\ &\quad + \alpha_1^2\mathbb{E}\{[-h_r(k) + 3h_r(k-1) \\ &\quad - 3h_r(k-2) + h_r(k-3)]^2\}\end{aligned}$$

Since random noises h_n is independent in different discrete time samples k, and previously discussed $\mathbb{E}\{\Delta^2 y[k]\} = 0$, the χ^2 detector expectation follows:

$$\mathbb{E}\{g_k|A_D\} = \frac{1}{\sigma_v}\Big(6\sigma_v^2 + 20\mu_n\Big).$$

□

III. Experiments

A. Simulation Experiment

In the simulation experiment, we demonstrated the performance of the RWM algorithm based on an attack toward a sinusoid signal. Then we applied the algorithm to a shipboard power system data integrity validation. The detailed physics model of the shipboard system is described in Appendix A. Without protection, attacker may damage the generator using delay attacks. We used RWM algorithm to detect the attacks in time and shutdown the system smoothly.

Sinusoid Signal: Using simple sinusoid signals as examples, we compare the effects of watermark overlay, detection, and removing algorithms. On the top left of Fig. 10 and Fig. 11, there are comparisons between the raw sensor signal $s[k]$ and watermarked signal $s_w[k]$ without attack. The noise looks random, but it contains hidden information not observable to our eyes. The top right plots of Fig. 10 and Fig. 11 are the comparisons of the watermarked signal before and after delay attack and the data injection attack, respectively. The watermarked signal after the one sample delay looks very similar to the signal before the attack in Fig. 10. After the watermark being removed, the recovered signal shows abnormal yet small noise after the event of the attack, which is shown at the bottom left of Fig. 10. The χ^2 test identified the broken consistency of the after attack signal at the right bottom of Fig. 10 and Fig. 11 . We can easily identify the attack time with a threshold using the χ^2 test.

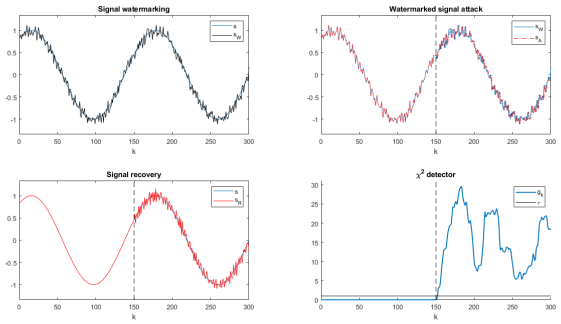


Fig. 10. Watermark authentication for 2 samples delay attack, window-size $w = 10$, variance $\nu = 1$, noise distribution uniform with magnitude 0.1.

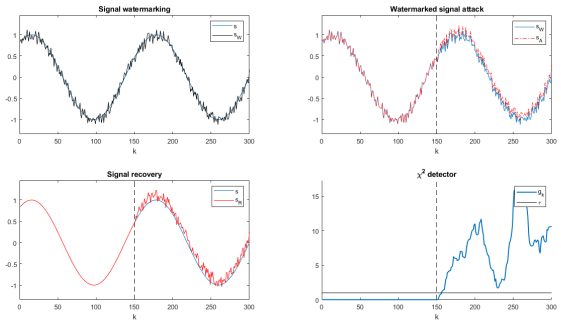


Fig. 11. Watermark authentication for data injection attack, window-size $w = 10$, variance $\nu = 1$, noise distribution uniform with magnitude 0.1.

Shipboard System: The Matlab Simulink model of the shipboard system is shown Fig. 12, where we assume the attacker can manipulate the sensor measurement on ω: the rotation speed, T_e: the torque, and λ_{dr}: the motor flux. When the system is working properly, the step response is shown in Fig. 3. As mentioned in Appendix. A, the system is sensitive to small delays not observable by humans. Under delay attack the system is unstable, as shown in Fig. 4. With the proposed RWM algorithm, we can quickly detect the delay and data injection attacks, as shown in Fig. 13. The χ^2 test within the watermark validator identified the attack immediately and give the system enough time to smoothly shutdown the motor and safely reduce the ω, as shown by the blue curves in Fig. 13. Without the RWM protection, the ω signal became unstable within 1 to 5 sec, which may damage the motor or other assets.

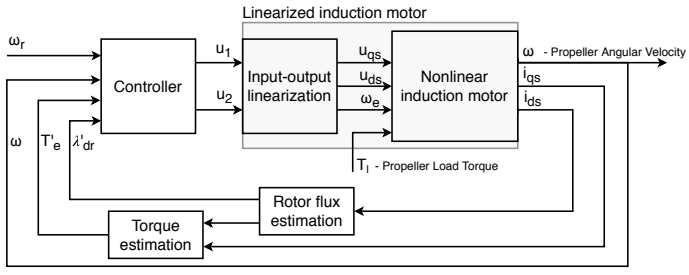


Fig. 12. Matlab simulation model of the shipboard control system.

B. Hardware Experiment

The key motivation of the RWM algorithm is to provide hard real-time data integrity checking on resource limited embedded devices. It is important to validate the speed of the algorithm on mainstream industrial control system, e.g., PLCs. From the RWM algorithm pseudo code in Sec. II-D, it is obvious that the complexity of the algorithm is rather low: Once the pseudo random numbers are generated, the complexity is $\mathcal{O}(n)$.

We evaluated the performance on Siemens S7-1500 PLC, as shown in Fig. 15. The S7-1500 PLC is a high-end PLC in the Simatic PLC product family and it is widely used in industrial automations, such as ship control, factory automation, energy automation, transportation systems, etc. We implemented the RWM algorithms in Structured Control Language (SCL) [42], which is one of the IEC 61131 standard PLC languages.

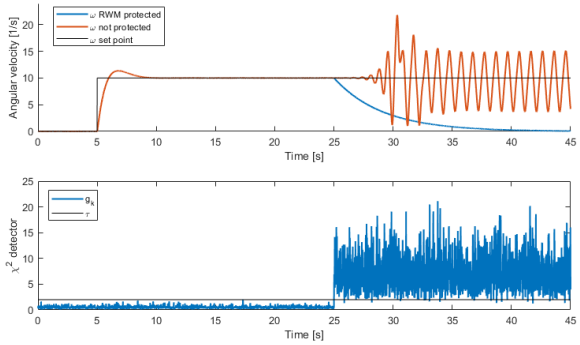


Fig. 13. Shipboard system response with delay attack at the 25-th second.

We implemented Alg. 1 and Alg. 2 and evaluated their performance on the S7-1500 PLC. Algorithm 1 is faster yet requires more memory. While Alg. 2 requires less memory but a bit more computation time. For both cases, the hashing table has 10^5 entries. We evaluated their performance in three categories.

1) Pseudo Random Number Generation: Both the offline and online watermark generation algorithms (Alg. 1, Alg. 2) are based on cryptic pseudo random number generations. Many industrial automation systems have limited random number generation features, e.g., the default random number generation function in SCL language can only output pseudo random numbers within a fixed range from -32768 to +32767. Since we need a fast algorithm with configurable range of the output, we implemented a matured pseudo random sequence generation algorithm [40] in SCL. More details of the algorithm is presented in Appendix. III-B. In Tab. I, we list the computation time to generate 10^5 random natural numbers that are no larger than $b^m - 1$. For better accuracy, we averaged the time values after about 100 experiments. The speed is fast enough for hard-real-time applications.

TABLE I
10^5 ELEMENT LOOKUP TABLE CREATION TIME IN MS.

Base b	5	5	5	5	2	5	31	53
Degree m	4	5	6	9	4	4	4	4
min[ms]	13	14	14	17	11	13	13	13
max[ms]	24	24	20	31	18	21	21	21
average[ms]	16	18	18	20	16	17	16	16

2) Watermark validation: The performance of χ^2 detector has been evaluated using different window sizes W. All the elapsed time values are expressed in ms per 10^3 iterations of the detector. The average values are calculated after 100 experiments.

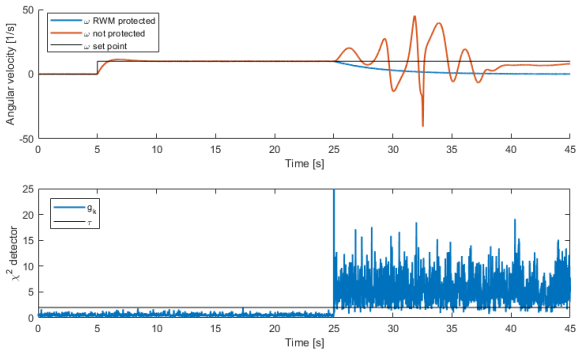


Fig. 14. Shipboard system response to inject attack at the 25-th second.

TABLE II
χ^2 DETECTOR ALGORITHM RESULTS IN MS PER 10^3 ITERATIONS

Window size W in time	5	10	20	50
min[ms]	2	5	11	28
max[ms]	9	13	22	42
average[ms]	5	9	14	32
avg per operation[μs]	5	9	14	32

3) Hashing algorithm: Hashing algorithm performance is compared for both hashing functions h_r and h_n. Both hashing function use lookup tables in memory and calculate the table index in relation to the hashing input. The results are expressed in ms per 10^6 iterations per experiment, and averaged after 100 experiments.

TABLE III
HASHING PERFORMANCE IN MS PER 10^6 ITERATIONSS

Hashing function	h_r	h_n
min[ms]	6	1
max[ms]	12	13
average[ms]	8	8

Now, we compare the performance of the RWM algorithm with two standard hashing algorithms, i.e., MD5 and SHA256. The experiment performed hashing operation for 10^5 float numbers. The experiments were conducted in two separate cases:

1) 10^5 consequent hashing operations.
2) 10^3 hashing operations for 100 bundled numbers

All the values are averaged after 5 experiments. The speed up ratio in Tab. IV is calculated by dividing the computation time of alternative methods (MD5, SHA256) by the time of the RWM algorithm.

TABLE IV
HASHING PERFORMANCE FOR DIFFERENT METHODS

Experiment		RWM	MD5	SHA256
Case 1	Time[ms]	16	12500	22000
	Per Num [μs]	0.16	125	220
	Speedup Ratio	-	781	1375
Case 2	Time[ms]	16	500	950
	Per Num [μs]	0.16	5	9.5
	Speedup Ratio	-	32	60

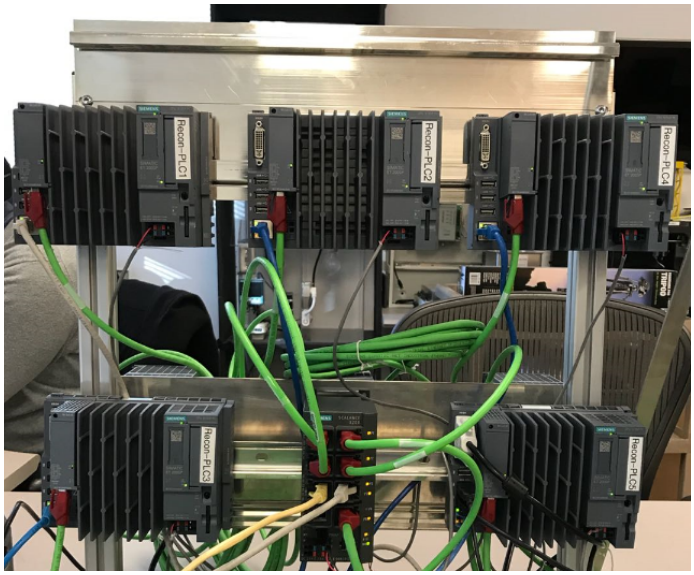


Fig. 15. Shipboard resilient control experimental system.

IV. CONCLUSIONS AND FUTURE WORKS

Today, there are no encryption or data integrity protections on the real-time channels in fieldbus protocols against data integrity attacks. As the emerging IIoT trend is connecting more field devices directly to the Internet or external networks, it is important to protect the real-time channels without compromising the control system performance. In This paper, we proposed a set of Recursive Watermark (RWM) algorithms for hard real-time data integrity validation on resource limited embedded systems to protect Profinet, a standard fieldbus protocol, from attacks at the real-time channels. Tested on a mainstream PLC, i.e., Siemens S7-1500, the algorithm is 32 to 1375 times faster than standard approach. In a shipboard power system case study, we demonstrated that the algorithm can detect fatal delay and data injection attacks. In order to limit the engineering efforts, the proposed algorithm is decoupled from the plant dynamics, and can be easily applied to different systems without system dynamics model. In future, we plan to investigate high order RWM systems.

ACKNOWLEDGMENT

The authors would like to present our appreciation to the supports from ONR, NRL and the CCIRS / RECON team. We are grateful for in-depth discussions with Dr. Arquimedes Canedo, Dr. Myong Kang, Mr. Jim Luo, Dr. Ryan Craven, Dr. Sukarno Mertoguno, Mr. Mike Veldink, Dr. Pranav Kumar, and Mr. András Varró.

APPENDIX A
SHIPBOARD POWER SYSTEM

For readers' convenience, we briefly present the simulation model used in this paper. The simulations in this paper is conducted based on a simplified all electric propulsion ship model [43]. We developed a Matlab Simulation model, i.e., Fig. 12 with parameters in Tab. V. In the system, there are two control variables, u_1 and u_2, and four outputs ω, λ_{dr}, i_{ds} and i_{qs}. Given the input u_1, T_e the electrical torque, and T_l the

load torque, the angular velocity ω of an induction motor [44] can be calculated from a decoupled [45] model:

$$T_e = \frac{\frac{3}{2}\frac{p}{2}\frac{L_M}{L_M+L_{lr}}}{sL_1+r_s}u_1, \quad (5)$$

$$L_1 = L_{ls} + \frac{L_M L_{lr}}{L_M + L_{lr}} \quad (6)$$

$$\omega = \frac{1}{sJ}\frac{p}{2}\Big(T_e - T_l\Big) = K_\omega \frac{1}{s}\Big(T_e - T_l\Big) \quad (7)$$

The rotor flux λ_{dr} is controlled by u_2, where

$$\lambda_{dr} = \frac{1}{s\frac{L_M+L_{ls}}{L_M} + \frac{r_s}{L_M}}u_2 \quad (8)$$

The stator current i_{ds} and i_{qs} are controlled by u_1 and u_2:

$$i_{ds} = \frac{\lambda_{dr}}{L_M} \quad (9)$$

$$i_{qs} = \frac{1}{\frac{3}{2}\frac{p}{2}\frac{L_M}{L_{lr}+L_M}}\frac{T_e}{\lambda_{dr}} \quad (10)$$

TABLE V
INDUCTION MOTOR PARAMETERS

Parameter	Symbol	Value	Unit
Rated power	P	400	hp
Pole pairs	p	4	-
Moment of inertia	J	4.1	kgm^2
Stator resistance	r_s	7.6	$m\Omega$
Rotor resistance	r_r	4.6	$m\Omega$
Stator inductance	L_{ls}	0.15	mH
Rotor inductance	L_{lr}	0.15	mH
Mutual inductance	L_M	5.2	mH

The currents i_{ds} and i_{qs} can be measured, but rotor flux λ_{dr} has to be estimated, e.g., using the following equation:

$$\hat{\lambda}_{dr} = \frac{L_M}{s\frac{L_{lr}}{r_r}+1}i_{ds} \quad (11)$$

Where $\hat{\lambda}_{dr}$ is estimated rotor flux.

This system is designed using cascade of PI controllers, with parameters shown in Tab. VI. Figures 16 and 17 the control system diagram, where circled values (ω,i_{qs}, i_{ds}) represent sensor measurement and values with ' sign are estimated states (T'_e, λ'_{dr}).

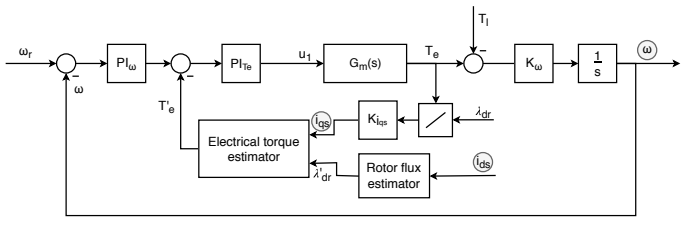


Fig. 16. Block diagram mechanical subsystem closed loop control.

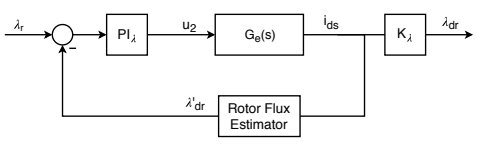


Fig. 17. Block diagram electrical subsystem closed loop control.

TABLE VI
PI CONTROLLERS PARAMETERS

Controller	P value	I value
PI_λ	1	0.1
PI_T	1	1
PI_ω	10	10

Table VI shows the parameters of the PI controllers used in this paper. PI_T and PI_ω are controllers in the torque T_e and shaft velocity ω cascades and PI_λ is controller in the flux loop.

APPENDIX B
PSEUDO RANDOM NUMBER GENERATION

As mentioned in Sec. III-B, the cyber-resilience of the RWM algorithm is based on a pseudo random number generator, which derive sequence of white noises that attackers cannot guess. Out of numerous candidates, we used a matured algorithm [40]. The algorithm can quickly generate white noises sequence of length n with autocorrelation functions of either 1 or $-1/n$. If $R[i]$ is the autocorrelation function at the i-th position, we have $R[0] = 1, R[i] = -1/n, i \in [1, n-1]$. We need to setup base b and degree m for the algorithm, where the sequence length n is defined by m with $n = b^m - 1$. The base b used by the mod-b operator is defined as:

$$x \oplus_b y = (x + y) \bmod b.$$

In this paper, we use $h(x)$ to generate pseudo random numbers for the proposed RWM algorithm.

$$h(x) = x^m \oplus_b a_{m-1}x^{m-1} \oplus_b \cdots \oplus_b a_1 x \oplus_b a_0,$$

where a_i are configurable parameters such that $\{a_{m-1}, \cdots, a_1, a_0\} \in [0, b-1]$.

REFERENCES

[1] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A. R. Sadeghi, M. Maniatakos, and R. Karri, "The cybersecurity landscape in industrial control systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, May 2016. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2015.2512235

[2] T. H. Morris and W. Gao, "Industrial control system cyber attacks," in *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research (ICS-CSR 2013)*, Leicester, UK, Sep. 2013, pp. 22–29. [Online]. Available: https://ewic.bcs.org/content/ConWebDoc/51165

[3] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *28th International Conference on Distributed Computing Systems Workshops (ICDCS 2008).* IEEE, Jun. 2008, pp. 495–500. [Online]. Available: http://dx.doi.org/10.1109/icdcs.workshops.2008.40

[4] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier," Security Response 5.6, Symantec Corp., Tech. Rep., 2011. [Online]. Available: https://preview.tinyurl.com/cpvzv2a

[5] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Systems*, vol. 35, no. 1, pp. 93–109, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1109/MCS.2014.2364724

[6] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *Critical Infrastructure Protection*, ser. IFIP International Federation for Information Processing, E. Goetz and S. Shenoi, Eds. Boston, MA: Springer US, 2007, vol. 253, ch. 6, pp. 73–82. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-75462-8_6

[7] Siemens, "PROFINET Real-Time communication," Dec. [Online]. Available: http://www.profibus.org.pl/index.php?option=com_docman&task=doc_view&gid=28

[8] F. Liu and H. Koenig, "A survey of video encryption algorithms," *Computers & Security*, vol. 29, no. 1, pp. 3–15, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404809000698

[9] T. V. Vu, D. Gonsoulin, F. Diaz, C. S. Edrington, and T. El-Mezyani, "Predictive control for energy management in ship power systems under high-power ramp rate loads," *IEEE Transactions on Energy Conversion*, vol. 32, no. 2, pp. 788–797, June 2017.

[10] S. Jothibasu and S. Santoso, "New electric shipboard topologies for high resiliency," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2975–2983, May 2018.

[11] R. A. Dougal, "Design tools for electric ship systems," in *Electric Ship Technologies Symposium, 2005 IEEE*, 2005, pp. 8–11. [Online]. Available: http://dx.doi.org/10.1109/ests.2005.1524645

[12] K. Huang, D. A. Cartes, and S. K. Srivastava, "A Multiagent-Based algorithm for ring-structured shipboard power system reconfiguration," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,*, vol. 37, no. 5, pp. 1016–1021, 2007. [Online]. Available: http://dx.doi.org/10.1109/tsmcc.2007.900643

[13] K. P. Logan, "Intelligent diagnostic requirements of future All-Electric ship integrated power system," *IEEE Transactions on Industry Applications*, vol. 43, no. 1, pp. 139–149, 2007. [Online]. Available: http://dx.doi.org/10.1109/tia.2006.886993

[14] *Multi-agent technology for self-healing shipboard power systems*, 2005. [Online]. Available: http://dx.doi.org/10.1109/isap.2005.1599264

[15] N. Doerry, "Naval power systems: Integrated power systems for the continuity of the electrical power supply." *IEEE Electrification Magazine*, vol. 3, no. 2, pp. 12–21, June 2015.

[16] N. Doerry and J. A. Jr., "Design considerations for a reference MVDC power system," in *SNAME Maritime Convention*, 2016.

[17] L. Zhang and H. Zhang, "A survey on security and privacy in emerging sensor networks: From viewpoint of close-loop," *Sensors (Basel, Switzerland)*, vol. 16, no. 4, pp. 443+, Apr. 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4850957/

[18] T. Morris and W. Gao, "Classifications of industrial control system cyber attacks," in *First International Symposium for ICS & SCADA Cyber Security Research 2013*, Leicester, UK, 9 2013.

[19] J. Rubio-Hernán, Cicco, and J. García-Alfaro, "Revisiting a Watermark-Based detection scheme to handle Cyber-Physical attacks," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Aug. 2016, pp. 21–28. [Online]. Available: http://dx.doi.org/10.1109/ARES.2016.2

[20] Z. Yang, P. Cheng, and J. Chen, "Learning-Based jamming attack against low-duty-cycle networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 650–663, Nov. 2017. [Online]. Available: http://dx.doi.org/10.1109/TDSC.2015.2501288

[21] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2009, pp. 911–918. [Online]. Available: http://dx.doi.org/10.1109/ALLERTON.2009.5394956

[22] C. Fang, Y. Qi, P. Cheng, and W. X. Zheng, "Cost-effective watermark based detector for replay attacks on cyber-physical systems," in *2017 11th Asian Control Conference (ASCC)*, Dec. 2017, pp. 940–945. [Online]. Available: http://dx.doi.org/10.1109/ASCC.2017.8287297

[23] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st International Conference on High Confidence Networked Systems*, ser. HiCoNS '12. New York, NY, USA: ACM, 2012, pp. 55–64. [Online]. Available: http://doi.acm.org/10.1145/2185505.2185515

[24] H. Zhang, P. Cheng, J. Wu, L. Shi, and J. Chen, "Online deception attack against remote state estimation," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 128–133, 2014, 19th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016416046

[25] Y. Mo and B. Sinopoli, "Integrity attacks on cyber-physical systems," in *Proceedings of the 1st International Conference on High Confidence Networked Systems*, ser. HiCoNS '12. New York, NY, USA: ACM, 2012, pp. 47–54. [Online]. Available: http://doi.acm.org/10.1145/2185505.2185514

[26] L. Ljung, *System identification (2nd ed.): theory for the user*, L. Ljung, Ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999. [Online]. Available: http://portal.acm.org/citation.cfm?id=293154

[27] A. Sargolzaei, K. K. Yen, M. N. Abdelghani, S. Sargolzaei, and B. Carbunar, "Resilient design of networked control systems under time delay switch attacks, application in smart grid," *IEEE Access*, vol. 5, pp. 15 901–15 912, 2017. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2017.2731780

[28] S. Bjorklund and L. Ljung, "A review of time-delay estimation techniques," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 3, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1109/CDC.2003.1272997

[29] P. R. K. B. Satchidanandan, "Dynamic watermarking: Active defense of networked cyber physical systems," *Proceedings of the IEEE*, vol. 105, no. 2, pp. 219–240, Feb. 2017. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2016.2575064

[30] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *INDIN '05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, Aug 2005, pp. 709–716.

[31] J. Bajpai and A. Kaur, "A literature survey - various audio watermarking techniques and their challenges," in *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Jan 2016, pp. 451–457.

[32] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2131–2153, Sep. 2018.

[33] J. Rubio-Hernan, L. De Cicco, and J. Garcia-Alfaro, "On the use of watermark-based schemes to detect cyber-physical attacks," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 8, Jun. 2017. [Online]. Available: https://doi.org/10.1186/s13635-017-0060-9

[34] A. M. A. R. R. Bala, "Bi-directional extreme learning machine for semi-blind watermarking of compressed images," *Journal of Information Security and Applications*, vol. 38, pp. 71–84, Feb. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214212617303162

[35] A. R. A. M. R. Bala, "A novel fuzzy frame selection based watermarking scheme for MPEG-4 videos using bi-directional extreme learning machine," *AppliedSoftComputingJournal*, pp. 603–620.

[36] R. G. A. M. S. Jain;, "A semi-blind HVS based image watermarking scheme using elliptic curve cryptography," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19 235–19 260, Aug. 2018. [Online]. Available: https://doi.org/10.1007/s11042-017-5351-0

[37] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, "Cryptographic hash functions: A survey," Tech. Rep., 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.8428

[38] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on SCADA systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1109/TCST.2013.2280899

[39] S. Weerakkody, O. Ozel, and B. Sinopoli, "A bernoulli-gaussian physical watermark for detecting integrity attacks in control systems," in *Proc. and Computing (Allerton) 2017 55th Annual Allerton Conf. Communication, Control*, Oct. 2017, pp. 966–973.

[40] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proceedings of the IEEE*, vol. 64, no. 12, pp. 1715–1729, Dec. 1976. [Online]. Available: http://dx.doi.org/10.1109/PROC.1976.10411

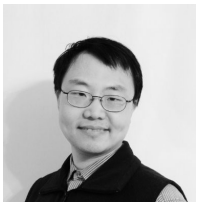
[41] R. K. Mehra and J. Peschon, "Correspondence item: An innovativeapproach to fault detection and diagnosis in dynamic systems," *Automatica*, vol. 7, no. 5, pp. 637–640, Sep. 1971. [Online]. Available: http://dx.doi.org/10.1016/0005-1098(71)90028-8

[42] Siemens, "Programming guideline for S7-1200/S7-1500," Tech. Rep. [Online]. Available: https://preview.tinyurl.com/http-www1-siemens-cz-ad-curr

[43] B. Zahedi and L. E. Norum, "Modeling and simulation of all-electric ships with low-voltage dc hybrid power systems," *IEEE Transactions on Power Electronics*, vol. 28, no. 10, pp. 4525–4537, 2013.

[44] P. C. Krause and C. Thomas, "Simulation of symmetrical induction machinery," *IEEE transactions on power apparatus and systems*, vol. 84, no. 11, pp. 1038–1053, 1965.

[45] G.-S. Kim, I.-J. Ha, and M.-S. Ko, "Control of induction motors for both high dynamic performance and high power efficiency," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 4, pp. 323–333, 1992.



Zhen Song Dr. Zhen Song is senior IEEE member. He is a senior data scientist at Siemens Building Technologies. From 2006 to 2018, he worked in Siemens Corporate Technology, Princeton, New Jersey, where he has been the principal investigator of government research projects sponsored by DOD, DOE, NRL, ONR and New York State. He developed the sensor system of ODIS robot, which has been deployed to Iraq, Afghanistan to protect US soldiers from car bombs. In 2007 he received Ph.D. degree in Electrical and Computer Engineering from Utah State University.

Antun Skuric Mr. Antun Skuric received his B.S.E.E. (2014), and an M.S.E.E. (2017) from Faculty of Electrical Engineering and Computing, University of Zagreb. Currently, he is working as a Research Associate for Mechatronics Laboratory of the Department of Electric Machines, Drives and Automation (ZESA) at the same university. He is co-founder of a startup GuitarFriend. His main research interest is optimal control applications on mechatronic systems.

Kun Ji Dr. Kun Ji received the Ph.D degree in Mechanical Engineering from Texas A&M University, College Station, in 2006. Since then, he has been with Siemens Corporate Technology, Princeton, New Jersey. His research interests focus on resilient control and distributed sensor networks, grid automation and industrial automation systems, building control and energy efficiency, and cyber-physical systems. He is an IEEE member.