Energy-Efficient Intrusion Detection and Mitigation for Networked Control Systems Security

Riccardo Muradore, *Member, IEEE*, and Davide Quaglia, *Member, IEEE*

***Abstract*—This paper proposes an energy-efficient security-aware architecture for wireless control systems to be used in factory automation. We face deception attacks that corrupt commands and measurements in a smart way and with intermittent behavior to produce the highest damage without being discovered. The intrusion is hard to distinguish from normal disturbance. Furthermore, protection against attacks is energy-consuming and it would be desirable to activate protection only when needed. We propose packet-based selective encryption to reduce energy consumption, and to detect when an attack starts and ends. Since energy consumption depends also on packet transmission rate, especially during attacks, we also propose to adapt it according to instantaneous control performance.**

***Index Terms*—Deception attack, digital signature, encryption, energy-efficiency, networked control system (NCS), security, wireless transmission.**

I. Introduction

SECURITY aspects in factory automation have become a hot topic in the last years since monitoring and control tasks are more and more complex. Such systems often employ distributed networks of embedded sensors and actuators that interact with the physical plant, and are monitored and controlled by a supervisory control and data acquisition (SCADA) system [25]. The communication through packet-based networks among different subsystems is necessary but, at the same time, risky in terms of confidentiality and data integrity [3], [8], [10], [27].

Security attacks could bring to severe damages especially where networked control systems (NCSs) are used to operate in dangerous environment (e.g., chemical plant) or in critical scenarios (e.g., teleoperation). Fig. 1 shows the basic block diagram of an NCS where the continuous-time plant $P(s)$ and the digital controller $C(z)$ are connected through a packet-based network. The plant sends packets containing the output y, whereas the controller sends packets containing the commands u to the plant aiming at keeping y as close as possible to the reference r. In this work, we consider *deception attacks* which affect the data integrity of packets by modifying their payload. In particular, we assume that an intermediate system of the network is tampered, so that it relays corrupted packets. The attacker can affect either command packets u (using d_{C2P}) or measurement packets y (using d_{P2C}) or both.

In general, NCSs present many challenges due to the time-varying delays and packet dropouts. This work does not focus on them and we assume stability for granted. We study methodologies to detect an attack and to mitigate its effect on the NCS from the point of view of both performance and damage. Clearly, there is a tradeoff between security and performance [19], [40] and the proposed approach can be combined with such literature to find an optimal configuration.

Usual techniques to protect packets' integrity are based on *digital signature*, which appends an encrypted summary of the message to the message itself. If the attacker corrupts such a message, its presence is revealed. Digital signature increases energy consumption mainly due to the increased size of the transmitted packet. This could be a problem in case of battery-powered wireless devices which are gaining interest in factory automation [1], [9], [16], [35], [38].

Traditionally, energy optimization focuses on the digital part of the system and on the executed software; well-known energy-saving techniques can be either hardware (HW)-based [33] such as clock-gating, voltage, and frequency scaling, or SW-based [7]. In the context of networked embedded systems, it is traditionally known that communications play a significant role in energy consumption [12] and, for this reason, energy-efficient transmission strategies have been designed recently [18], [37].

While energy overhead can be tolerated during an attack, it represents a waste of resources when the attack is not active. Therefore, the most important issue to optimize system resources is *intrusion detection*. Traditional anomaly-based intrusion detection systems (IDSs) monitor network traffic and compare it against an established baseline [13]. The baseline will identify what is "normal" for that network, what sort of bandwidth is generally used, what protocols are used, and what ports and devices generally connect to each other. Even if applied to control applications [39], [41], traditional approaches look for "formal" or "network-oriented" anomalies and do not analyze the content of packets from the point of view of a control application. For example, altered commands transported by a formally correct protocol are not detected by a traditional IDS. In the context of control systems, some attacks have been designed to be virtually undetectable [24]. Past literature shows that intrusion detection is an open problem [17], [25], [26]. Furthermore, in a simple example at the beginning of the paper, we will show that packet deception cannot be detected simply by looking at the control performance since in many cases, injected data are not distinguishable

Manuscript received April 14, 2014; revised August 22, 2014, December 24, 2014 and March 02, 2015; accepted April 07, 2015. Date of publication April 21, 2015; date of current version June 02, 2015. Paper no. TII-14-0450.

The authors are with the Department of Computer Science, University of Verona, Verona 37134, Italy (e-mail: riccardo.muradore@univr.it; davide.quaglia@univr.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2015.2425142

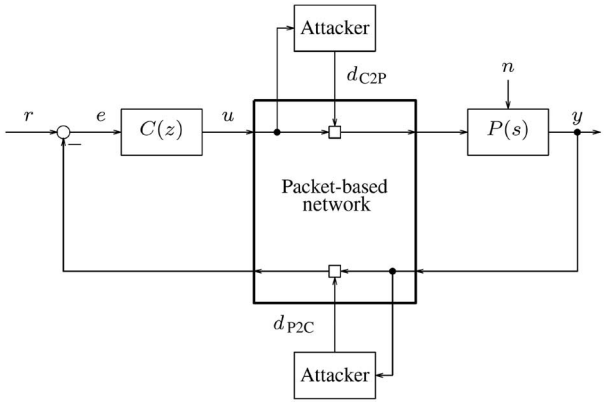

Fig. 1. Block diagram of an NCS for factory automation under deception attack.

from normal perturbations of the physical plant. The proposed architecture does not rely on a particular detection mechanism but rather it aims at detecting the begin and end of the attack and reacting against it. In particular, we propose the *selective encryption* of the packets exchanged between controller and plant, and we present an attack-detection methodology based on the comparison between encrypted and unencrypted commands.

Selective encryption was used to guarantee different levels of smart meter privacy [11] and to reduce energy consumption in wireless communications, e.g., for the transmission of voice [14] and ECG data [22].

Another issue is *attack mitigation*, i.e., the countermeasure to be adopted when an attack is detected. From one side, this technique should eliminate damage risks and performance loss; however, from the other side, it should preserve the possibility to detect that the attack is over, so that resource consumption can be reduced. Attack mitigation has been addressed in the context of wireless transmission [30] and smart grid applications [34]. In this work, we propose to encrypt all the packets of the flow under attack except some anchor packets to detect when attack is over.

Recent work on the impact of packet losses on control performance shows that not all packets (i.e., commands and output measurements) are equally important [2], [32]. This finding suggests to further improve energy efficient by *varying the packet transmission rate* according to the control performance.

All these mechanisms need an extended architecture, which is also presented in this paper. The components of this architecture are suitable to be embedded in smart devices by following guidelines in literature [23].

This paper is organized as follows. Section II provides a motivation and a definition of the problem. Section III presents the proposed architecture for energy-efficient intrusion detection and mitigation. Simulation results are reported in Section IV, and conclusion is drawn in Section V.

II. Problem Description

A. Wireless Control Systems

Fig. 1 shows the basic block diagram of an NCS where the continuous-time plant $P(s)$ and the digital controller $C(z)$ are connected through a packet-based wireless network. Both controller and plant are represented using a linear model whose transfer functions are given by $C(z)$ and $P(s)$, where z is the Z-transform variable and s is the Laplace variable, respectively. The plant sends packets containing the output y, whereas the controller sends packets containing the commands u to the plant aiming at keeping y as close as possible to the reference r.

Wireless networks are spreading in the context of machine-to-machine communication (e.g., for factory automation) since they can be easily deployed, without additional cost for wiring, and extended to introduce new controllers, sensors, and actuators. Furthermore, in some mobile or harsh environments, wireless communications are the only solution. There are various wireless protocols in literature, with both deterministic and statistical latency [1]. Wireless medium may introduce transmission issues, e.g., delay and packet losses, which can affect control performance. In literature, various techniques have been proposed to address these problems [15]. When wires are not present, energy should be supplied through batteries or harvested from the environment: in both cases, energy efficiency becomes a strong requirement to guarantee long device lifetime without human intervention [6].

The main cause of energy consumption is transmission; for instance, the well-known Texas Instruments CC2530 SoC[1] consume about 30 mA to transmit and about 6.5 mA to perform computation [36]. The evolution of processors will progressively reduce the energy consumption for computation, but the energy consumption for transmission cannot be easily reduced since it strongly depends on application requirements (e.g., transmission range). Therefore, *the proposed security approach aims at minimizing the transmission overhead* due to encryption. Energy can also be saved by reducing the transmission rate of commands and output measurements when the control performance are above a desired threshold.

B. Cyber-Attacks and Protection

Wireless networks are particularly prone to security attacks since the attacker does not need to tamper the wire to listen communications. In this work, we are interested in message corruption, as it can lead to severe damage of the NCS. Therefore, we assume a "man-in-the-middle" attack approach, which changes the messages in a tampered intermediate node according to the attacker strategy.

Attack countermeasures are based on several encryption methods, classified into symmetric, e.g., Advanced Encryption Standard (AES), and asymmetric (e.g., RSA). To assess message integrity, digital signature is used. In this scheme, the message signature is generated by the sender by encrypting a short digest of the message using sender's private key; digest is created using a hash function known also at receiver side; the signature is transmitted together with the message; the receiver decrypts the signature with sender's public key and compares the result with a locally computed digest; if they are equal, the message integrity is verified. If the content of the message is changed during transmission, it does not correspond with the

[1] SoC stands for system on chip.

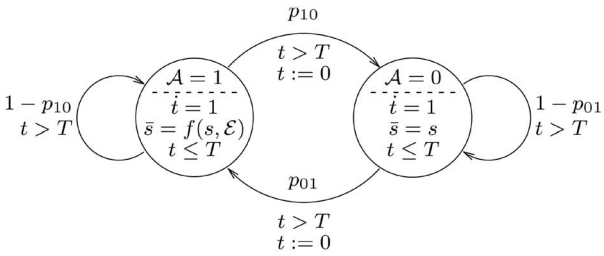

Fig. 2. Model of the attacker.

signature and therefore, the receiver detects the attack. The basic assumption is that it should be computationally infeasible to generate a valid signature for a party without knowing party's private key. When symmetric key is used, the signature is named Message Authentication Code. To detect also replay attacks, a counter can be inserted in the signed message. In this work, we assume the presence of an end-to-end security protocol. In other words, packet signing and integrity check are performed at controller and plant side while intermediate network devices just relay packets. In this way, a man-in-the-middle attack on a tampered network device cannot modify signed data without being discovered. If the attacker knows the transmission protocol, it is able to understand whether a packet contains a message signature. Therefore, without loss of generality, in this work, we just define a signed message as "encrypted" ($\mathcal{E} = 1$) and we assume that the attacker should not alter it to stay hidden.

Assumptions on the attacker strategy are very important to design an effective security solution. Traditionally, the attacker of controlled systems has three main objectives: 1) damage the system under control; 2) reduce control performance; and 3) remain undiscovered for a long time. To achieve these objectives, in this work, we assume that the attacker:

1) alters signal from controller to plant and plant to controller to damage the plant and to reduce control performance;
2) keeps untouched the encrypted messages to avoid being discovered.

A possible formal model of the attacker is obtained using a stochastic (i.e., Markov chain) hybrid system as shown in Fig. 2. The locations refer to the attack ($\mathcal{A} = 1$) or not-attack ($\mathcal{A} = 0$) status; p_{ij} is the transition probability from state i to state j that may occur when the condition $t > T$ is satisfied. T is the minimum time duration of both an attack and the interval between two attacks. The expression $\bar{s} = f(s, \mathcal{E})$ means that the packets containing the signal s are corrupted only if unencrypted ($\mathcal{E} = 0$). The signal s could be the command u or the measurement y.

Packet encryption leads to more energy consumption. For instance, AES with 192-bit key and 100-byte packets leads to about 8% more energy consumption on a wireless device using IEEE 802.11 protocol [29]. Assuming the use of IEEE 802.15.4 protocol standard with a maximum packet size of 133 bytes, AES-based signature is 16 bytes [5], thus leading to an energy overhead of about 12%; in case of smaller packets, such overhead would be even higher. Energy overhead is due to the higher use of computational and communication resources.

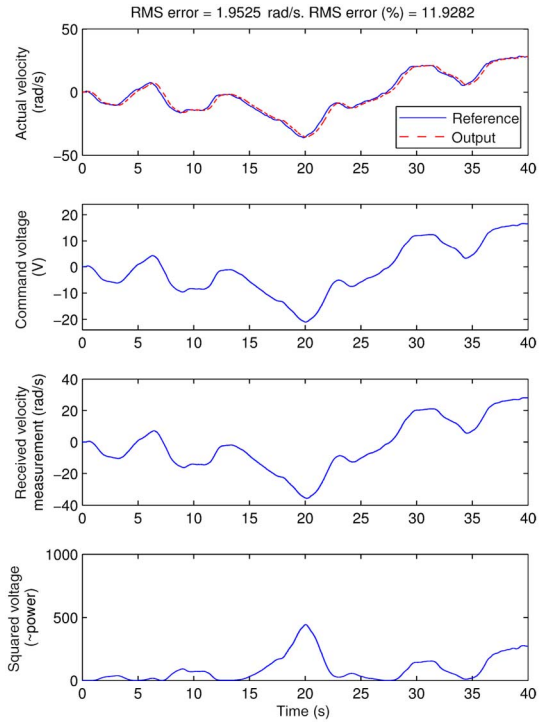

Fig. 3. Behavior of an NCS without attack. From top to bottom: control performance; applied commands; received measurements; and energy of the applied commands.

In our work, we decided to focus only on the latter aspect. The presence of a signature increases the size of the packet, which requires more energy to be transmitted. This contribution to energy overhead is independent of CPU power, hardware architecture, transmission standard, and encryption standard. Furthermore, the evolution of embedded systems and cryptography will probably decrease the computational overhead while the transmission overhead will not change easily being more dependent on application requirements (e.g., transmission range).

Since the protection of all packets consumes a huge amount of energy, two objectives should be achieved.

1) The protection should be activated only when the attack is on-going.
2) The transmission rate should be reduced when instantaneous control performance is better than desired.

Unfortunately, attack detection is a hard task as shown in the next example.

C. Motivating Example

It is worth noting that, in general, attack detection may be difficult if performed by analyzing the control performance. In the next example, we analyze the behavior of the NCS used in our experiments in which the plant is a first-order system [see (19)] and the controller is proportional-integral.

Fig. 3 reports the behavior of the NCS when no attack is performed. The first plot compares reference and output signals; the controller has been designed so that the rms of the tracking error is around 2 ± 1 rad/s in nominal condition. The second and third plots show the behavior of the commands

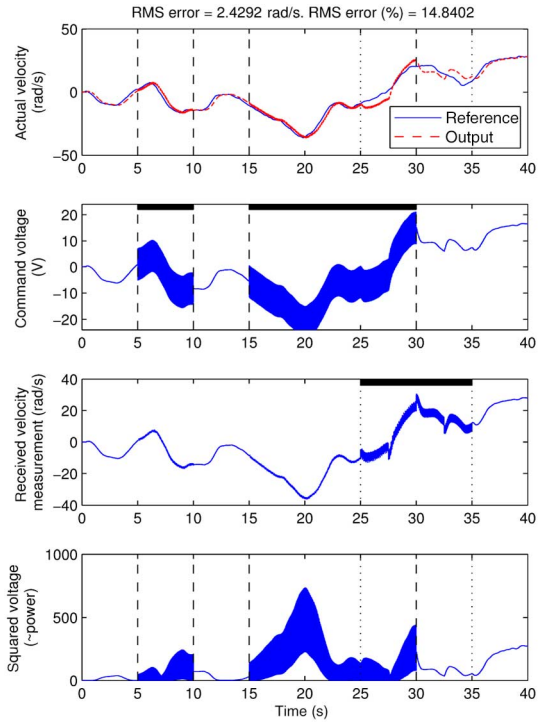

Fig. 4. Behavior of an NCS under attack. From top to bottom: control performance; applied commands; received measurements; and energy of the applied commands. Attacks: $d_{\mathrm{C2P}}(t) = A\sin(\omega_a t)$ ($A = 6$, $\omega_a = 2\pi 40$ rad/s) is added to the commands on the intervals $[5, 10]$ and $[15, 30]$ s; $d_{\mathrm{P2C}}(t) = A\ \mathrm{square}(t/P)$ ($A = 5$, period $P = 5$ s, duty cycle 50%) is added to the measurements on the interval $[25, 35]$ s.

and measurements; all signals, independently of their protection level, arrive untouched at plant side. The fourth plot shows the energy dissipated by the plant's actuator to apply the command. Such energy may be a source of damages for the actuator and the plant itself and therefore, its value is carefully considered during control design. Fig. 4 reports the behavior of the same system when an attack is performed in the intervals between dashed vertical lines on the controller-to-plant channel and between dotted lines on the plant-to-controlled channel. In the first plot, it is worth noting that the tracking error is only slightly affected by the attack showing that the attacker cannot be easily detected just by looking at this metric. The second and third plots show the presence of the attacker signal d_{C2P} superimposed to the original command and of d_{P2C} superimposed to the measurement, respectively. These signals cannot be distinguished from normal perturbation at plant and controller side and they are compensated by the closed-loop control, thus making the attack undetectable. The fourth plot shows that the attack increases the energy dissipated by the plant's actuator, thus increasing its damage probability. It is worth highlighting that d_{C2P} has a much stronger effect on the energy than d_{P2C}, as it is clear from the closed-loop block diagram in Fig. 1. On the other hand, d_{P2C} has a much stronger effect on e than d_{C2P}.

Furthermore, in general, an attack cannot be revealed by detecting the change of statistical properties of a sequence of commands or measurements; in fact, such change can be part of the normal control behavior, i.e., after a sudden change in the reference signal. Past literature avoided this problem by assuming a steady-state condition [25].

D. Objectives and Assumptions

A novel energy-efficient security-aware control architecture should have:

1) an energy-efficient mechanism to promptly detect attacks;
2) an attack mitigation strategy which is also able to detect the end of the attack interval;
3) a mechanism to save transmission energy without compromising control performance.

The proposed architecture is based on the concept of *selective encryption* according to which not all packets belonging to a given path (i.e., from controller to plant and vice versa) are protected. We assume that the transmission protocol allows to use the signature on a packet-by-packet basis. The signature approach is quite independent of the transport protocol, as it strictly requires to modify just the payload of the packet. Clearly, more powerful solutions can be obtained with the support of the protocol; e.g., IETF proposed a security-enabled real-time transport protocol, which can be used in NCSs [28].

The following objectives will be addressed:

1) detection of attack interval boundaries by comparing the statistical properties of the encrypted and unencrypted sequences;
2) adaptation of the transmission rate according to the instantaneous control performance.

This work is based on the following assumptions and working conditions.

A1) Energy consumption for computation is negligible with respect to energy consumption for transmission; the ratio is about 22% for the 5-year-old Texas Instruments CC2530 SoC [36]; while microcontroller consumption is reducing every year, transmission energy is going to remain quite unchanged if application requirements (e.g., transmission range) and the network infrastructure are not deeply modified.

A2) Energy consumption for packet encryption is negligible; in modern SoCs, encryption is performed by dedicated hardware with a very low energy footprint.

A3) Packet encryption adds a signature at the end of the message; therefore, the transmission of an encrypted packet requires more energy than the transmission of an unencrypted packet because more bits are transmitted; this assumption is independent of CPU power, hardware architecture, transmission standard, and encryption standard.

A4) Packet encryption does not alter the sample time of the system and the transmission delay; in fact, we assume that the sample time step is larger enough to allow the higher transmission time of an encrypted packet; for instance, the time overhead to transmit the 16-byte signature of an encrypted packet in IEEE 802.11 b is 12 μs; this way, the proposed approach does not require to change timing requirements for the NCS.

A5) End-to-end security protocols are used, i.e., packet signing and integrity check are performed at controller and

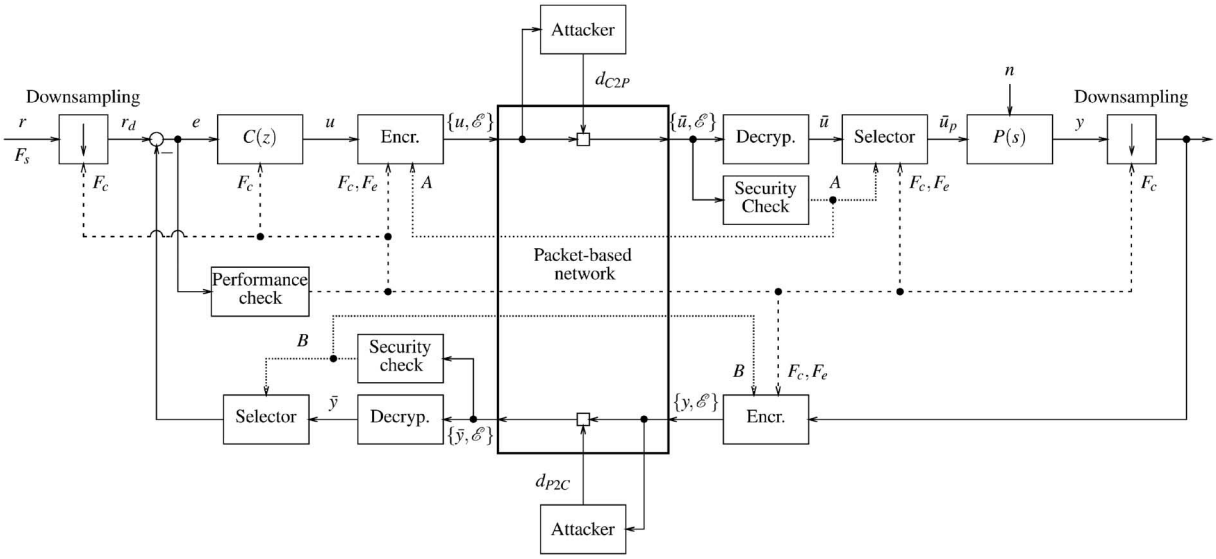

Fig. 5. Energy-efficient intrusion detection and mitigation architecture.

plant side while intermediate network devices just relay packets. In this way, a man-in-the-middle attack on a tampered network device cannot modify signed data without being discovered.

A6) The attacker alters signal from controller to plant and plant to controller to damage the plant and to reduce control performance; it keeps untouched the encrypted messages to avoid being discovered.

A7) If an encrypted packet is corrupted, then the presence of the attacker is notified to the system manager for further security checks; this is the common approach in traditional IDSs and firewalls.

A8) The network is not affected by time-varying transmission delays and packet drops; time-varying delays can be compensated by a buffer at receiver side [31], whereas packet drops can be compensated by retransmission as done in TCP; in both cases, the resulting effect is a constant transmission delay.

The architecture and the components achieving these objectives will be described in the next section.

III. Proposed Architecture

In this section, we propose an architecture to handle the problems highlighted in the previous section. Fig. 5 shows the block diagram of the architecture.

The intrusion detection mechanism is implemented in the *Security Check* blocks while the adaptation of transmission rate according to the instantaneous control performance is performed in the *Performance Check* block; they will be described in detail in the specific sections; here, we list the meaning of the other blocks.

1) *Controller $C(z)$:* It is a discrete-time system running at F_c, with $F_c \leq F_s$, where F_s is the maximum sampling frequency the feedback system can run. It computes the command u to be sent through the network based on the tracking error $e = r - \bar{y}$, where r is the reference and $\bar{y}$ is the decrypted measurement received from the plant (or its down-sampled version $\bar{y}_{\mathrm{DS}}$). The difference equation describing $C(z)$ is parametrized on the sample time $T_c = 1/F_c$ to allow the controller to be easily adapted to a different sampling frequency.
2) *Encryption block:* This system encrypts a fraction of the incoming packets. $\mathcal{E} = 1$ means that the current packet is encrypted and $\mathcal{E} = 0$ means that the packet content is unencrypted. Encryption means that the signature of the message is inserted in the packet.
3) *Decryption block:* This block checks whether the packet is encrypted and, in this case, it verifies the integrity of the contained message; if an alteration is found, the attacker is revealed. It is worth noting that the proposed intrusion-detection mechanism is not performed by this block; in fact, we assume that the attacker is smart and it does not corrupt encrypted packets to avoid to be revealed.
4) *Plant $P(s)$:* A continuous-time system with input $\bar{u}$ (or its down-sampled version $\bar{u}_{\mathrm{DS}}$) and output y.
5) *Selector:* The behavior of this block depends on the output of the Security Check block; if an intrusion is detected ($\mathcal{A} = 1$ in the controller-to-plant channel or $\mathcal{B} = 1$ in the plant-to-controller channel), the selector discards unencrypted packets, so that they are not used since their content is not trusted.
6) *Attacker:* The attacker tampers only unencrypted packets. In this work, we assume additive corruptions ([27]) of the commands sent by the controller to the plant

$$\bar{u}(t) = \begin{cases} u(t), & \text{if } \mathcal{E} = 1 \\ u(t) + d_{\mathrm{C2P}}(t), & \text{if } \mathcal{E} = 0 \end{cases} \tag{1}$$

and of the measurements sent by the plant to the controller

$$\bar{y}(t) = \begin{cases} y(t), & \text{if } \mathcal{E} = 1 \\ y(t) + d_{\mathrm{P2C}}(t), & \text{if } \mathcal{E} = 0 \end{cases} \tag{2}$$

where $t = kT_c$, $k \in \mathbb{N}$.

The reference signal $r(\cdot)$ is sampled at frequency F_s (the maximum frequency loop) and it can be down-sampled at F_c when needed.

A. Security Check

The attack is seen by the feedback configuration as an additive disturbance, as modeled in (1) and (2). Therefore, the feedback configuration itself can (partially) compensate for it, as it is well known from every basic course on control theory. This means that, e.g., a modified version $\bar{u}$ of the control command u could generate a similar tracking error e because the sensitivity transfer function attenuates the impact of d_{C2P}. In other words, just looking at the statistics of the tracking error is not enough to detect an attack as shown comparing the time series in Figs. 3 and 4. A similar reasoning could be done for an attack on the feedback channel, i.e., d_{P2C}, even though the transfer function from d_{P2C} to e is different than the transfer function from d_{C2P} to e.

For this reason, we introduce a selective encryption of the signals (i.e., commands or measurements) sent through the network not only to protect them *per se* in an energy-efficient way, but also to provide a way to detect an intrusion. To explain the idea, let us define F_s ($T_s = 1/F_s$) as the maximum sampling rate of the controlled loop and F_e ($T_e = 1/F_e$) as the constant frequency of the subsampled time series used to detect the intrusion.

At the plant side, we can compare the statistics of two time series: the first one is related to encrypted packets

$$U^e = \left[\cdots \ \bar{u}((k-1)T_e) \ \bar{u}(kT_e) \ \bar{u}((k+1)T_e) \ \cdots\right] \tag{3}$$

the other is related to unencrypted packets

$$U^{ne} = \left[\cdots \ \bar{u}((k-1)T_e + T_e/2) \ \bar{u}(kT_e + T_e/2) \ \bar{u}((k+1)T_e + T_e/2) \cdots\right] \tag{4}$$

where $T_e/2$ is a temporal offset.

In the rest of this paper, the following symbols will be used:

- ● encrypted packet that could be or not be sent according to the value of control performance (see Section III-B);
- ○ unencrypted packet that could be or not be sent according to the value of control performance;
- ■ encrypted packet collected in U^e at frequency F_e;
- □ unencrypted packet collected in U^{ne} at frequency F_e.

The last two kinds of packets are called *anchor packets* since they are always present as required by the intrusion detection algorithm.

Fig. 6 shows the (encrypted ■/unencrypted □) packets at F_e. These two time series must always be available, whereas the packets in between indicated with ○ or ● are sent or not according to the instantaneous control performance. The upper figure represents the case when no attack is detected ($\mathcal{A} = 0$) while the lower one represents the case when an attack is detected ($\mathcal{A} = 1$).

In normal working condition (i.e., the attacker does not modify data), the statistics of U^e and U^{ne} should be quite close. If the attacker modified the unencrypted packets, the two time series will be statistically different.

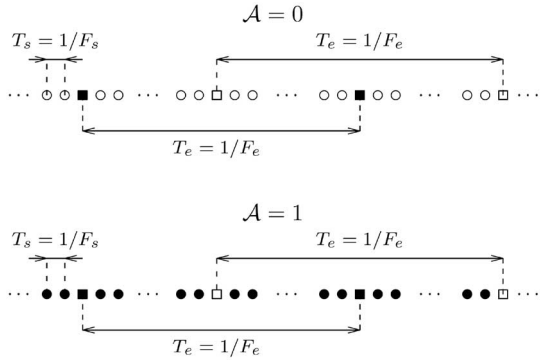

Fig. 6. Encrypted (■) and unencrypted (□) time series for the intrusion detection.

Several methods can be found in statistics to compare the two time series and to answer the questions. Are they consistent? Are they generated by the same statistical distribution? We refer the reader to the book [20] and in particular to the chapter discussing the algorithms for testing the *goodness of fit*.

It is important to highlight that to compute the statistics and at the same time to be reactive to abrupt attacks, the metrics are evaluated on subsets of (3) and (4), i.e., on two moving windows of the same length $U^e_{[k-W^u,k]}$ and $U^{ne}_{[k-W^u,k]}$. The number of samples W^u is a design parameter and its choice is a tradeoff between promptness of the detection (the smaller W^u, the better) and meaningfulness of the statistical analysis (the larger the W^u, the better). In fact, the meaningfulness of the statistics is related to the detection performance, i.e. the false rejection rate (FRR) and the false acceptance rate (FAR).

In Section IV, some simulation results are provided where only the first two statistical moments, i.e., 1) average; and 2) variance, are used. However, more sophisticated tools could be designed taking also into account *a priori* information such as the spectrum of the reference signal r, the mathematical models for the plant and the controller, and the measurement noise distribution. The goal of this paper is to show how the attack detection and mitigation can be implemented in a wireless control system. The same strategy can also be adopted in more complex cyber-physical systems where digital and analog devices work together.

The algorithm implemented within the *Security Check* block at the plant side in Fig. 5 is as follows:

function $\mathcal{A}$=SECURITYCHECK$\left(U^e_{[k-W^u,k]}, U^{ne}_{[k-W^u,k]}\right)$
▷ Compute means
  $\hat{\mu}^u_e(k)$ =mean$(U^e_{[k-W^u,k]})$
  $\hat{\mu}^u_{ne}(k)$ =mean$(U^{ne}_{[k-W^u,k]})$
▷ Compute standard deviations
  $\hat{\sigma}^u_e(k)$ =std$(U^e_{[k-W^u,k]})$
  $\hat{\sigma}^u_{ne}(k)$ =std$(U^{ne}_{[k-W^u,k]})$
▷ Testing hypothesis on means and standard deviations
  if $|\hat{\mu}^u_e - \hat{\mu}^u_{ne}| > T^u_\mu$ OR $|\hat{\sigma}^u_e - \hat{\sigma}^u_{ne}| > T^u_\sigma$ **then**
    $\mathcal{A} = 1$ ▷ Attack detected
  else
    $\mathcal{A} = 0$ ▷ No attack
  end if
end function

When the testing of at least one of the following hypothesis:

$$H_1 : |\hat{\mu}_e^u - \hat{\mu}_{ne}^u| < T_\mu^u$$
$$H_2 : |\hat{\sigma}_e^u - \hat{\sigma}_{ne}^u| < T_\sigma^u$$

fails, it means that an attack has been *statistically* detected and some countermeasures have to be taken to mitigate its effects. First of all, only the encrypted data should be used in the feedback control, i.e., the ones that certainly have not been tampered. The second step is to alert the encryption block, so that more packets must be encrypted to preserve the performance level since only encrypted packets will be actually used.

The behavior of the *Security Check* block at the controller side $\mathcal{B} = \text{SECURITYCHECK}(Y^e_{[k-W^y,k]}, Y^{ne}_{[k-W^y,k]})$ is exactly the same in addition to the constants W^y, T^y_μ, and T^y_μ that are now related to the measurement signal.

When the wireless control system is under attack ($\mathcal{A} = 1$ and/or $\mathcal{B} = 1$), all the packets belonging to the attacked flow are encrypted, thus increasing energy consumption. Therefore, their transmission rate should be carefully adapted to the desired control performance to avoid energy waste. For this reason, the attack mitigation strategy has been combined with a performance check mechanism as described in the next section.

Remark: The selection of the thresholds T^u_μ and T^u_σ (or T^y_μ, T^y_σ) is of paramount importance in the proposed algorithms. When the transfer functions of the involved systems and the statistics of the signals are known, analytical expressions can be derived. Let $S_{rr}(e^{j\omega})$, $S_{nn}(e^{j\omega})$ and $T_{ru}(z)$, $T_{nu}(z)$ be the spectrum of the reference r and noise n, and the transfer functions mapping r into u and n into u, respectively. For example, to compute T^u_σ, we derive the spectrum of the command[2]

$$S_u(e^{j\omega}) = S_n(e^{j\omega})|T_{nu}(e^{j\omega})|^2 + S_r(e^{j\omega})|T_{ru}(e^{j\omega})|^2 \quad (5)$$

and the spectrum of its moving average (MA)

$$S_{u_{\text{MA}}}(e^{j\omega}) = S_u(e^{j\omega})|T_{\text{MA}}(e^{j\omega})|^2 \quad (6)$$

where

$$T_{\text{MA}}(z) = \frac{1 + z + \cdots + z^{W^u}}{W^u z^{W^u}}. \quad (7)$$

The theoretical variance is then

$$\sigma^2_{u_{\text{MA}}} = R_{u_{\text{MA}}}(0) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} S_{u_{\text{MA}}}(e^{j\omega})d\omega. \quad (8)$$

If the difference between $\hat{\sigma}^u_e$ and $\hat{\sigma}^u_{ne}$ is smaller than, e.g., 20% of $\sigma_{u_{MA}}$ (i.e., $T^u_\sigma = \frac{1}{5}\sigma_{u_{\text{MA}}}$), the SECURITYCHECK module does not trigger an attack alert. Similar arguments can be exploited to analytically derive other thresholds. This approach is theoretically reasonable, but it assumes to know perfectly the time-invariant plant model and the statistics of the signals. This is rarely the case and thus, in our opinion, it is wiser to derive the thresholds by analyzing time series taken when the system is in operation and there are no attacks. For example, in our simulation setup, we set $T^u_\sigma := 2\max_{k\in\mathcal{T}}|\hat{\sigma}^u_e(k) - \hat{\sigma}^u_{ne}(k)|$, where $\hat{\sigma}^u_e(\cdot)$ and $\hat{\sigma}^u_{ne}(\cdot)$ are the variance values when no attack is performed (i.e., during the interval $\mathcal{T}$). They are quite similar since the only difference is due to temporal mismatching. This approach is much more robust to unmodeled dynamics and uncertain statistics and, in particular, it does not require time-invariant statistics.

[2]We assume r and n independent, and we use $z = e^{j\omega}$.

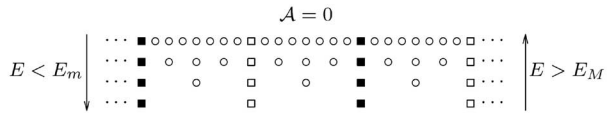

Fig. 7. Increasing/decreasing the data frequency according to the value of E when the system is not under attack ($\mathcal{A} = 0$). Meaning of the symbols: ○, optional unencrypted data; ●, optional encrypted data; □, anchor unencrypted data; ■, anchor encrypted data.

The other thresholds are computed similarly

$$T^u_\mu := 2\max_{k\in\mathcal{T}}|\hat{\mu}^u_e(k) - \hat{\mu}^u_{ne}(k)| \quad (9)$$
$$T^y_\mu := 2\max_{k\in\mathcal{T}}|\hat{\mu}^y_e(k) - \hat{\mu}^y_{ne}(k)| \quad (10)$$
$$T^y_\sigma := 2\max_{k\in\mathcal{T}}|\hat{\sigma}^y_e(k) - \hat{\sigma}^y_{ne}(k)|. \quad (11)$$

B. Performance Check and Intrusion Mitigation

When the NCS is not under attack, the frequency of the control loop F_c is equal to or smaller than F_s according to the performance. To save energy, the policy consists in sending the smallest number of samples that guarantees to match the required performance. In this "normal" scenario, the encrypted packets are just the ones needed to compute the sequence $U^e_{[k-W^u,k]}$ (or $Y^e_{[k-W^y,k]}$), i.e., data sampled at F_e. Fig. 7 shows how the *performance check* adapts the sampling rate according to the rms error E on a M-length moving window $e(k-M), e(k-M+1), \ldots, e(k)$. To avoid chattering, the requirement on the performance takes the form

$$E_m < E < E_M \quad (12)$$

where the lower and upper bounds E_m and E_M are application-dependent.

If the control loop is over-performing ($E < E_m$), the sampling rate of the control loop F_c can be decreased, whereas if the control loop is under-performing ($E > E_M$), the sampling rate has to be increased. The highest value for F_c is equal to F_s, whereas the lowest is equal to either F_e in case of attack (when only encrypted anchor packets are used) or $2F_e$ when no attack is active and also unencrypted packets are used. To simplify the analysis, we assume that F_s and F_e are related by a power-of-two coefficient.

Fig. 7 reports a basic example showing the mechanism of increasing or decreasing the sample frequency F_c in case of $E > E_M$ or $E < E_m$, respectively, when the system is not under attack ($\mathcal{A} = 0$).

The notification of the intrusion detection is sent to the *selector*, which discards all unencrypted packets, as well as to the *encryption block* at the opposite side of the system, so that all the packets are encrypted to guarantee the same level of performance. Moreover, in this case, the *performance check* block will change the number of transmitted packets

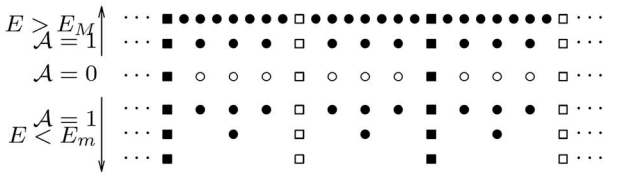

Fig. 8. Example of increasing/decreasing the data frequency according to the value of E when the system detect an attack (from $\mathcal{A} = 0$ to $\mathcal{A} = 1$). Meaning of the symbols: ○, optional unencrypted data; ●, optional encrypted data; □, anchor unencrypted data; ■, anchor encrypted data.

using the same policy explained above to keep the performance between E_m and E_M with the minimum energy consumption. Fig. 8 shows a possible evolution of the frequency F_c when an attack is detected and when the performance are below E_m or above E_M.

When the control loop frequency changes, the controller also has to be updated accordingly. For example, to change the update rate of a PID controller $C(z)$, its discrete-time formulation as a function of the sample time ($T_c = 1/F_c$) can be implemented

$$u(k) = u(k-1) + \left(k_P + k_I T_c + \frac{k_D}{T_c}\right) e(k) - \left(k_P + 2\frac{k_D}{T_c}\right) e(k-1) + \frac{k_D}{T_c} e(k-2). \quad (13)$$

A similar parametrization can be done if the controller is designed in the state space domain, [4]. Even though not reported here, a stability analysis has to be done to be sure that the switching between different controllers does not compromise the stability of the overall system. The stability analysis is not the focus of this paper; many techniques can be applied as surveyed in [21].

The adaptation of F_c can be implemented with the following algorithm in the *performance check* block.

```
1: function F_c = PERFORMANCECHECK(E)
                                        ▷ Check performance
2:     if E > E_M then
                                        ▷ Send more data
3:         F_c = min{F_c^max, 2F_c^old}
4:     else if E < E_m then
                                        ▷ Send less data
5:         F_c = max{F_c^min, F_c^old/2}
6:     else
                                        ▷ Do nothing
7:     end if
8: end function
```

In the actual implementation of this algorithm in Section IV, a time constant WT_c is introduced between two changes of the transmission rate. This regularization mechanism is needed to avoid too many changes in a short time.

The combination of the *security check* and of the *performance check* modules allows to statistically detect an intrusion and to mitigate its effects. The objective of the proposed architecture is not only to improve the security of the transmission (that could be guaranteed by encrypting all the packets sent at the maximum sampling frequency) but also to adapt the transmission rate according to the instantaneous control performance to save energy. This mechanism is effective especially during an attack when more energy is consumed since all packets are encrypted.

C. Energy Analysis

This section aims at analyzing the energy consumption of the proposed architecture and comparing it with traditional approaches.

Let c_e and c_{ne} be the energy used to transmit an encrypted or an unencrypted packet, respectively. The proposed approach is based on the transmission of a mixture of encrypted and unencrypted packets organized in a regular pattern as depicted in Fig. 6. Let n be the number of optional packets between two anchor packets; this number is related to F_c. If we consider a time period of length T_e, there are always $2n$ optional encrypted/unencrypted packets (●/○), one anchor encrypted packet (■) and one anchor unencrypted packet (□); therefore, the power consumed by the transmitter in case of detected attack is given by

$$\mathcal{P}_1 = \frac{2nc_e + c_e + c_{ne}}{T_e} \quad (14)$$

otherwise by

$$\mathcal{P}_0 = \frac{2nc_{ne} + c_{ne} + c_e}{T_e}. \quad (15)$$

Let p be the probability of attack; it can be easily computed from the Markov model in Fig. 2. We assume that the *security check* block takes always the same time to detect an attack at the begin and end. Then, the expected value of power consumption will be given by

$$\mathbf{E}[\mathcal{P}] = p\mathcal{P}_1 + (1-p)\mathcal{P}_0. \quad (16)$$

By replacing (14) and (15) in (16), after some mathematical manipulations, the following expression is derived:

$$\mathbf{E}[\mathcal{P}] = \frac{2n}{T_e}[pc_e + (1-p)c_{ne}] + \frac{c_{ne} + c_e}{T_e}. \quad (17)$$

As expected, the power consumption has two contributions: 1) the first contribution is variable and depends on the amount of packets sent between two anchor packets, which is strongly related to F_c and therefore, to the *performance check* algorithm; and 2) the second contribution is constant and it represents the cost of attack detection. This contribution cannot be avoided even if the probability of attack is zero and the control performance allows the transmission of the minimum number of packets.

A common alternative to our proposed approach consists in protecting all the packets, no matter an attack is present or not. If we consider the same interval T_e and assume the same transmission rate, the power consumption in this case is given by

$$\mathcal{P}_a = \frac{2nc_e + 2c_e}{T_e}. \quad (18)$$

We can compute the *relative gain* on power saving for our approach with respect to the common one in best and worst cases ($p = 0$ and $p = 1$, respectively) as follows:

$$\frac{\mathcal{P}_a - \mathbf{E}[\mathcal{P}]}{\mathcal{P}_a} = \begin{cases} \frac{2n+1}{2n+2}(1 - \frac{c_{ne}}{c_e}) \simeq (1 - \frac{c_{ne}}{c_e}), & \text{if } p = 0 \\ \frac{1}{2n+2}(1 - \frac{c_{ne}}{c_e}), & \text{if } p = 1. \end{cases}$$

Since $c_{ne} < c_e$, there is always a gain in using our approach. When the attack probability is low, the gain is quite independent from the transmission rate F_c. When the attack probability is high, the power saving gain is proportional to the transmission rate; this analytical result shows the importance of the *performance check* block in our architecture since it reduces the transmission rate when instantaneous control performance are above the desired threshold.

IV. Simulation Results

In this section, the proposed energy-efficient intrusion detection and mitigation architecture (Fig. 5) is validated on a wireless control system. The plant is a dc motor with transfer function mapping voltage $v(t)$ into angular velocity $\omega(t)$ given by

$$P(s) = \frac{\hat{\omega}(s)}{\hat{V}(s)} = \frac{K_m}{(Js + b)(Ls + R) + K_m K_e} \tag{19}$$

where the electromechanical parameters are $R = 3.9\ \Omega$, $L = 0.0023$ H, K_e =0.091 V/(rad/s), $J = 0.001$ kg m^2 (rotor+load), $B = 0.0115$ Nm/(rad/s) (rotor+load), and $K_m = 0.09$ Nm/A. The controller $C(z)$ is a PI controller in which the integral gain is proportional to the sampling frequency of the control loop as in (13) where the derivative part has not been considered. We assume an additive Gaussian noise n for the tachometer with standard deviation[3] equal to 0.04 rad/s ($\sim$2.3 deg /s). The transmission delays are equal to 20 ms in both channels.

According to the algorithm detailed in Section III, the following control parameters have been chosen.

1) *Security check block:* Length of the windows $W^u = W^y = 20$; thresholds for the average mean $T^u_\mu = 0.5$, $T^y_\mu = 0.8$; thresholds for the average standard deviation $T^u_\sigma = 0.2$, $T^y_\sigma = 0.6$. The sample time of the time series U^e and U^{ne} is $T_e = 64T_s$, where $T_s = 1$ ms.
2) *Performance check block:* Length of the windows $M = 50$; thresholds for the error rms: $E_m = 1$, $E_M = 3$; upper and lower bounds of the sample time for the control loop $T^{\max}_c = 32T_s$, $T^{\min}_c = T_s$.

In the present case, we assume that the attacker adds a high-frequency vibration signal on the commands

$$\bar{u}(kT_c) = u(kT_c) + d_{\mathrm{C2P}}(kT_c) \quad [\text{if } \mathcal{E}(k) = 0]$$
$$d_{\mathrm{C2P}}(kT_c) = A\sin(\omega_a kT_c)$$

[3]Angular velocity derived by encoders will have to consider also the error due to the numerical derivative, which could be quite large when the velocity is close to zero and the sampling rate is high.

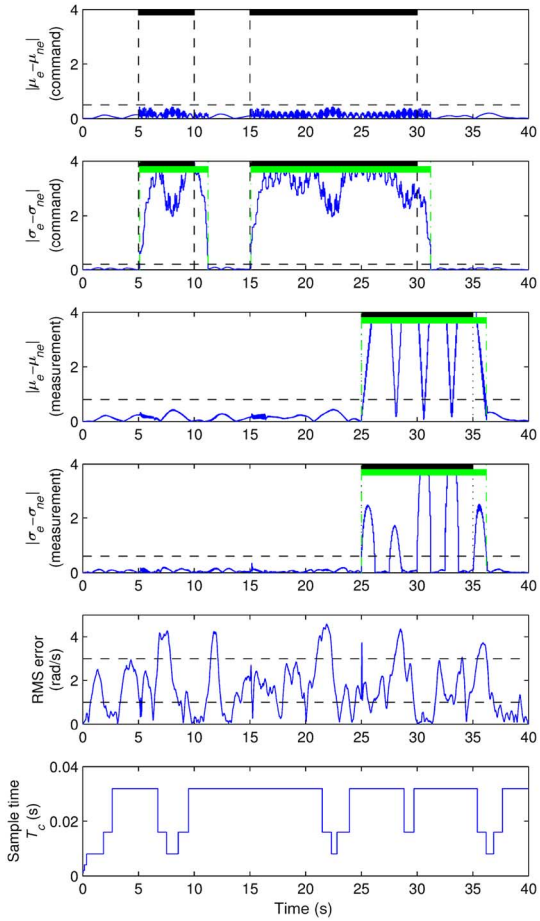

Fig. 9. Application of the energy-efficient intrusion detection and mitigation. First plot: $|\hat{\mu}^u_e - \hat{\mu}^u_{ne}|$ with threshold T^u_μ. Second plot: $|\hat{\sigma}^u_e - \hat{\sigma}^u_{ne}|$ with threshold T^u_σ. Third plot: $|\hat{\mu}^y_e - \hat{\mu}^y_{ne}|$ with threshold T^y_μ. Fourth plot: $|\hat{\sigma}^y_e - \hat{\sigma}^y_{ne}|$ with threshold T^y_σ. Fifth plot: error rms E and thresholds $E_m = 1$, $E_M = 3$. Sixth plot: sample time of the controller T_c. Attacks: $d_{\mathrm{C2P}}(t) = A\sin(\omega_a t)$ ($A = 6$, $\omega_a = 2\pi 40$ rad/s) is added to the commands on the intervals $[5, 10]$ and $[15, 30]$ s; $d_{\mathrm{P2C}}(t) = A\ \mathrm{square}(t/P)$ ($A = 5$, period $P = 5$ s, duty cycle 50%) is added to the measurements on the interval $[25, 35]$ s.

where $A = 6$, $\omega_a = 2\pi 40$ rad/s, and a square signal on the measurements

$$\bar{y}(kT_c) = y(kT_c) + d_{\mathrm{P2C}}(kT_c) \quad [\text{if } \mathcal{E}(k) = 0]$$
$$d_{\mathrm{P2C}}(kT_c) = A\,\mathrm{square}(kT_c/P)$$

where $A = 5$, the period P is 5 s and the duty cycle is 50%.

The sample time T_c is the current sample time of the control loop. As explained in the previous section, this value changes at run-time according to the rms error E. These disturbances are just an example of a possible attack: the proposed architecture can detect also offset, ramp, and whatever signals that change the statistical properties of the sequence. The attack is switched ON during the intervals $[5, 10]$ and $[15, 30]$ s in the controller-to-plant channel and during the interval $[25, 35]$ s in the plant-to-controller channel.

In Section III, we have already introduced the average and the standard deviation as statistical metrics. The first four plots in Fig. 9 show the values of $|\hat{\mu}^u_e - \hat{\mu}^u_{ne}|$, $|\hat{\sigma}^u_e - \hat{\sigma}^u_{ne}|$ and

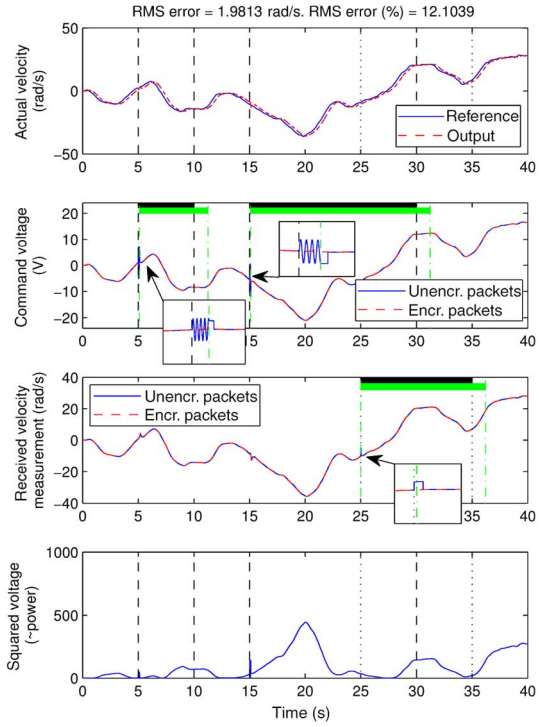

Fig. 10. Application of the energy-efficient intrusion detection and mitigation. Attacks: $d_{\mathrm{C2P}}(t) = A\sin(\omega_a t)$ ($A = 6$, $\omega_a = 2\pi 40$ rad/s) is added to the commands on the intervals $[5, 10]$ and $[15, 30]$ s; $d_{\mathrm{P2C}}(t) = A\ \mathrm{square}(t/P)$ ($A = 5$, period $P = 5$ s, duty cycle 50%) is added to the measurements on the interval $[25, 35]$ s.

$|\hat{\mu}_e^y - \hat{\mu}_{ne}^y|$, $|\hat{\sigma}_e^y - \hat{\sigma}_{ne}^y|$ with their thresholds derived as explained in the Remark in Section III. It is possible to see that the metric based on $|\hat{\mu}_e^u - \hat{\mu}_{ne}^u|$ is not able to detect the attack: the encrypted and unencrypted MAs overlap (as expected because this type of disturbance has zero mean). Vice versa, a constant bias would be clearly detected. The metric based on $|\hat{\sigma}_e - \hat{\sigma}_{ne}|$ detects the sinusoidal attack. The detection lag is indicated by the difference between the vertical dashed–black line (when the attack actually starts) and the green dashed–dotted line (when the attack is statistically detected). The lag is the side effect of the averaging on a window of length W^u. The larger the W^u, the larger the lag. Of course to compute a consistent value for the variance, this number cannot be too small.

Similar considerations can be given analyzing the metrics $|\hat{\mu}_e^y - \hat{\mu}_{ne}^y|$ and $|\hat{\sigma}_e^y - \hat{\sigma}_{ne}^y|$. Since the disturbance d_{P2C} is a square signal with a large period, both metrics are able to easily detect the attack on the feedback channel.

The last two plots show the error rms (fifth row) that the mitigation system tries to maintain between the upper and the lower bounds by adapting the sample time T_c (sixth row).

The effect of the lags in detecting the beginning of the attack and its ending is shown in Fig. 10: the plot on the top shows the comparison between the reference and the plant output, the two plots in the middle show the command and the measurement, and the plot on the bottom shows the squared command. Looking at the commands, it is easy to see that the values are with the superimposed sinusoid until the attack is detected. The situation is clearly improved with respect to Fig. 4 when the system is under attack but no detection and mitigation algorithms were implemented. In the interval between the beginning of the attack and its detection, the commands are tampered, but later the plant uses only encrypted packets, which are not corrupted by the attacker. The tampered unencrypted data in U^{ne} and Y^{ne} are thrown away thanks to the *selectors* placed before the plant and the controller.

V. Conclusion

We proposed an energy-efficient security-aware wireless control architecture. We have shown that the intrusion is hard to be distinguished from normal disturbance at plant side. Encryption-based packet protection is energy-consuming for battery-powered devices. We showed that selective encryption allows to save energy and to detect attack at the begin and end. We also showed how the number of encrypted packets can be adapted according to the presence of the attack, so that more energy is used only when needed. Since packet transmission consumes energy, we also proposed to adapt transmission rate to instantaneous control performance. Simulation results showed that the technique promptly reacts to attacks while energy saving was demonstrated analytically.

References

[1] *IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer.* IEEE Standard 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011), Apr. 2012, pp. 1–225.

[2] A.-A. Ahmadi, F. R. Salmasi, M. Noori-Manzar, and T.A. Najafabadi, "Speed sensorless and sensor-fault tolerant optimal PI regulator for networked DC motor system with unknown time-delay and packet dropout," *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 708–717, Feb. 2014.

[3] S. Amin, G. A. Schwartz, and A. Hussain, "In quest of benchmarking security risks to cyber-physical systems," *IEEE Netw.*, vol. 27, no. 1, pp. 19–24, Jan. 2013.

[4] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1997.

[5] D. J. Bernstein, "The Poly1305-AES message-authentication code," in *Fast Software Encryption*, H. Gilbert and H. Handschuh, Eds. Berlin, Germany: Springer, 2005, vol. 3557, pp. 32–49.

[6] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives," in *Proc. IEEE Conf. Design Autom. Test Eur. (DATE)*, Mar. 2005, vol. 1, pp. 196–201.

[7] C. Brandolese, "Source-level estimation of energy consumption and execution time of embedded software," in *Proc. 11th EUROMICRO Conf. Digital Syst. Design Archit. Methods Tools (DSD)*, Sep. 2008, pp. 115–123.

[8] C. E. Capovilla, I. R. Santana Casella, A. J. Sguarezi Filho, T. A. dos Santos Barros, and E. Ruppert Filho, "Performance of a direct power control system using coded wireless OFDM power reference transmissions for switched reluctance aerogenerators in a smart grid scenario," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 52–61, Jan. 2015.

[9] F. De Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella, "On the use of wireless networks at low level of factory automation systems," *IEEE Trans. Ind. Informat.*, vol. 2, no. 2, pp. 129–143, May 2006.

[10] D. Dzung, M. Naedele, T. P. von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proc. IEEE*, vol. 93, no. 6, pp. 1152–1177, Jun. 2005.

[11] D. Engel, "Wavelet-based load profile representation for smart meter privacy," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT'13)*, Feb. 2013, pp. 1–6.

[12] D. Feng *et al.*, "A survey of energy-efficient wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 167–178, Jan. 2013.

[13] A. A. Ghorbani, W. Lu, and M. Tavallaee, "*Network Intrusion Detection and Prevention: Concepts and Techniques*", 1st ed. New York, NY, USA: Springer, 2009.
[14] J. D. Gibson *et al.*, "Selective encryption and scalable speech coding for voice communications over multi-hop wireless links," in *Proc. IEEE Mil. Commun. Conf.(MILCOM'04)*, Oct. 2004, vol. 2, pp. 792–798.
[15] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.
[16] J. Hirai, K. Tae-Woong, and A. Kawamura, "Practical study on wireless transmission of power and information for autonomous decentralized manufacturing system," *IEEE Trans. Ind. Electron.*, vol. 46, no. 2, pp. 349–359, Apr. 1999.
[17] S. Hussain, M. Mokhtar, and J. M. Howe, "Sensor failure detection, identification, and accommodation using fully connected cascade neural network," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1683–1692, Mar. 2015.
[18] B. H. Jung, R.U. Akbar, and D. K. Sung, "Throughput, energy consumption, and energy efficiency of IEEE 802.15.6 body area network (BAN) MAC protocol," in *Proc. IEEE 23rd Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2012, pp. 584–589.
[19] L. Man *et al.*, "Static security optimization for real-time systems," *IEEE Trans. Ind. Informat.*, vol. 5, no. 1, pp. 22–37, Feb. 2009.
[20] E. L. Lehmann, *Testing Statistical Hypotheses*. Berlin, Germany: Springer-Verlag, 1997.
[21] D. Liberzon, *Switching in Systems and Control*. New York, NY, USA: Springer, 2003.
[22] T. Ma *et al.*, "Assurance of energy efficiency and data security for ECG transmission in BASNs," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 1041–1048, Apr. 2012.
[23] F. Macia-Perez *et al.*, "Network intrusion detection system embedded on a smart sensor," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 722–732, Mar. 2011.
[24] J. Markoff, "A silent attack, but not a subtle one," *New York Times*, vol. 160, no. 55176, pp. 1–6, Sep. 2010.
[25] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on SCADA systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1396–1407, Jul. 2014.
[26] Y. Mo, J. P. Hespanha, and B. Sinopoli, "Resilient detection in the presence of integrity attacks," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 31–43, Jan. 2014.
[27] Y. Mo *et al.*, "Cyber–physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, Jan. 2012.
[28] S. Pallapothu and S. Mahajan, *Selective Encryption Support in SRTP*. Fremont, CA, USA: Internet-Draft Draft-smahajan-SRTP-selective-encryption-01.txt, IETF Secretariat, Feb. 2007.
[29] P. Prasithsangaree and P. Krishnamurthy, "Analysis of energy consumption of RC4 and AES algorithms in wireless LANs," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2003, vol. 3, pp. 1445–1449.
[30] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Radio-frequency fingerprinting for mitigating primary user emulation attack in low-end cognitive radios," *IET Commun.*, vol. 8, no. 8, pp. 1274–1284, May 2014.
[31] L. Repele, R. Muradore, D. Quaglia, and P. Fiorini, "Improving performance of networked control systems by using adaptive buffering," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4847–4856, Sep. 2014.
[32] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S.S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.
[33] G. Semeraro *et al.*, "Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling," in *Proc. 8th Int. Symp. High Perform. Comput. Archit.*, Feb. 2002, pp. 29–40.
[34] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 580–591, Mar. 2014.
[35] J. A. Stankovic, T. F. Abdelzaher, C. Lu, L. Sha, and J. C. Hou, "Real-time communication and coordination in embedded sensor networks," *Proc. IEEE*, vol. 91, no. 7, pp. 1002–1022, Jul. 2003.
[36] Texas Instruments, *CC2530: A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications*. Dallas, TX, USA: Texas Instruments, 2011.
[37] R. Want, B. Schilit, and D. Laskowski, "Bluetooth LE finds its niche," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 12–16, Oct. 2013.
[38] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Trans. Ind. Electron.*, vol. 49, no. 6, pp. 1265–1282, Dec. 2002.
[39] Y. Yang *et al.*, "Multiattribute SCADA-specific intrusion detection system for power networks," *IEEE Trans. Power Del.*, vol. 29, no. 3, pp. 1092–1102, Jun. 2014.
[40] W. Zeng and M.-Y. Chow, "Optimal tradeoff between performance and security in networked control systems based on coevolutionary algorithms," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 3016–3025, Jul. 2012.
[41] Y. Zhang, L. Wang, W. Sun, R. C. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.

Riccardo Muradore (S'99–M'04) received the Laurea degree in information engineering, and the Ph.D. degree in electronic and information engineering from the University of Padova, Padua, Italy, in 1999 and 2003, respectively.

He held a Postdoctoral Fellowship with the Department of Chemical Engineering, University of Padova, from 2003 to 2005. Then, he spent three years with the European Southern Observatory, Munich, Germany, as a Control Engineer working on adaptive optics systems. In 2008, he joined the A Laboratory for Teleoperation and Autonomous Intelligent Robots (ALTAIR) Robotics Laboratory, University of Verona, Verona, Italy, where since 2013, he has been an Assistant Professor. His research interests include robust control, robotics, teleoperation, networked control systems, and adaptive optics.

Davide Quaglia (S'00–M'03) received the Ph.D. degree in computer engineering from Politecnico di Torino, Turin, Italy, in 2003.

He is currently an Assistant Professor with the Department of Computer Science, University of Verona, Verona, Italy, where he currently teaches hardware (HW) architectures for bioinformatics and design of networked embedded systems. He is also a Co-Founder and Active Project Leader with EDALab s.r.l., Verona, a spin-off company of the University of Verona. He has authored or coauthored about 50 papers\textbooks. His research interests include networked-embedded systems, networked control systems, and cyber-physical systems.

He is Permanent Member of the Technical Program Committee of the Euromicro Conference on Digital System Design.