Formal Impact Metrics for Cyber-physical Attacks

Ruggero Lanotte
DISUIT
Univ. of Insubria, Como, Italy
ruggero.lanotte@uninsubria.it

Massimo Merro
dept. of Computer Science
Univ. of Verona, Verona, Italy
massimo.merro@univr.it

Andrei Munteanu
dept. of Computer Science
Univ. of Verona, Verona, Italy
andrei.munteanu@univr.it

Simone Tini
DISUIT
Univ. of Insubria, Como, Italy
simone.tini@uninsubria.it

2021 IEEE 34th Computer Security Foundations Symposium (CSF) | 978-1-7281-7607-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSF51468.2021.00040

***Abstract*—*Cyber-Physical systems* (CPSs) are exposed to *cyber-physical attacks*, *i.e.*, security breaches in cyberspace that adversely affect the physical processes of the systems.**

We define two *probabilistic metrics* to estimate the *physical impact* of attacks targeting cyber-physical systems formalised in terms of a probabilistic hybrid extension of Hennessy and Regan's *Timed Process Language*. Our *impact metrics* estimate the impact of cyber-physical attacks taking into account: (i) the *severity of the inflicted damage* in a given amount of time, and (ii) the *probability* that these attacks are actually accomplished, according to the dynamics of the system under attack. In doing so, we pay special attention to *stealthy attacks*, *i.e.*, attacks that cannot be detected by *intrusion detection systems*. As further contribution, we show that, under precise conditions, our metrics allow us to estimate the impact of attacks targeting a complex CPS in a *compositional* way, *i.e.*, in terms of the impact on its sub-systems.

***Index Terms*—Cyber-physical attacks, impact metrics, timed and hybrid models**

I. Introduction

Cyber-Physical Systems (CPSs) are physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing and communication core [1]. The growing connectivity and integration of these systems has triggered a dramatic proliferation of *cyber-physical attacks* [2], *i.e.*, security breaches in cyberspace to bring the physical plant into a state desired by the attacker. Some notorious examples are: (i) the *STUXnet* worm, which reprogrammed PLCs of nuclear centrifuges in Iran [3]; (ii) the *CRASHOVERRIDE* attack on the Ukrainian power grid, otherwise known as *Industroyer* [4]; (iii) the recent *TRITON/TRISIS* malware that targeted a petrochemical plant in Saudi Arabia [5].

Cyber-physical attacks tamper with both the cyber and the physical layer, as they may manipulate both the sensor measurements and the controller commands:

- *attacks on sensors* consist of reading and possibly replacing genuine sensor measurements with fake ones;
- *attacks on actuators* consist of reading, dropping and replacing controller commands with malicious ones.

According to [6], the possible *goals* of cyber-physical attacks can be classified into three main groups: (i) *equipment damage*, such as overstress of equipments, to reduce their expected life cycle, or violation of safety limits; (ii) *production damage*, in order to compromise product quality, product rate or operating costs; (iii) *compliance violations*, such as increasing of environmental pollution.

A clear understanding of both *attacker's goals* and the *severity of the damages* inflicted when such goals are achieved, is fundamental to conduct a *risk assessment* which ranks vulnerabilities that may be used by the attacker to achieve its goals. Motivated by the risk assessment application, the objective of this paper is to define formal *probabilistic metrics* to estimate the impact of cyber-physical attacks taking into account both (i) the *severity of the inflicted damage* in a given amount of time, and (ii) the *probability* that these attacks are actually accomplished, according to the dynamics of the CPS under attack. In doing so, we pay special attention to *stealthy attacks*, *i.e.*, attacks that cannot be detected by *intrusion detection systems* (IDSs) which typically monitor sensor signals, control commands, and network communications. In fact, to remain stealthy, attacks would have to closely follow the physical behaviour of the system, causing damage that can be negligible in the short term, but devastating in the long run.

Contribution: First of all, we introduce a formal language to specify both CPSs and cyber-physical attacks. For this very purpose, we resort to *process calculi*, a successful and widespread formal approach in *concurrency theory* for representing complex systems such as concurrent, distributed and mobile systems [7], [8], and used in many areas, including verification of security protocols [9] and security analysis of cyber-physical attacks [10].

In Section II, we propose a *hybrid probabilistic process calculus*, inspired by the calculi appeared in [10], [11], and with a clearly-defined *probabilistic behavioural semantics* [12]. In our calculus, cyber-physical systems are represented by making a clear distinction between the *physical component* (also called *plant*) describing the physical process, and the *logical component* describing the cyber devices (i.e., controllers, IDS, supervisors, etc.) governing the physical process as well as the interaction with other cyber devices. At the logical level, our calculus allows us to specify also (MITM) malicious activities targeting physical devices. In particular, we can model integrity attacks on both sensor measurements and actuator commands, as well as drops of actuator commands.

In this formal setting, we design two probabilistic metrics to estimate the impact of (possibly stealthy) cyber-physical attacks. As in Urbina et al. [13], we assume attacks with a complete knowledge of the system under attack, *i.e.*, they know the physical model, the goals in terms of damage that can be inflicted, and the thresholds selected to raise alerts via IDSs.

Our impact metrics are very general, and do not depend on the definition of our calculus, as they rely on the following notions associated to an arbitrary CPS equipped with an IDS:

- a *timed and probabilistic labelled transition semantics* [12] to formally describe the dynamics of both physical and

logical components of (possibly compromised) CPSs;
- a set $\mathcal{I}$ of *weighted attacker's goal indicators* whose runtime values denote how close is the attacker to reach each of these goals; we assume that IDSs know such indicators (and their associated severities) but they cannot access their values at runtime;
- a *detection policy* $\mathcal{P}$ that given an *alert signal* (raised by the IDS) and a goal indicator in $\mathcal{I}$ returns a *statically-determined* estimate of the progresses of the detected attack in achieving the goal denoted by that indicator.

Based on these concepts, in Section III we define two *probabilistic impact metrics*, $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ and $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$, to measure the *average false negatives* and the *average false positives*, respectively, faced by a CPS under attack, in the first n instants of time, according to a weighted attacker's goal indicators $\mathcal{I}$ and a detection policy $\mathcal{P}$ associated to the CPS.

Intuitively, given a (possibly compromised) CPS M, the values $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M) \in [0,1]$ and $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M) \in [0,1]$ represent the *average effectiveness* and the *average precision*, respectively, in detecting attacks that reach goals in $\mathcal{I}$, raising alert signals (via the IDS of M) which are interpreted by the detection policy $\mathcal{P}$ to estimate the severity of the damages inflicted by those attacks, in the first n time instants.

Thus, for instance, when $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M) = 0$ the detection policy is very effective in detecting all possible attacks achieving goals in $\mathcal{I}$. On the other hand, high values of $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M)$ tell us that the detection policy of M underestimates the progresses of the attacker in achieving the goals represented via the attacker's goal indicators $\mathcal{I}$ (high number of *false negatives*).

This metric does not say anything about the precision of the detection: there can be plenty of false positives, thus, apparently, it might seem that a certain attack has inflicted a severe damage, when actually that damage is negligible with respect to $\mathcal{I}$.

The average precision of the detection is measured using the metric $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$. Thus, for instance, when $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M) = 0$ the detection policy $\mathcal{P}$ is very accurate in signalling true attacks achieving goals in $\mathcal{I}$. Whereas, high values of $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$ tell us that the detection policy overestimates the progresses of the attacker in achieving the goals represented via the attacker's goal indicators $\mathcal{I}$ (high number of *false positives*).

True positive rates and *false positive rates* have been widely used in the literature to derive metrics, such as ROC curves, for classifying the accuracy of IDSs in terms of the intrusion detection capability, the Bayesian detection rate, and expected cost, which are all multi-criteria optimisation problems between these two rates [14]. In this paper, we take a different point of view. In fact, our metrics will not be used to compare IDSs, but instead for a formal estimation of the impact of (possibly stealthy) cyber-physical attacks, in terms of the physical inflicted damages represented via the attacker's goal indicators $\mathcal{I}$, and the logical signals raised by the IDS to detect attacks reaching such goals, according to the detection policy $\mathcal{P}$.

As a further contribution, we show that, under precise conditions, our metrics allows us to estimate the impact of cyber-physical attacks on a complex CPS in a *compositional manner*, *i.e.*, in terms of the impact on its sub-systems. In this respect, our process calculus supports features to compose CPSs in such a way as to avoid interference on physical and/or control variables. Basically, we build up composite CPSs putting together smaller sub-systems whose physical processes remain under the exclusive control of their associated logical components (controllers, IDSs, etc). Thus, interactions between logical components of different sub-systems are always possible, whereas interactions between physical processes may only occur if the corresponding logical components agree on when and how that should happen. Examples of this kind of CPSs can be found in several domains, such as: (i) manufacturing robotic applications (painting, welding, assembly, etc), in which different robotic arms are handled by different, but coordinated, controllers; (ii) drone swarms, where a fleet of drones explores and analyses physical perimeters; (iii) enrichment nuclear facilities, in which series of nuclear centrifuges work under the exclusive control of their programmable logic controllers.

In Section IV, we use our metrics to estimate the impact of four different, carefully chosen, cyber-physical attacks that target a non-trivial use case, consisting of two supervised self-coordinating refrigerated engine systems simulated in MATLAB [15]. Two out of four attacks are actually *stealthy attacks*.

In Section V, we draw conclusions and discuss related work.

II. A CALCULUS FOR CPSs AND ATTACKS

In this section, we define our *Probabilistic Calculus of Cyber-Physical Systems and Attacks*, called pCCPSA, a formal abstract language to specify both physical and logical components of CPSs, as well as cyber-physical attacks. The calculus is inspired by the process calculi appeared in [10], [11]. Unlike those calculi, for the sake of simplicity, we abstract away from physical devices (sensors and actuators) by allowing the cyber components to directly access *observable physical variables* and *control variables*. We also represent *unobservable physical variables*, that cannot be accessed by cyber components.

Let us start with some preliminary notations.

Notation 1. *We use* $v, w \in \mathsf{V}$ *for* variables*, partitioned in* observable physical variables $y, y_k \in \mathsf{Y}$*,* unobservable physical variables $x, x_k \in \mathsf{X}$*, and* control variables $a, a_k \in \mathsf{A}$*. Moreover we use* $c, d \in \mathsf{C}$ *for* communication channels*.* Values*, ranged over by* $m, m' \in \mathsf{M}$*, are built from basic values, such as integers and real numbers. Given a set of variables* V*, we write* $\mathbb{R}^{\mathsf{V}}$ *to denote the set of functions assigning a real value to each variable in* V*. For* $\xi \in \mathbb{R}^{\mathsf{V}}$*,* $v \in \mathsf{V}$ *and* $r \in \mathbb{R}$*, we write* $\xi[v \mapsto r]$ *to denote the function* $\psi \in \mathbb{R}^{\mathsf{V}}$ *such that* $\psi(w) = \xi(w)$*, for any* $w \neq v$*, and* $\psi(v) = r$*. Given* $\xi_1 \in \mathbb{R}^{\mathsf{V}_1}$ *and* $\xi_2 \in \mathbb{R}^{\mathsf{V}_2}$ *such that* $\mathsf{V}_1 \cap \mathsf{V}_2 = \emptyset$*, we write* $\xi_1 \uplus \xi_2$ *for the function in* $\mathbb{R}^{\mathsf{V}_1 \cup \mathsf{V}_2}$ *such that* $(\xi_1 \uplus \xi_2)(v) = \xi_1(v)$*, if* $v \in \mathsf{V}_1$*, and* $(\xi_1 \uplus \xi_2)(v) = \xi_2(v)$*, if* $v \in \mathsf{V}_2$*.*

As pCCPSA is a probabilistic process calculus, we provide the necessary mathematical machinery for its formal definition.

Definition 1. *A* discrete probability distribution *over a set of objects* O *is a function* $\gamma \colon \mathsf{O} \to [0,1]$ *with* $\sum_{o \in \mathsf{O}} \gamma(o) = 1$*. The* support *of* γ *is the set* $\mathsf{supp}(\gamma) = \{o \in \mathsf{O} \colon \gamma(o) > 0\}$*.*

With $\mathcal{D}(\mathsf{O})$ we denote the set of all finite-support probability distributions over O. For any $o \in \mathsf{O}$, the point distribution at *o, written $\overline{o}$, assigns probability 1 to o and 0 to all other objects.*

In our language, cyber-physical systems consist of two components: (i) the *physical component*, also called *physics*, describing the physical process, and (ii) the *logical component*, also called *logics*, that interacts with the physics and supports communications between logical sub-components. The physics has two sub-components: the *evolution law*, which describes the time-dependent evolution of the physical process, and the *state* of the system, which is supposed to change at runtime.

Definition 2 (Physics)**.** *Let V be a set of variables. The* physics *of a CPS defined on V is a pair $\langle \varepsilon, \xi_{\mathsf{V}} \rangle$, where:*

(a) ε denotes the evolution law *represented as a function of type $\mathbb{R}^{\mathsf{V}} \to \mathcal{D}(\mathbb{R}^{\mathsf{V}})$;*

(b) $\xi_{\mathsf{V}} \in \mathbb{R}^{\mathsf{V}}$ is the state function *recording the current* state *of the variables in V.*

A state function ξ_{V} returns the current value associated to each variable in V. For $\mathsf{V} = \mathsf{X} \cup \mathsf{Y} \cup \mathsf{A}$, we will abbreviate $(\xi_{\mathsf{V}})_{|\mathsf{X}}$ with ξ_{X}, $(\xi_{\mathsf{V}})_{|\mathsf{Y}}$ with ξ_{Y}, and $(\xi_{\mathsf{V}})_{|\mathsf{A}}$ with ξ_{A}. An evolution law ε models the evolution of the physical system, in which changes on the control variables A may reflect on both observable variables Y and unobservable ones X. Notice that, given a state ξ_{V}, $\varepsilon(\xi_{\mathsf{V}})$ returns a *probability distribution over all possible next states* to model the presence of *uncertainty* in our models.

Example 1 (Physics of a refrigerated engine system)**.** *We provide a running example (inspired from [10]), called Eng, in which the temperature of a simple engine is maintained within a specific range by means of a cooling system.*

The set of variables $\mathsf{V} = \mathsf{X} \cup \mathsf{Y} \cup \mathsf{A}$ of the system consists of: (i) an observable variable $temp \in \mathsf{Y}$ recording the temperature of the engine and with initial value 95; (ii) six unobservable variables $past_k \in \mathsf{X}$, for $1 \leq k \leq 6$, recording the temperatures in the last 6 time instants, and all initially set to 95; (iii) an unobservable variable $stress \in \mathsf{X}$ denoting the level of stress of the engine, due to its operating conditions, with initial value 0 and maximal value 1; (iv) a control variable $cool \in \mathsf{A}$ to turn on/off the cooling system, with initial value off; (v) a control variable $speed \in \mathsf{A}$ to set the engine speed, ranging over values in the set $\{\mathsf{slow}, \mathsf{half}, \mathsf{full}\}$; in normal conditions the engine runs at half power.

Thus, the state function ξ_{V} is defined as $\xi_{\mathsf{V}} = \xi_{\mathsf{X}} \uplus \xi_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}$, for $\mathsf{V} = \mathsf{X} \cup \mathsf{Y} \cup \mathsf{A}$, and $\mathsf{X} = \{past_1, \ldots, past_6, stress\}$, $\mathsf{Y} = \{temp\}$ and $\mathsf{A} = \{cool, speed\}$; for the sake of simplicity, we assume ξ_{A} to be a mapping where $\xi_{\mathsf{A}}(cool) \in \{\mathsf{on}, \mathsf{off}\}$ and $\xi_{\mathsf{A}}(speed) \in \{\mathsf{slow}, \mathsf{half}, \mathsf{full}\}$ such that $\xi_{\mathsf{A}}(cool) = \mathsf{off}$ when $\xi_{\mathsf{A}}(cool) \geq 0$, $\xi_{\mathsf{A}}(cool) = \mathsf{on}$ when $\xi_{\mathsf{A}}(cool) < 0$, $\xi_{\mathsf{A}}(speed) = \mathsf{slow}$ when $\xi_{\mathsf{A}}(speed) < 0$, $\xi_{\mathsf{A}}(speed) = \mathsf{half}$ when $\xi_{\mathsf{A}}(speed) = 0$ and $\xi_{\mathsf{A}}(speed) = \mathsf{full}$ when $\xi_{\mathsf{A}}(speed) > 0$.

The evolution law ε affects both the observable variables in Y and the unobservable variables in X as follows: (i) when the cooling system is active the temperature decreases by a value chosen in the real interval $[0.8, 1.2]$ according to a discrete probability distribution with granularity 10^{-1}; (ii) when the cooling system is inactive the temperature increases by a value m determined according to a discrete probability distribution with granularity 10^{-1}, and depending on the engine speed: thus, when the speed is slow the value m is chosen in the real interval $[0.1, 0.3]$, when the speed is half then m is chosen in the real interval $[0.4, 0.7]$, when the speed is full then m is chosen in the real interval $[0.8, 1.2]$; the reader may notice as the uncertainty of the temperature increase grows with the speed of the engine; (iii) the variables $past_k$, for $k \in 1 \ldots 6$, are updated to record the last six temperatures of the system; (iv) the variable $stress$ remains unchanged if the temperature was in the interval $[50, 100]$ for at least 3 of the last 6 time instants; otherwise, the variable is increased (reaching at most the value 1) by a constant $\mathsf{stress\_incr} = \frac{1}{50}$.

Formally, given a state $\xi_{\mathsf{V}} = \xi_{\mathsf{X}} \uplus \xi_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}$, the application $\varepsilon(\xi_{\mathsf{V}})$ returns a distribution γ such that, for any ξ'_{X} and ξ'_{Y} with $\gamma(\xi'_{\mathsf{X}} \uplus \xi'_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}) > 0$, it holds that:

- *If $\xi_{\mathsf{A}}(cool) = \mathsf{on}$ (active cooling), then $\gamma(\xi'_{\mathsf{X}} \uplus \xi'_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}) = \frac{1}{5}$ and $\xi'_{\mathsf{Y}}(temp) = \xi_{\mathsf{Y}}(temp) + m$, where $m \in \{-1 + k \cdot 10^{-1} : k \in \mathbb{Z} \wedge -2 \leq k \leq 2\}$;*
- *If $\xi_{\mathsf{A}}(cool) = \mathsf{off}$ (no cooling) and $\xi_{\mathsf{A}}(speed) = \mathsf{slow}$, then $\gamma(\xi'_{\mathsf{X}} \uplus \xi'_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}) = \frac{1}{3}$ and $\xi'_{\mathsf{Y}}(temp) = \xi_{\mathsf{Y}}(temp) + m$, where $m \in \{0.2 + k \cdot 10^{-1} : k \in \mathbb{Z} \wedge -1 \leq k \leq 1\}$;*
- *If $\xi_{\mathsf{A}}(cool) = \mathsf{off}$ (no cooling) and $\xi_{\mathsf{A}}(speed) = \mathsf{half}$, then $\gamma(\xi'_{\mathsf{X}} \uplus \xi'_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}) = \frac{1}{4}$ and $\xi'_{\mathsf{Y}}(temp) = \xi_{\mathsf{Y}}(temp) + m$, where $m \in \{0.5 + k \cdot 10^{-1} : k \in \mathbb{Z} \wedge -1 \leq k \leq 2\}$;*
- *If $\xi_{\mathsf{A}}(cool) = \mathsf{off}$ (no cooling) and $\xi_{\mathsf{A}}(speed) = \mathsf{full}$, then $\gamma(\xi'_{\mathsf{X}} \uplus \xi'_{\mathsf{Y}} \uplus \xi_{\mathsf{A}}) = \frac{1}{5}$ and $\xi'_{\mathsf{Y}}(temp) = \xi_{\mathsf{Y}}(temp) + m$, where $m \in \{1 + k \cdot 10^{-1} : k \in \mathbb{Z} \wedge -2 \leq k \leq 2\}$;*
- *$\xi'_{\mathsf{X}}(past_1) = \xi_{\mathsf{Y}}(temp)$ and $\xi'_{\mathsf{X}}(past_k) = \xi_{\mathsf{X}}(past_{k-1})$, for $2 \leq k \leq 6$;*
- *$\xi'_{\mathsf{X}}(stress) = \xi_{\mathsf{X}}(stress)$ if $|\{k : \xi_{\mathsf{X}}(past_k) \in [50, 100] \wedge 1 \leq k \leq 6\}| \geq 3$, and $\xi'_{\mathsf{X}}(stress) = \min(1, \xi_{\mathsf{X}}(stress) + \mathsf{stress\_incr})$, otherwise.*

As regards the formalisation of the *logics* of CPSs in pCCPSA, we build on Hennessy and Regan's *Timed Process Language* (TPL) [16]. We extend TPL with two constructs to read and write variables in $\mathsf{Y} \cup \mathsf{A}$. In this manner, we will model both honest and malicious activities on variables.

Definition 3 (Logics)**.** Logical processes *are defined as follows:*

$$
\begin{array}{rcl}
P, Q & ::= & \mathsf{nil} \mid \mathsf{tick}.P \mid P \parallel Q \mid \lfloor \pi.P \rfloor Q \\
 & \mid & [b]\{P\},\{Q\} \mid P \backslash c \mid X \mid \mathsf{rec}\, X.P \\
\pi & ::= & \mathsf{read}\, y(z) \mid \mathsf{write}\, a\langle m \rangle \mid \mathsf{readAll}\, (\tilde{z}) \\
 & \mid & \mathsf{read}\, a(z) \mid \mathsf{write}\, y\langle m \rangle \mid \mathsf{rcv}\, c(z) \mid \mathsf{snd}\, c\langle m \rangle
\end{array}
$$

We write nil for the *terminated process*. The process $\mathsf{tick}.P$ sleeps for one time unit and then continues as P. $P \parallel Q$ denotes the *parallel composition* of concurrent *threads* P and Q. The process $\lfloor \pi.P \rfloor Q$ denotes activities under timeout, meaning that either the activity π succeeds and the process evolves into P, in the current time unit, or the process timeouts and it evolves into Q, in the next time unit. Prefix $\mathsf{read}\, y(z)$, for $y \in \mathsf{Y}$, models a *read of the observable physical variable* y; whereas the prefix $\mathsf{write}\, a\langle m \rangle$, for $a \in \mathsf{A}$, represents a *write on the control variable* a. These prefixes are executed by the

controller. We also assume a prefix $\mathsf{readAll}\,(\tilde{z})$, executed by IDSs, to read at once all observable physical variables and control variables. Prefixes $\mathsf{rcv}\,c(z)$ and $\mathsf{snd}\,c\langle m\rangle$ model input and output activities on channel c, respectively. Furthermore, prefixes of the form $\mathsf{read}\,a(z)$ and $\mathsf{write}\,y\langle m\rangle$, with $a \in \mathsf{A}$ and $y \in \mathsf{Y}$, denote (MITM) *malicious activities* targeting physical devices. In particular, with $\mathsf{read}\,a(z)$ we model a *drop of an actuator command* associated to the control variable a; while $\mathsf{write}\,y\langle m\rangle$ represents an *integrity attack on the sensor* associated to the physical variable y.

Remark 1. *Notice that neither honest nor malicious code may access to unobservable physical variables in* X.

The process $[b]\{P\},\{Q\}$ is the standard conditional, where b is a decidable guard. The process $P\backslash c$ is the standard channel restriction operator. Sometimes we write $P\backslash\{c_1, c_2, \ldots, c_n\}$ to mean $P\backslash c_1\backslash c_2 \cdots \backslash c_n$. In processes of the form $\mathsf{tick}.Q$ and $\lfloor \pi.P \rfloor Q$, the occurrence of Q is said to be *time guarded*. The process $\mathsf{rec}\,X.P$ denotes (time-guarded) recursion. As further notation, we write $T\{^m/_v\}$ for the substitution of the variable v with the value m in any expression T. Similarly, in $T\{^P/_X\}$ the process variable X is replaced with the process P.

In the rest of the paper we use the following abbreviations.

Notation 2. *We write* $\pi.P$ *for the process* $\mathsf{rec}\,X.\lfloor \pi.P \rfloor X$, *where* X *does not occur in* P. *We write* $\mathsf{snd}\,c$ *and* $\mathsf{rcv}\,c$, *when channel* c *is used for pure synchronisation.*

Everything is in place to define our cyber-physical systems.

Definition 4 (Cyber-physical system)**.** *A* cyber-physical system *with variables* $\mathsf{V} = \mathsf{Y} \cup \mathsf{X} \cup \mathsf{A}$ *is given by two components:*

(a) a physical component $\langle \varepsilon, \xi_{\mathsf{V}} \rangle$*;*

(b) a logical component P *that may access only observable physical variables and control variables in* $\mathsf{Y} \cup \mathsf{A}$, *and can communicate, via channels, with other logical components.*

We write $\langle \varepsilon; \xi_{\mathsf{V}} \rangle \bowtie P$ *to denote the whole CPS, and use* M *and* N *to range over CPSs. Sometimes, when the evolution law* ε *is clearly identified, we write* $\xi_{\mathsf{V}} \bowtie P$ *instead of* $\langle \varepsilon; \xi_{\mathsf{V}} \rangle \bowtie P$. *CPSs of the form* $\xi_{\mathsf{V}} \bowtie P$ *are called* evolution-free. *Finally, we write* $\mathcal{M}$ *for the set of all possible CPSs in our calculus.*

Example 2 (A refrigerated engine system)**.** *Let us complete the formalisation of our running example Eng in* pCCPSA, *by defining its* logical component. *In Example 1, we already said that the temperature of our engine system is maintained by its controller, Ctrl, within a specific range by means of a cooling system. Furthermore, the system is equipped with an IDS that is in charge of checking whether the cooling system is active whenever the temperature is above a certain threshold. If this condition is violated then the IDS tries to* mitigate *the damages.*

The whole system is represented in pCCPSA *as follows:*

$$Eng \;=\; \langle \varepsilon; \xi_{\mathsf{V}} \rangle \bowtie (Ctrl \parallel IDS)\backslash\{ins\}$$

where the evolution law ε *and the state function* ξ_{V} *have been already defined in Example 1, while* ins *is a private channel for transmitting instructions on the speed of the engine.*

Thus, the logical component of the engine consists of two parallel processes: $Ctrl$ *that models the controller activity, and* IDS *that implements intrusion detection together with a mild form of* mitigation.

Intuitively, at each scan cycle the process $Ctrl$ *first checks the temperature and then waits for instructions to change the regime of the engine. When the sensed temperature is above* 100 *degrees, the controller activates the coolant; the cooling activity is maintained for* 5 *consecutive time instants. Otherwise, if the temperature is not above the threshold* 100 *then the controller turns off the cooling system. In both cases, before re-starting its scan cycle, the controller waits for instructions/requests to change the engine regime, coming from either its local IDS, via channel* ins, *or other IDSs, via channel* req_{in}. *In particular, if the controller receives instructions from its local IDS to slow down the engine (because of an anomalous overheating) then it will command so by changing* $speed$. *Otherwise, if the local IDS does not see any anomaly, while the IDS of another system is requesting the controller to work at full power, then the process* $Ctrl$ *will command so by acting on variable* $speed$.

The process IDS *checks whether the cooling system is active when the temperature is above* 100 *degrees. If this safety condition is violated then: (i) it signals at supervisory level the presence of an overheating anomaly at engine* ID, *via a channel* $warn$; *(ii) it sends the controller instructions to slow down the engine, via a channel* ins; *(iii) it requests to other engine systems to run at full power, via a channel* req_{out}, *to compensate the lack of performance of its own engine. Of course, such a request of compensation addressed to other engine systems may be accepted or not, depending whether the engine receiving the request is in condition to run at full power or not. However, if the IDS realises that the overheating anomaly is resolved then it asks (both local and external) controllers to reset their engines at half power. Formally,* $Ctrl$ *and* IDS *are defined as follows:*

$$\begin{aligned}
Ctrl &= \mathsf{rec}\,X.\mathsf{tick}.\mathsf{read}\;temp(z).[z > 100]\{Cooling\},\\
&\qquad\qquad \{\mathsf{write}\;cool\langle \mathsf{off}\rangle.Check\}\\
Cooling &= (\mathsf{write}\;cool\langle \mathsf{on}\rangle.\mathsf{tick})^5.Check\\
Check &= \mathsf{rcv}\;ins(z).\mathsf{rcv}\;req_{\mathrm{in}}(r).[z = \mathsf{slow}]\{\mathsf{write}\;speed\langle z\rangle.X\},\\
&\qquad\qquad \{\mathsf{write}\;speed\langle r\rangle.X\}\\
IDS &= \mathsf{rec}\,Y.\mathsf{readAll}\,(\tilde{z}).[z_{temp} > 100 \;\textit{and}\; z_{cool} = \mathsf{off}]\\
&\qquad \{\mathsf{snd}\;warn\langle \mathrm{ID}, \mathsf{hot}\rangle.\mathsf{snd}\;ins\langle \mathsf{slow}\rangle.\mathsf{snd}\;req_{\mathrm{out}}\langle \mathsf{full}\rangle.Y\},\\
&\qquad \{\mathsf{snd}\;warn\langle \mathrm{ID}, \mathsf{ok}\rangle.\mathsf{snd}\;ins\langle \mathsf{half}\rangle.\mathsf{snd}\;req_{\mathrm{out}}\langle \mathsf{half}\rangle.Y\}
\end{aligned}$$

In process $Cooling$, *with a small abuse of notation, we write* $(\ldots)^5.Check$ *to denote that the process between brackets is repeated* 5 *consecutive times.*

We remark that, for the sake of simplicity, our IDS is quite basic: for instance, it does not check whether the temperature is too low. However, it is straightforward to replace our IDS with a more sophisticated one, containing more informative tests on process variables and/or on control variables.

A. Composing cyber-physical systems

In Example 2, we have modelled an engine system that could be a component of a larger CPS. In this section, we describe how we can safely compose CPSs to build up larger systems

avoiding interferences on physical and/or control variables. The basic idea is the following: we can put together different CPSs provided that their physical components remain under the exclusive control of the corresponding logical components (controllers, IDSs, etc). Said in other words, in a composite system, any possible interaction between physical processes must be "filtered" by the associated logics.

Thus, we will say that two CPSs are *physically-disjoint* if their physics have unrelated variables, whereas their logics may share logical channels to support communications among them. This means that in physically-disjoint CPSs logical components may communicate with each other, whereas (indirect) interactions between physical processes may only occur if the associated logical components agree on when and how that should happen.

Definition 5 (Physically-disjoint CPSs)**.** *Let* $M_j = \langle \varepsilon_j ; \xi_{\mathsf{V}_j} \rangle \bowtie P_j$*, for* $j \in \{1,2\}$ *be two CPSs. We say that* M_1 *and* M_2 *are physically-disjoint if* $\mathsf{V}_1 \cap \mathsf{V}_2 = \emptyset$*. In this case, we write* $M_1 \uplus M_2$ *to denote the CPS defined as* $\langle (\varepsilon_1 \uplus \varepsilon_2) ; (\xi_{\mathsf{V}_1} \uplus \xi_{\mathsf{V}_2}) \rangle \bowtie (P_1 \parallel P_2)$*, where* $\varepsilon_1 \uplus \varepsilon_2$ *is the evolution law* ε *such that* $\varepsilon(\xi_{\mathsf{V}_1} \uplus \xi_{\mathsf{V}_2})(\xi'_{\mathsf{V}_1} \uplus \xi'_{\mathsf{V}_2}) = \varepsilon_1(\xi_{\mathsf{V}_1})(\xi'_{\mathsf{V}_1}) \cdot \varepsilon_2(\xi_{\mathsf{V}_2})(\xi'_{\mathsf{V}_2})$*. The generalisation* $M_1 \uplus \ldots \uplus M_n$ *to* n *CPSs is straightforward.*[1]

Another possible interference-free way of composing sub-systems is to allow interactions between a CPS M and a *pure-logical component* Q, that is a process acting only on communication channels and not on variables. This is the case of *supervisory components* interacting with lower-level controllers on (possibly private) communication channels.

Definition 6 (Parallel composition and restriction)**.** *Let* $M = \langle \varepsilon ; \xi_{\mathsf{V}} \rangle \bowtie P$*,* Q *a pure-logical process,* c *a channel. We write* $M \parallel Q$ *for* $\langle \varepsilon ; \xi_{\mathsf{V}} \rangle \bowtie (P \parallel Q)$*, and* $M \backslash c$ *for* $\langle \varepsilon ; \xi_{\mathsf{V}} \rangle \bowtie (P \backslash c)$*.*

Notice that in the composite system $M \parallel Q$, the process Q cannot interfere with the physical evolution of M, although it can definitely interact with M via communication channels, eventually affecting its physical behaviour.

Figure 1 provides an example of what kind of CPS architectures can be represented in pCCPSA by composing sub-systems via the three operators $\uplus$, $\parallel$ and $\backslash$.

Example 3 (A simple supervised self-coordinating engine system)**.** *We provide a simple* supervised self-coordinating engine system *consisting of two engines, a left engine,* Eng^{L}*, and a right one,* Eng^{R}*, interacting with a supervisory component that checks the correct functioning of the two engines. The composite system is defined as follows:*

$$Sys \;=\; \big((Eng^{\mathrm{L}} \uplus Eng^{\mathrm{R}}) \parallel Supervisor\big)\backslash\{warn\}$$

*where both the left and the right engines are properly re-labeled in order to: (i) embed the identities (*L *or* R*) of the engines, (ii) distinguish the corresponding variables of the two engines, (iii) fix the channels* req$\mathrm{L}$ *and* req$\mathrm{R}$ *for requesting full power of the engines* L *and* R*, respectively. Formally,*

[1]We refer the reader to Notation 1 for the definition of $\xi_{\mathsf{V}_1} \uplus \xi_{\mathsf{V}_2}$.

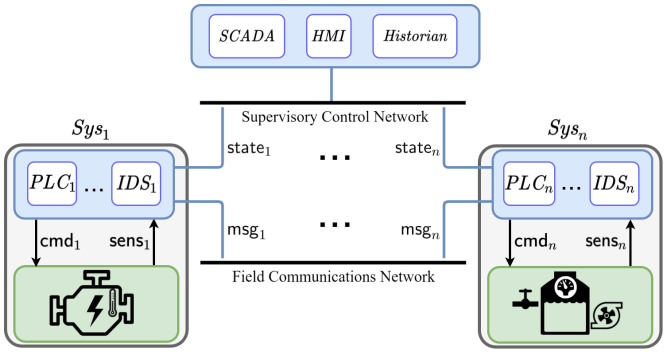

Figure 1. A CPS $\big((Sys_1 \uplus \ldots \uplus Sys_n) \parallel SCADA \parallel HMI \parallel Hist\big)\backslash\{state_i\}$

- $Eng^{\mathrm{L}} = Eng\ \{^{\mathrm{L}}/_{\mathrm{ID}}\}\{^{req\mathrm{L}}/_{req_{\mathrm{in}}}\}\{^{req\mathrm{R}}/_{req_{\mathrm{out}}}\}\{^{v\mathrm{L}}/_{v}\}$*, for any* $v \in \mathsf{V}$*;*
- $Eng^{\mathrm{R}} = Eng\ \{^{\mathrm{R}}/_{\mathrm{ID}}\}\{^{req\mathrm{R}}/_{req_{\mathrm{in}}}\}\{^{req\mathrm{L}}/_{req_{\mathrm{out}}}\}\{^{v\mathrm{R}}/_{v}\}$*, for any* $v \in \mathsf{V}$*.*

The pure-logical process Supervisor is defined as follows:

$$\begin{aligned}&\mathsf{rec}\ X.\mathsf{rcv}\ warn(u_1,z_1).\mathsf{rcv}\ warn(u_2,z_2).[z_1 = z_2 = \mathsf{ok}]\\ &\quad \{\mathsf{snd}\ okay.\mathsf{tick}.X\}, \{[z_1 = \mathsf{ok}]\{\mathsf{snd}\ fail\langle u_2, z_2\rangle.\mathsf{tick}.X\},\\ &\quad \{[z_2 = \mathsf{ok}]\{\mathsf{snd}\ fail\langle u_1, z_1\rangle.\mathsf{tick}.X\}, \{\mathsf{snd}\ alarm.\mathsf{tick}.X\}\}\}\end{aligned}$$

Intuitively, if both engines get in a warning state then an alarm signal *is sent, otherwise, if only one engine is facing a warning then the supervisor yields a* failure *to signal which engine is not working properly. Finally, if both engines are working fine then an* okay *signal is transmitted.*

B. Probabilistic labelled transition semantics

We start recalling the definition of *probabilistic labelled transition system* (pLTS) [12] over a generic set of terms.

Definition 7 (pLTS [12])**.** *A* probabilistic labelled transition system*, or* pLTS *for short, is a triple* $(\mathcal{T}, \mathbf{A}, \rightarrow)$*, where: (i)* $\mathcal{T}$ *is a countable set of* terms*, (ii)* $\mathbf{A}$ *is a countable set of* action labels*, and (iii)* $\rightarrow \subseteq \mathcal{T} \times \mathbf{A} \times \mathcal{D}(\mathcal{T})$ *is a* transition relation*.*

A classical LTS [17] is a special case of a pLTS where all transitions take to point distributions over terms.

Now, we provide the dynamics of pCCPSA in two steps: First, we give a pretty standard LTS for logical processes, then we build up a pLTS for CPSs by lifting transition rules from processes to CPSs, to deal with the probability distributions occurring in evolution laws.

In Table I, we provide the labelled transition semantics for logical components; here, transition rules have the form $P \xrightarrow{\lambda} Q$, where λ ranges over actions in the set $\{\mathsf{tick}, \tau, \overline{c}m, cm, v?m, v!m, (\tilde{z})\}$ denoting: passage of time, unobservable interaction, channel-based transmission/reception, read/write on observable physical variables and/or control variables, respectively, and supervisory read of all variables in $\mathsf{Y} \cup \mathsf{A}$. Rules (Rcv), (Send) and (Comm) model standard channel communication on some channel c. Rule (Read), for $v \in \mathsf{Y}$, models the read of the observable physical variable v; whereas (Write), for $v \in \mathsf{A}$, represents the write on the control variable v. However, for $v \in \mathsf{A}$, rule (Read) models a malicious attempt of *dropping an actuator command* acting on the control variable v. Similarly, when $v \in \mathsf{Y}$, rule (Write) expresses an *integrity attack*

Table I
LTS FOR LOGICAL PROCESSES

$$\text{(Rcv)}\ \frac{-}{\lfloor \mathsf{rcv}\, c(z).P \rfloor Q \xrightarrow{cm} P\{^m/_z\}} \qquad \text{(Snd)}\ \frac{-}{\lfloor \mathsf{snd}\, c\langle m\rangle.P \rfloor Q \xrightarrow{\overline{c}m} P} \qquad \text{(Comm)}\ \frac{P \xrightarrow{\overline{c}m} P' \quad Q \xrightarrow{cm} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'}$$

$$\text{(Read)}\ \frac{v \in \mathsf{Y} \cup \mathsf{A}}{\lfloor \mathsf{read}\, v(z).P \rfloor Q \xrightarrow{v?m} P\{^m/_z\}} \qquad \text{(Write)}\ \frac{v \in \mathsf{Y} \cup \mathsf{A}}{\lfloor \mathsf{write}\, v\langle m\rangle.P \rfloor Q \xrightarrow{v!m} P} \qquad \text{(Attack)}\ \frac{P \xrightarrow{v?m} P' \quad Q \xrightarrow{v!m} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'}$$

$$\text{(ReadAll)}\ \frac{-}{\lfloor \mathsf{readAll}\, (\tilde{z}).P \rfloor Q \xrightarrow{(\tilde{z})} P} \qquad \text{(TimeNil)}\ \frac{-}{\mathsf{nil} \xrightarrow{\mathsf{tick}} \mathsf{nil}} \qquad \text{(Sleep)}\ \frac{-}{\mathsf{tick}.P \xrightarrow{\mathsf{tick}} P} \qquad \text{(TimeOut)}\ \frac{-}{\lfloor \pi.P \rfloor Q \xrightarrow{\mathsf{tick}} Q}$$

$$\text{(Then)}\ \frac{[\![b]\!] = \mathsf{true} \quad P \xrightarrow{\lambda} P'}{[b]\{P\},\{Q\} \xrightarrow{\lambda} P'} \qquad \text{(Else)}\ \frac{[\![b]\!] = \mathsf{false} \quad Q \xrightarrow{\lambda} Q'}{[b]\{P\},\{Q\} \xrightarrow{\lambda} Q'} \qquad \text{(Res)}\ \frac{P \xrightarrow{\lambda} P' \quad \lambda \notin \{cv, \overline{c}v\}}{P\backslash c \xrightarrow{\lambda} P'\backslash c}$$

$$\text{(Rec)}\ \frac{P\{^{\mathsf{rec}\, X.P}/_X\} \xrightarrow{\lambda} P'}{\mathsf{rec}\, X.P \xrightarrow{\lambda} P'} \qquad \text{(Par)}\ \frac{P \xrightarrow{\lambda} P' \quad \lambda \neq \mathsf{tick}}{P \parallel Q \xrightarrow{\lambda} P' \parallel Q} \qquad \text{(TimePar)}\ \frac{P_1 \xrightarrow{\mathsf{tick}} P_1' \quad P_2 \xrightarrow{\mathsf{tick}} P_2'}{P_1 \parallel P_2 \xrightarrow{\mathsf{tick}} P_1' \parallel P_2'}$$

on (the sensors which reads) the physical variable v. Thus, in rule (Attack), if P is a malicious agent then we model the successful dropping of an actuator command emitted by Q; whereas if the malicious agent is Q then the same rule denotes the transmission of a fake value to the agent P reading the observable physical variable v. Rule (ReadAll) denotes the reading of all observable physical variables and control variables. Rules (TimeNil), (Sleep) and (TimeOut) model the passage of (discrete) time. Rules (Then), (Else), (Res), (Rec) are standard. Rule (Par) propagates untimed actions over parallel components, whereas rule (TimePar) does the same for timed actions. We omit the symmetric counterparts of (Comm), (Attack) and (Par).

In Table II, we lift the transition rules from logical processes to evolution-free CPSs, by relying on the following notations.

Notation 3. *We adopt the following notation for probability distributions: given a distribution σ over states and a process P, we write $\sigma \bowtie P$ to denote the distribution over (evolution-free) CPSs defined as $(\sigma \bowtie P)(\xi_{\mathsf{V}} \bowtie P') = \sigma(\xi_{\mathsf{V}})$, if $P = P'$, and $(\sigma \bowtie P)(\xi_{\mathsf{V}} \bowtie P') = 0$, otherwise. Moreover, given an evolution law ε, we write $\langle \varepsilon; \sigma \rangle \bowtie P$ to extend $\sigma \bowtie P$ to full CPSs as follows: $(\langle \varepsilon; \sigma \rangle \bowtie P)(\langle \varepsilon'; \xi_{\mathsf{V}} \rangle \bowtie P') = (\sigma \bowtie P)(\xi_{\mathsf{V}} \bowtie P')$, if $\varepsilon = \varepsilon'$, and $(\langle \varepsilon; \sigma \rangle \bowtie P)(\langle \varepsilon'; \xi_{\mathsf{V}} \rangle \bowtie P') = 0$, otherwise.*

Transition rules have the form $M \xrightarrow{\alpha} \gamma$, where M is an evolution-free CPS and γ is a distribution over (evolution-free) CPSs. For simplicity, as evolution laws contain only static information, the resulting transition rules are parameterised on an evolution law. Thus, Table II provides the transitions rules for evolution-free CPSs.

In Table II, actions, ranged over by α, are in the set $\mathbf{A} = \{\tau, \overline{c}m, cm, \mathsf{tick}\}$. These actions denote: internal activities (τ), channel transmission ($\overline{c}m$ and cm), and the passage of time (tick). Rules (Receive) and (Send) model reception and transmission on a channel c, respectively. Rule (Tau) lifts silent actions from processes to systems. This includes channel communications and attacks addressed to both physical and control variables. Rule (Read) models the reading of the observable physical variable y. Notice that the presence of an attack capable of supplying a fake value (via an action $y!m$) may prevent the access to the correct value by the controller (see rule (Attack) in Table I). Rule (Write) models the writing of a value m on an control variable a; here, the presence of an attacker capable of performing a drop action $a?m$ may prevent the access to the variable (see, again, rule (Attack) in Table I). Rule (ReadIDS) models a read of the current state of all observable physical variables and control variables, this action requires the passage of time as in the next rule, that we explain in more detail. In Rule (Time) timed actions are lifted from logical processes to CPSs; here $\varepsilon(\xi_{\mathsf{V}})$ returns a probability distribution over possible state functions for the next time slot, according to the current state ξ_{V} and the evolution law ε.

Notice that in our pLTS we defined transitions rules of the form $\xi_{\mathsf{V}} \bowtie P \xrightarrow{\alpha} \sigma \bowtie P'$, parametric on some evolution law ε. As evolution laws do not change at runtime, $\xi_{\mathsf{V}} \bowtie P \xrightarrow{\alpha} \sigma \bowtie P'$ entails $\langle \varepsilon; \xi_{\mathsf{V}} \rangle \bowtie P \xrightarrow{\alpha} \langle \varepsilon; \sigma \rangle \bowtie P'$, thus providing the pLTS for (full) CPSs.

Now, having defined the labelled transitions that can be performed by a CPS of the form $\langle \varepsilon; \xi_{\mathsf{V}} \rangle \bowtie P$, we can easily concatenate these transitions to define the possible *execution traces* of a CPS $M \in \mathcal{M}$. Formally,

Definition 8 (Traces, derivates and channel communications)**.** *We write $M \xrightarrow{\alpha} M'$ if there exists a transition $M \xrightarrow{\alpha} \gamma$ such that $M' \in \mathsf{supp}(\gamma)$. Let $t = \alpha_1 \dots \alpha_n$, for $n \geq 0$, a (possibly empty) sequence of actions $\alpha_i \in \mathbf{A} \setminus \{\tau\}$. A* trace *$M \stackrel{t}{\Rightarrow} M'$ is a (possibly empty) sequence of transitions $M = M_1(\xrightarrow{\tau})^* \xrightarrow{\alpha_1} (\xrightarrow{\tau})^* \dots (\xrightarrow{\tau})^* \xrightarrow{\alpha_n} (\xrightarrow{\tau})^* M_{n+1} = M'$. Given a CPS $M \in \mathcal{M}$ we write $\mathsf{der}(M)$ to denote the set $\{M' \in \mathcal{M} : \exists t \text{ s.t. } M \stackrel{t}{\Rightarrow} M'\}$ of the* derivatives *of M. Moreover, we write $\mathsf{out}(M)$ to denote the set $\{\overline{c}m \in \mathbf{A} : \exists M' \in \mathsf{der}(M) \text{ s.t. } M' \xrightarrow{\overline{c}m} \gamma\}$ of the* signals *coming from the (derivatives of) M, and $\mathsf{inp}(M)$ to denote the set $\{cm \in \mathbf{A} : \exists M' \in \mathsf{der}(M) \text{ s.t. } M' \xrightarrow{cm} \gamma\}$ of the possible* channel receptions *of the (derivatives of) M.*

Remark 2 (Absence of zeno behaviours)**.** *Since recursion is always timed guarded and all timed transitions of CPSs are*

Table II
pLTS FOR CPSs $\xi_{\mathsf{V}} \bowtie P$ PARAMETRIC ON THE EVOLUTION LAW ε

$$\text{(Receive)}\ \frac{P \xrightarrow{cm} P'}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{cm} \overline{\xi_{\mathsf{V}}} \bowtie P'} \qquad \text{(Send)}\ \frac{P \xrightarrow{\overline{c}m} P'}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{\overline{c}m} \overline{\xi_{\mathsf{V}}} \bowtie P'} \qquad \text{(Tau)}\ \frac{P \xrightarrow{\tau} P'}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{\tau} \overline{\xi_{\mathsf{V}}} \bowtie P'}$$

$$\text{(Read)}\ \frac{\mathsf{V} = \mathsf{X} \cup \mathsf{Y} \cup \mathsf{A} \quad y \in \mathsf{Y} \quad P \xrightarrow{y?m} P' \quad \xi_{\mathsf{V}}(y) = m}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{\tau} \overline{\xi_{\mathsf{V}}} \bowtie P'} \qquad \text{(Write)}\ \frac{\mathsf{V} = \mathsf{X} \cup \mathsf{Y} \cup \mathsf{A} \quad a \in \mathsf{A} \quad P \xrightarrow{a!m} P'}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{\tau} \overline{\xi_{\mathsf{V}}[a \mapsto m]} \bowtie P'}$$

$$\text{(ReadIDS)}\ \frac{P \xrightarrow{(\tilde{z})} P' \quad P \xrightarrow{\mathsf{tick}} \quad P \not\xrightarrow{y?m} \quad P \not\xrightarrow{a!m} \quad \widetilde{m} = \xi_{\mathsf{V}}(\mathsf{Y} \cup \mathsf{A})}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{\mathsf{tick}} \varepsilon(\xi_{\mathsf{V}}) \bowtie P'\{^{\widetilde{m}}/_{\tilde{z}}\}} \qquad \text{(Time)}\ \frac{P \xrightarrow{\mathsf{tick}} P' \quad P \not\xrightarrow{\alpha} \quad \alpha \in \{(\tilde{z}),\, y?m,\, a!m\}}{\xi_{\mathsf{V}} \bowtie P \xrightarrow{\mathsf{tick}} \varepsilon(\xi_{\mathsf{V}}) \bowtie P'}$$

inherited from timed transitions of their logical components (rules (ReadIDS) and (Time)), it follows that the number of untimed transitions between two timed ones is always bounded.

III. TWO PROBABILISTIC IMPACT METRICS

As said in the Introduction, the two impact metrics $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ and $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$ that we are going to define will be based on the following concepts: (a) a *timed and probabilistic labelled transition semantics* [12] to formally describe the probabilistic dynamics of (possibly compromised) CPSs; (b) a set $\mathcal{I}$ of *weighted attacker's goal indicators* whose runtime values denote how close is the attacker to reach each of these goals; (c) a *detection policy* $\mathcal{P}$ that given an alert signal $\alpha \in \mathbf{A}$ (raised by the IDS) and an attacker's goal indicator in $\mathcal{I}$ returns a *statically-determined* estimate on the progresses of the detected attack in achieving that goal.

In Table II, we already provided a pLTS to formalise the semantics of CPSs and attacks. Now, let us formally define the two parameters $\mathcal{I}$ and $\mathcal{P}$ for a given CPS $M \in \mathcal{M}$.

The set of *weighted attacker's goal indicators* is defined as $\mathcal{I} = \{(i_1, r_1), \ldots, (i_k, r_k)\}$, where i_j is a function of type $\mathsf{der}(M) \to [0,1]$ and $r_j \in [0,1]$, for $1 \leq j \leq k$. Each pair $(i, r) \in \mathcal{I}$ focuses on some specific *attacker's goal* to compromise the CPS M under examination and all its derivatives in $\mathsf{der}(M)$. Intuitively, for any $M' \in \mathsf{der}(M)$, $i(M')$ returns a *dynamic quantification* of how close is the attacker injected in M' to reach the goal i, whereas the weight r provides us with a *static information* about the damage inflicted by an attack achieving the goal represented by the indicator i. Thus, when $i(M') = 0$ the physical state of M' is uncorrupted with respect to the attacker's goal indicator i (*e.g.*, the runtime values of the *stress* variable of the system of Example 2 are close to 0), while $i(M') = 1$ denotes full corruption of the physical state with respect to the goal indicator i (*e.g.*, our *stress* variable reaches values close to 1); intermediate values tell us to which extent the physical state of the system M' is corrupted. In the following, we will write $\mathcal{I}^- = \{i_1, \ldots, i_k\}$ to denote the set of (pure) attacker's goal indicators.

The *detection policy* $\mathcal{P} \colon \mathsf{out}(M) \times \mathcal{I}^- \to [0,1]$ *statically* associates alert signals $\alpha \in \mathsf{out}(M)$ and goal indicators $i \in \mathcal{I}^-$ to weights that denote an estimate of the progresses of the detected attack, associated to the signal α, in achieving the goal with indicator i. Thus, for example, when $\mathcal{P}(\alpha, i) = 1$ the presence at runtime of the alert signal α says to system engineers that the attacker's goal denoted by the indicator i has been fully achieved by the attacker; whereas, when $\mathcal{P}(\alpha, i) = 0$ the attack detected via the alert signal α has not done any progress in the achievement of the goal represented by i. In case of intermediate values, the presence of the signal α says to which extent the attacker's goal i has been achieved.

As an example, let us define the weighted attacker's goal indicators and the detection policy for the CPS of Example 2.

Example 4 (Attacker's goal indicators and detection policy for *Eng*)**.** *The set of weighted attacker's goal indicators for Eng is defined as follows:* $\mathcal{I}_{\text{E}} = \{(stress, 1)\}$*. This means that the attacker's goal indicator stress is a critical indicator whose variations denote critical attacks. Here, the only attacker's goal indicator coincides with the unobservable physical variable stress.*[2] *The detection policy* $\mathcal{P}_{\text{E}} : \mathsf{out}(Eng) \times \mathcal{I}^-_{\text{E}} \to [0,1]$ *is defined as follows:* $\mathcal{P}_{\text{E}}(\overline{warn}\langle \text{ID}, \mathsf{hot}\rangle, stress) = 1$ *and* $\mathcal{P}_{\text{E}}(\alpha, i) = 0$*, otherwise. This means that the presence of the signal* $\overline{warn}\langle \text{ID}, \mathsf{hot}\rangle$ *emitted by the IDS provides an estimate that the attacker fully achieved the goal of affecting the stress of the engine system. On the other hand, transmissions at communications channel* req_{out} *are considered irrelevant from the point of view of the attacker's goal indicators* $\mathcal{I}^-_{\text{E}}$*.*

In order to define our impact metrics $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ and $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$, we define a new probabilistic transition $M \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}} \gamma$ summing up "attacker's progresses" associated to execution traces spanning over a single time slot, according to the detection policy $\mathcal{P}$. Intuitively, the weights $q_j \in [0,1]$, for $j \in \{1, \ldots, k\}$, sums up estimated progresses achieved by the attacker in one time slot with respect to each attacker's goal indicators i_j, according to the alert signals emitted by the IDS. Formally,

Definition 9. *Let $M \in \mathcal{M}$ be a CPS with an associated set of weighted attacker's goal indicators $\mathcal{I}$, such that $\mathcal{I}^- = \{i_1, \ldots, i_k\}$, and detection policy $\mathcal{P}$. We write $M \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}} \gamma$ if $M \overset{t}{\Rightarrow} M'$, for a sequence of untimed actions $t = \alpha_1 \ldots \alpha_n$,[3] with $\alpha_1, \ldots, \alpha_n \in \mathbf{A} \backslash \{\tau, \mathsf{tick}\}$, such that $M' \xrightarrow{\mathsf{tick}} \gamma$ and $q_j = \min(1, \sum_{l=1, \alpha_l \in \mathsf{out}(M)}^{n} \mathcal{P}(\alpha_l,\, i_j))$, for $j \in \{1, \ldots, k\}$.*[4]

[2] We recall that physical variables can be seen as functions that given the current state of the CPS return the current value of the variable.

[3] We recall that untimed actions always lead to point distributions.

[4] By Remark 2 the number of untimed actions preceding a tick is always finite.

Table III
DEFINITIONS OF THE METRICS $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M)$ AND $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$ FOR $n > 0$.

$$\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M) = \min\left(1,\ \max_{M \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}} \gamma} \frac{\sum_{j=1}^{k} r_j \cdot \max(0\,, i_j(M) - q_j) + (n-1) \cdot \sum_{M' \in \mathsf{supp}(\gamma)} \gamma(M') \cdot \mathbf{FN}^{n-1}_{\mathcal{I},\mathcal{P}}(M')}{n}\right)$$

$$\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M) = \min\left(1,\ \max_{M \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}} \gamma} \frac{\sum_{j=1}^{k} r_j \cdot \max(0\,, q_j - i_j(M)) + (n-1) \cdot \sum_{M' \in \mathsf{supp}(\gamma)} \gamma(M') \cdot \mathbf{FP}^{n-1}_{\mathcal{I},\mathcal{P}}(M')}{n}\right)$$

Now, we are ready to define the two metrics $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}} \colon \mathcal{M} \to [0,1]$ and $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}} \colon \mathcal{M} \to [0,1]$ to measure the *average effectiveness* and the *average precision*, respectively, of the detection mechanism of a (possibly compromised) CPS $M \in \mathcal{M}$ in recognising attacks achieving goals in $\mathcal{I}$, by raising alert signals (via the IDS) that are interpreted by the detection policy $\mathcal{P}$ to estimate the severity of the damages inflicted by those attacks, in the first n time instants.

Basically, for each (probabilistic) execution trace leading from a system M to a system M' in at most n time instants, the metric $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ measures the *effectiveness* of the detection by computing the average (weighted) deviation between the actual damage inflicted by attackers with goals indicators $i_j \in \mathcal{I}^-$, and the estimate returned by the policy $\mathcal{P}$ of those damages. Such a deviation is weighted using both the severity r_j of the damage associated to i_j and the probability to reach M' from M. Thus, for example, high values of $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M)$ tell us that the detection mechanism underestimates the progresses of the attacker in achieving the attacker's goal indicators represented in $\mathcal{I}$ (high number of *false negatives*). The definition of $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ is by induction on the time $n \in \mathbb{N}$. Precisely, $\mathbf{FN}^0_{\mathcal{I},\mathcal{P}}(M) = 0$, while, for $n > 0$, the definition is given in Table III and relies on all derivatives M' of M, reached in at most n time instants.

As for the metric $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$, for each (probabilistic) execution trace leading from a system M to a system M' in at most n time instants, $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$ measures the *precision* of the detection by computing the average (weighted) deviation between the estimate provided by the policy $\mathcal{P}$ of the damage inflicted by attackers with goals in $\mathcal{I}$, and the actual damage achieved by the attackers, according to each weighted attacker's goal indicator $(i_j, r_j) \in \mathcal{I}$. Such a deviation is weighted using both the severity r_j of the damage associated to the indicator i_j and the probability to reach M' from M. Thus, for example, high values of $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$ tell us that the detection mechanism overestimates the progresses of the attacker in achieving the attacker's goal indicators represented in $\mathcal{I}$ (high number of *false positives*). The definition of $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$ is by induction on the time $n \in \mathbb{N}$. Precisely, $\mathbf{FP}^0_{\mathcal{I},\mathcal{P}}(M) = 0$, while, for $n > 0$, the definition is given in Table III and relies on all derivatives of M, reached in at most n time instants.

As an example, we can apply our metrics to formally state that the detection policy of the engine system *Eng* of Example 2 is effective and precise, i.e., there are neither false negatives nor false positives, with respect to the attacker's goal indicators $\mathcal{I}_{\mathrm{E}}$ and the detection policy $\mathcal{P}_{\mathrm{E}}$, as defined in Example 4.

Proposition 1. $\mathbf{FN}^n_{\mathcal{I}_{\mathrm{E}},\mathcal{P}_{\mathrm{E}}}(Eng) = 0$ *and* $\mathbf{FP}^n_{\mathcal{I}_{\mathrm{E}},\mathcal{P}_{\mathrm{E}}}(Eng) = 0$.

We conclude this section with a remark on how our metrics deal with nondeterministic behaviours.

Remark 3 (Worst-case approach)**.** *Although both metrics work probabilistically, when dealing with nondeterministic behaviours of the compromised system (*e.g.*, a possible drop of an actuator command achieving one of the goals in $\mathcal{I}$) the metrics adopt a worst-case approach. More precisely, the metric $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ chooses the behaviour of the system that most underestimate the attacker's progress; whereas the metric $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$ chooses the behaviour that most overestimate the attacker's progress.*

A. Composing the impacts on CPS sub-systems

In Section II-A, we described how to compose larger CPSs putting together physically-disjoint components and pure-logical ones. In this section, we describe how to compute the impact of an attack on a composite CPS in terms of the impacts on its sub-systems. In doing so, we have to understand how to compose the attacker's goal indicators and the detection policies of sub-components. For instance, when putting together two physically-disjoint components M_1 and M_2 we must define how to compose their corresponding attacker's goal indicators $\mathcal{I}_1$ and $\mathcal{I}_2$ and the detection policies $\mathcal{P}_1$ and $\mathcal{P}_2$, according to some notion of *security clearance* associated to the each component. Intuitively, the security clearance of a physically-disjoint CPS component should denote the importance of that component from a security point of view (*e.g.*, in an airplane, securing the engine control system may be more critical than securing the air conditioning system).

Definition 10 (Distributing $\mathcal{I}$ and $\mathcal{P}$ over $\uplus$)**.** *Let M_1 and M_2 be two physically-disjoint CPSs, with disjoint weighted attacker's goal indicators $\mathcal{I}_1$ and $\mathcal{I}_2$, and detection policies $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Let $p_1, p_2 \in [0,1]$ be the* security clearances *associated to M_1 and M_2, respectively. Then, the CPS $M_1 \uplus M_2$ has attacker's goal indicators $p_1 \cdot \mathcal{I}_1 \uplus p_2 \cdot \mathcal{I}_2$, and detection policy $p_1 \cdot \mathcal{P}_1 \uplus p_2 \cdot \mathcal{P}_2$, where:*

- $p_1 \cdot \mathcal{I}_1 \uplus p_2 \cdot \mathcal{I}_2 = \bigcup_{(i,r) \in \mathcal{I}_1} (i, p_1 \cdot r) \cup \bigcup_{(i,r) \in \mathcal{I}_2} (i, p_2 \cdot r)$*;*
- *$p_1 \cdot \mathcal{P}_1 \uplus p_2 \cdot \mathcal{P}_2$ is such that $(p_1 \cdot \mathcal{P}_1 \uplus p_2 \cdot \mathcal{P}_2)(\alpha, i) = \mathcal{P}_1(\alpha, i)$, if $i \in \mathcal{I}_1^-$, and $(p_1 \cdot \mathcal{P}_1 \uplus p_2 \cdot \mathcal{P}_2)(\alpha, i) = \mathcal{P}_2(\alpha, i)$, if $i \in \mathcal{I}_2^-$.*

The generalisation to CPSs as $M_1 \uplus \ldots \uplus M_n$ is straightforward.

A similar definition is necessary for both channel restriction and parallel composition between CPSs and pure-logical processes (we recall that we use pure-logical processes to represent *supervisory components* interacting with lower-level controllers

on communication channels). Note that we have not associated yet attacker's goal indicators and detection policies to pure-logical processes. However, as in a system $M \parallel Q$ the pure-logical process Q may communicate with M, it is reasonable to assume that it may emit signals about the indicators of M.

Definition 11 (Distributing $\mathcal{I}$ and $\mathcal{P}$ over $\parallel$ and restriction)**.** *Let $M \in \mathcal{M}$ be a CPS with attacker's goal indicators $\mathcal{I}$ and detection policy $\mathcal{P}$. Let Q be a pure-logical process with the following detection policy $\mathcal{P}_Q \colon \mathsf{out}(Q) \times \mathcal{I}^- \to [0,1]$, wrt $\mathcal{I}$, such that for any $(\alpha, i) \in \mathsf{dom}(\mathcal{P}) \cap \mathsf{dom}(\mathcal{P}_Q)$ it holds that $\mathcal{P}(\alpha, i) = \mathcal{P}_Q(\alpha, i)$. Let c be a channel. Then,*

- *$M \parallel Q$ has weighted attacker's goal indicators $\mathcal{I}$ and detection policy $\mathcal{P} \parallel \mathcal{P}_Q$, where $(\mathcal{P} \parallel \mathcal{P}_Q)(\alpha, i) = \mathcal{P}(\alpha, i)$, if $(\alpha, i) \in \mathsf{dom}(\mathcal{P})$; $(\mathcal{P} \parallel \mathcal{P}_Q)(\alpha, i) = \mathcal{P}_Q(\alpha, i)$, otherwise, and*
- *$M \backslash c$ has weighted attacker's goal indicators $\mathcal{I}$ and detection policy $(\mathcal{P} \backslash c)$, where $(\mathcal{P} \backslash c)(\alpha, i) = \mathcal{P}(\alpha, i)$ if c does not occur in α, and $(\mathcal{P} \backslash c)(\alpha, i) = 0$, otherwise.*

Example 5 (Attacker's goal indicators and detection policy for the system of Example 3)**.** *Let's define the attacker's goal indicators $\mathcal{I}_{Sys}$ and the detection policy $\mathcal{P}_{Sys}$ for the composite system Sys of Example 3. We derive them from the components Eng^{L}, Eng^{R} and Supervisor (for brevity, here, we will call it Super). The security clearances of the two engine systems are: $p_1 = p_2 = 0.7$ (we assume equal criticality). Thus, $\mathcal{I}_{Sys} = 0.7 \cdot \mathcal{I}_{\mathrm{L}} \uplus 0.7 \cdot \mathcal{I}_{\mathrm{R}}$ and $\mathcal{P}_{Sys} = ((0.7 \cdot \mathcal{P}_{\mathrm{L}} \uplus 0.7 \cdot \mathcal{P}_{\mathrm{R}}) \parallel \mathcal{P}_{Super}) \setminus \{warn\}$, where: (i) the goal indicators $\mathcal{I}_{\mathrm{L}}$ and $\mathcal{I}_{\mathrm{R}}$ for Eng^{L} and Eng^{R}, respectively, are obtained from the goal indicators $\mathcal{I}_{\mathrm{E}}$ in Example 4, in the obvious manner; (ii) the same holds for the two policies $\mathcal{P}_{\mathrm{L}}$ and $\mathcal{P}_{\mathrm{R}}$; (iii) the detection policy $\mathcal{P}_{Super}$ is defined as follows:*

- *$\mathcal{P}_{Super}(\overline{alarm}, stress\mathrm{L}) = \mathcal{P}_{Super}(\overline{alarm}, stress\mathrm{R}) = 1$;*
- *$\mathcal{P}_{Super}(\overline{fail}\langle \mathrm{L}, \cdot \rangle, stress\mathrm{L}) = \mathcal{P}_{Super}(\overline{fail}\langle \mathrm{R}, \cdot \rangle, stress\mathrm{R}) = 1$;*
- *$\mathcal{P}_{Super}(\overline{fail}\langle \mathrm{L}, \cdot \rangle, stress\mathrm{R}) = \mathcal{P}_{Super}(\overline{fail}\langle \mathrm{R}, \cdot \rangle, stress\mathrm{L}) = 0$;*
- *$\mathcal{P}_{Super}(\overline{okay}, stress\mathrm{L}) = \mathcal{P}_{Super}(\overline{okay}, stress\mathrm{R}) = 0$;*
- *$\mathcal{P}_{Super}(\alpha, i) = 0$, otherwise.*

Now, everything is in place to formally state in which terms our impact metrics are compositional with respect to the notion of compositionality given in Definitions 5 and 6 of Section II-A.

We start by computing our metrics on composite systems consisting of physically-disjoint CPSs, under the hypothesis that channel communications among those CPSs do not carry any information on the achievement of attacker's goal indicators, unlike alert signals that are addressed to the supervisory level.

Basically, the numeric values of the impact metrics of a composite system of the form $M_1 \uplus \ldots \uplus M_k$ are obtained through a weighted sum of the impacts on the single components.

Theorem 1 (Compositionality w.r.t. $\uplus$)**.** *Let $M_i \in \mathcal{M}$, for $i \in \{1, \ldots, k\}$, be k physically-disjoint CPSs, with security clearances $p_i \in [0,1]$, disjoint attacker's goal indicators $\mathcal{I}_i$, and detection policies $\mathcal{P}_i$, respectively, such that for any $i, j \in \{1, \ldots, k\}$, $i \neq j$, whenever $\overline{c}m \in \mathsf{out}(M_i)$ and $cm \in \mathsf{inp}(M_j)$ it holds that $\mathcal{P}_i(\overline{c}m, h) = 0$, for any $h \in \mathcal{I}_i^-$. Then,*

1) *$\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M_1 \uplus \ldots \uplus M_k) = \min(1, p_1 \cdot \mathbf{FN}^n_{\mathcal{I}_1,\mathcal{P}_1}(M_1) + \ldots + p_k \cdot \mathbf{FN}^n_{\mathcal{I}_k,\mathcal{P}_k}(M_k))$, when $\mathcal{I} = p_1 \cdot \mathcal{I}_1 \uplus \ldots \uplus p_k \cdot \mathcal{I}_k$ and $\mathcal{P} = p_1 \cdot \mathcal{P}_1 \uplus \ldots \uplus p_k \cdot \mathcal{P}_k$ and $\sum_{(i,r) \in \mathcal{I}_j} r \leq 1$, for any $j \in \{1, \ldots, k\}$;*
2) *$\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M_1 \uplus \ldots \uplus M_k) = \min(1, p_1 \cdot \mathbf{FP}^n_{\mathcal{I}_1,\mathcal{P}_1}(M_1) + \ldots + p_k \cdot \mathbf{FP}^n_{\mathcal{I}_k,\mathcal{P}_k}(M_k))$, when $\mathcal{I} = p_1 \cdot \mathcal{I}_1 \uplus \ldots \uplus p_k \cdot \mathcal{I}_k$ and $\mathcal{P} = p_1 \cdot \mathcal{P}_1 \uplus \ldots \uplus p_k \cdot \mathcal{P}_k$ and $\sum_{(i,r) \in \mathcal{I}_j} r \leq 1$, for any $j \in \{1, \ldots, k\}$.*

When composing CPSs with pure-logical processes (up to eventual channel restrictions) we get, in general, a weaker result. This is because the introduction of a parallel pure-logical process potentially increases the number of (observable) alert signals, whereas channel restriction, in general, reduces their number as it makes signals non-observable.

Proposition 2 (Compositionality w.r.t. $\parallel$ and restriction)**.** *Let $M \in \mathcal{M}$ be a CPS with attacker's goal indicators $\mathcal{I}$, and detection policy $\mathcal{P}$. Let Q be a pure-logical process with detection policy $\mathcal{P}_Q$ over goal indicators $\mathcal{I}$ such that whenever $\overline{c}m \in \mathsf{out}(M)$ and $cm \in \mathsf{inp}(Q)$ then $\mathcal{P}(\overline{c}m, h) = 0$, for any $h \in \mathcal{I}^-$. Let c be a channel. Then,*

1) *$\mathbf{FN}^n_{\mathcal{I},\mathcal{P}'}(M \parallel Q) \leq \mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M)$, for $\mathcal{P}' = (\mathcal{P} \parallel \mathcal{P}_Q)$;*
2) *$\mathbf{FP}^n_{\mathcal{I},\mathcal{P}'}(M \parallel Q) \geq \mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$, for $\mathcal{P}' = (\mathcal{P} \parallel \mathcal{P}_Q)$;*
3) *$\mathbf{FN}^n_{\mathcal{I},\mathcal{P}'}(M \backslash c) \geq \mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M)$, for $\mathcal{P}' = (\mathcal{P} \backslash c)$;*
4) *$\mathbf{FP}^n_{\mathcal{I},\mathcal{P}'}(M \backslash c) \leq \mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$, for $\mathcal{P}' = (\mathcal{P} \backslash c)$.*

However, in a composite system, when all alert signals $\mathcal{A} \subseteq \mathsf{out}(M)$ coming from a CPS M in a single time slot are intercepted by a supervisor (logical-process) Sup which analyses and aggregates those signals, converting them in alert signals $\mathcal{B} \subseteq \mathsf{out}(Sups)$ in the same time slot, then we can strengthen the results of Proposition 2. As an example, suppose that $\mathcal{P}$ and $\mathcal{P}_{Sup}$ are the detection policy of M and Sup, respectively, with respect to the same set $\mathcal{I}$ of attacker's goal indicators. And suppose that at each time slot the detection policy of the supervisor amplifies the estimate of the attacker's progress made by the detection policy of M, *i.e.*, $\min(1, \sum_{\beta \in \mathcal{B}} \mathcal{P}_{Sup}(\beta, i)) \geq \min(1, \sum_{\alpha \in \mathcal{A}} \mathcal{P}(\alpha, i))$, for any $i \in \mathcal{I}$, abbreviated $\mathcal{P}_{Sup} \geq \mathcal{P}$.[5] Then, we can prove that the detection policy of the supervised system $(M \parallel Sup) \backslash \mathsf{chn}(\mathcal{A})$[6] is at least as aggressive as the detection policy $\mathcal{P}$ of the CPS M. As expected, the increased aggressiveness of the detection policy results in (a possibly) smaller number of false negatives and in (a possibly) greater number of false positives in the composite supervised system. More generally,

Theorem 2 (Compositionality on supervised systems)**.** *Let $M \in \mathcal{M}$ be a CPS with attacker's goal indicators $\mathcal{I}$, alert signals $\mathcal{A} \subseteq \mathsf{out}(M)$, and detection policy $\mathcal{P}$. Let Sup be a pure-logical process, with detection policy $\mathcal{P}_{Sup}$, collecting all alert signals in $\mathcal{A}$ and converting them in alert signals in $\mathcal{B} \subseteq \mathsf{out}(Sup)$, on fresh[7] communication channels. If $\mathcal{P}_{Sup} \sim \mathcal{P}$, where $\sim$ is an operator among $\{\leq, =, \geq\}$, then:*

[5]The reader is referred to the appendix for a formal definition of detection policy comparison, such as $\mathcal{P}_{Sup} \geq \mathcal{P}$, $\mathcal{P}_{Sup} = \mathcal{P}$ or $\mathcal{P}_{Sup} \leq \mathcal{P}$.

[6]Here, $\mathsf{chn}(\mathcal{A})$ denotes the set of channels occurring in the actions of $\mathcal{A}$.

[7]Here, "fresh" means that the channels used in $\mathcal{B}$ do not occur elsewhere.

1) $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(M) \sim \mathbf{FN}^n_{\mathcal{I},\mathcal{P}'}((M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A}))$, *for* $\mathcal{P}' = (\mathcal{P} \parallel \mathcal{P}_{Sup}) \setminus \mathsf{chn}(\mathcal{A})$
2) $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}'}((M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A})) \sim \mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$, *for* $\mathcal{P}' = (\mathcal{P} \parallel \mathcal{P}_{Sup}) \setminus \mathsf{chn}(\mathcal{A})$.

Based on the compositionality results stated in Theorem 1 and Theorem 2, we can lift the result of Proposition 1 to our composite system *Sys* of Example 3.

Proposition 3. *Let* $\mathcal{I}_{Sys}$ *and* $\mathcal{P}_{Sys}$ *be as defined in Example 5. Let* $\mathcal{I} = \mathcal{I}_{Sys}$ *and* $\mathcal{P} = \mathcal{P}_{Sys}$. *Then,* $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(Sys) = 0$ *and* $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(Sys) = 0$, *for any* $n \in \mathbb{N}$.

IV. Our impact metrics at work

In this section, we analyse four attacks targeting the *Eng* system given in Example 2, and aiming at causing either *overstress of the engine* [6] and/or *deception of the IDS* (increasing the false positives) in a possibly stealthy manner.

In order to test the (computational) feasibility of our metrics $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}$ and $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$, we simulate in MATLAB [15] the *Eng* system together with the four attacks mentioned above, computing their impacts during *attack windows* whose duration varies between 300 and 500 time instants. We rely on the compositionality results of Theorem 1 and Theorem 2 to estimate the impact of each attack, for the same attack window, on the composite system *Sys* of Example 3.

Our analyses are conducted on a notebook with the following set-up: 2.8 GHz Intel i7 7700 HQ, with 16 GB memory (plus 48 GB for swap), and Ubuntu 20.04 LTS OS. For each attack, the computation of the two metrics required at most 14 hours. The source files of both the *Eng* system and the attacks, together with analysis results and scripts to reproduce them, can be found at: https://bitbucket.org/formal_projects/impact_metrics/.

The *first attack* is an integrity attack to the actuator of the *Eng* system. It consists of forging fake actuator commands to turn off the cooling system. Formally,

$$Att_1 = \mathsf{rec}\, X.\mathsf{read}\ temp(y).[y \leq 99.5]\{\mathsf{write}\ cool\langle \mathsf{off}\rangle.\mathsf{tick}.X\}, \{\mathsf{tick}.X\}$$

Here, the attacker turns off the cooling system as soon as the temperature reaches 99.5 degrees, even before the completion of the 5-ticks cooling cycle. In this manner, the temperature of the engine rises quickly above the threshold 100, accumulating stress in the system, and requiring continuous activations of the cooling system. This is a *stealthy attack* (*i.e.*, undetected by our IDS) as it keeps the system close to the threshold 100 without ever exceeding it; only the attacker's goal indicator *stress* keeps track of the physical damage inflicted by the attack.

The impact of the attack in the first 500 time instants is represented in Figure 2. As the reader can see, the average number of false negatives of the *Eng* system under attack, *i.e.*, $\mathbf{FN}^n_{\mathcal{I}_{\mathrm{E}},\mathcal{P}_{\mathrm{E}}}(Eng \parallel Att_1)$, grows with the size of the attack window. Thus, after 500 time instants the metric $\mathbf{FN}^n_{\mathcal{I}_{\mathrm{E}},\mathcal{P}_{\mathrm{E}}}$ approaches the value 0.6, meaning that, in average, the attack achieves around 60% of its goals, according to $\mathcal{I}_{\mathrm{E}}$, *i.e.*, the goal indicator *stress*. Notice that this attack does not affect the false positives at all.

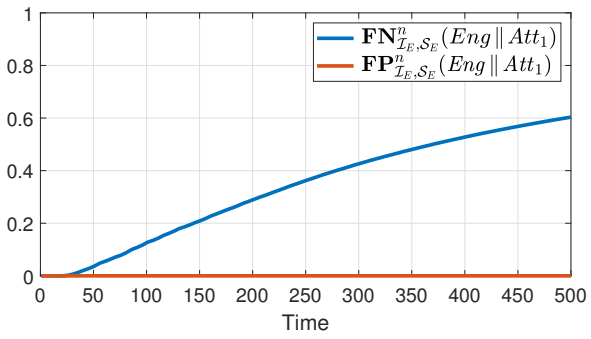

Figure 2. Impact of the first attack on the *Eng* system.

Now, we move to our composite system *Sys* of Example 3, and suppose that the attack Att_1 tampers with the left engine only. By relying on the compositionality results of our metrics we can infer the following result:

Proposition 4 (Impact of Att_1 on *Sys* when attacking Eng^{L})**.** *Let Sys be the system defined in Example 3 with attacker's goal indicators* $\mathcal{I}_{Sys}$ *and detection policy* $\mathcal{P}_{Sys}$, *as defined in Example 5. Let* $\mathcal{I} = \mathcal{I}_{Sys}$, $\mathcal{P} = \mathcal{P}_{Sys}$ *and* Att^{L}_1 *be* $Att_1\{^{temp^{\mathrm{L}}}/_{temp}\}\{^{cool^{\mathrm{L}}}/_{cool}\}$. *Then,*

- $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(Sys \parallel Att^{\mathrm{L}}_1) = 0.7 \cdot \mathbf{FN}^n_{\mathcal{I}_{\mathrm{L}},\mathcal{P}_{\mathrm{L}}}(Eng^{\mathrm{L}} \parallel Att^{\mathrm{L}}_1)$
- $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(Sys \parallel Att^{\mathrm{L}}_1) = 0.7 \cdot \mathbf{FP}^n_{\mathcal{I}_{\mathrm{L}},\mathcal{P}_{\mathrm{L}}}(Eng^{\mathrm{L}} \parallel Att^{\mathrm{L}}_1)$.

Proposition 4 says that, after 500 time instants, the metric $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ of the composite system approaches the value $0.7 \cdot 0.6 = 0.42$. As expected, the attack does not affect the false positives, *i.e.*, $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(Sys \parallel Att^{\mathrm{L}}_1) = 0$.

The *second attack* is an *integrity attack* to the sensor of the temperature of the *Eng* system. The attack adds a negative offset 0.6 to the temperature detected by the controller:

$$Att_2 = \mathsf{rec}\, X.\mathsf{read}\ temp(y).\mathsf{write}\ temp\langle y{-}0.6\rangle.\mathsf{tick}.X$$

In this attack, we have a different situation as the IDS raises an alert each time we have $100 < temp \leq 100.6$ and $cool = \mathsf{off}$. As a consequence, this attack *is not stealthy*.

The impact of our second attack in the first 300 time instants is represented in Figure 3. The effects of this attack are recorded in the goal indicator *stress*; the *false negatives* grow with the size of the attack window, reaching the value 0.18 after 300 time instants. As regards the *false positives*, as soon as the attack begins, the metric $\mathbf{FP}^n_{\mathcal{I}_{\mathrm{E}},\mathcal{P}_{\mathrm{E}}}$ reaches the value 0.1 and then stabilises. This indicates that, in average, the system under attack experiences 10 false positives every 100 time units, due to unjustified alert signals at channel *warn*.

The relatively low number of false negatives is basically due to the mitigation activity of the IDS (consisting in slowing down the engine) that delays the effects of the attack. In fact, without that mitigation the average number of false negatives would have been significantly higher, as depicted in Figure 4. Of course, a smarter attacker that would also drop the controller's command to slow down the *speed* (control variable) would cancel the mitigation efforts of the IDS.

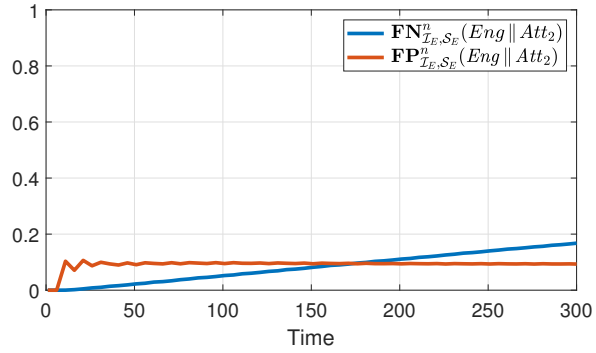

Figure 3. Impact of the second attack on the *Eng* system.

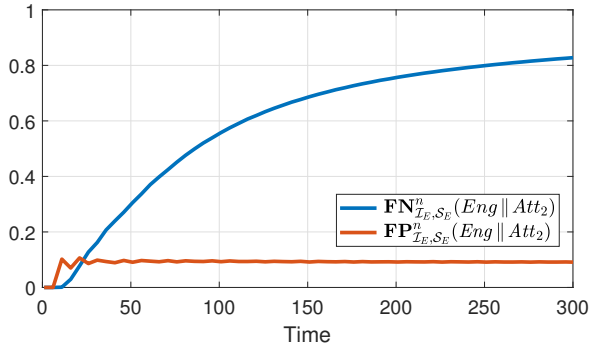

Figure 4. Impact of the second attack with detection but no mitigation.

Now, we move to our composite system *Sys* of Example 3, and suppose that the attack Att_2 tampers with the right engine only. By relying on the compositionality results of our metrics we can infer the following result:

Proposition 5 (Impact of Att_2 on *Sys* when attacking Eng^{R})**.** *Let Sys be the system defined in Example 3 with attacker's goal indicators $\mathcal{I}_{Sys}$ and detection policy $\mathcal{P}_{Sys}$, as defined in Example 5. Let $\mathcal{I} = \mathcal{I}_{Sys}$ and $\mathcal{P} = \mathcal{P}_{Sys}$. Let Att_2^{R} be the attack $Att_2\{^{temp\mathrm{R}}/_{temp}\}$. Then,*

- $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(Sys \,\|\, Att_2^{\mathrm{R}}) = 0.7 \cdot \mathbf{FN}^n_{\mathcal{I}_{\mathrm{R}},\mathcal{P}_{\mathrm{R}}}(Eng^{\mathrm{R}} \,\|\, Att_2^{\mathrm{R}})$
- $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(Sys \,\|\, Att_2^{\mathrm{R}}) = 0.7 \cdot \mathbf{FP}^n_{\mathcal{I}_{\mathrm{R}},\mathcal{P}_{\mathrm{R}}}(Eng^{\mathrm{R}} \,\|\, Att_2^{\mathrm{R}})$.

The false negatives of the composite system under attack reach the value $0.7 \cdot 0.18 = 0.126$. The false positives reach the value $0.7 \cdot 0.1 = 0.07$, and then stabilises.

Our *third attack* is a *DoS/integrity* attack on the actuator driving the cooling systems. The attacker tampers with the control variable *cool* to possibly drop two consecutive commands sent by the controller to turn on the cooling system. Once dropped, these commands are replaced with commands to turn off the cooling system. Formally, the attack is the following:

$$
\begin{array}{rcl}
Att_3 & = & \mathsf{rec}\, X.\mathsf{tick}^5.DropForge \\
DropForge & = & \mathsf{rec}\, Y.\mathsf{read}\ cool(y).[y = \mathsf{on}] \\
 & & \quad \{\mathsf{write}\ cool\langle \mathsf{off}\rangle.OnceMore\}, \\
 & & \quad \{\mathsf{write}\ cool\langle \mathsf{off}\rangle.\mathsf{tick}.Y\} \\
OnceMore & = & \mathsf{tick}.\mathsf{read}\ cool(y).\mathsf{write}\ cool\langle \mathsf{off}\rangle.\mathsf{tick}.X
\end{array}
$$

The attack repeatedly alternates between a *stand-by phase*, lasting 5 time units, and a *drop-and-forge phase*, lasting 2 time units, in which an integrity attack on the variable *cool* may occur once or twice, depending whether the controller transmits one or two (consecutive) commands to turn on the cooling system. Here, the goal indicator *stress* gets increased as the attacker may suddenly turn off the cooling system, reducing the 5-ticks cooling cycle. Thus, either the temperature remains just above the threshold 100 or the temperature drops below 100, but then it rapidly rises above that threshold.

The impact of this attack in the first 300 time instants is represented in Figure 5. The number of undetected and/or underestimated situations of overstress recorded by the goal indicator *stress* rapidly grows with the duration of the attack. After only 80 time instants the metric $\mathbf{FN}^n_{\mathcal{I}_{\mathrm{E}},\mathcal{P}_{\mathrm{E}}}$ reaches already the value 0.8; after 300 time instants it reaches the value 0.9. As regards the false positives, there is a small peak in the first time instants of malicious activity as the IDS raises alarms which are not related to increased overstress of the system. Thereafter, the average number of false positives quickly drops to 0. From the magnitude of false negatives it follows that the IDS misses most of the attacks targeting the stress of the system; as a consequence, basically, this *is a stealthy attack*.

Now, we move to our composite system *Sys* of Example 3, and suppose that the attack Att_3 tampers with the right engine only. By relying on the compositionality results of our metrics we can infer the following result:

Proposition 6 (Impact of Att_3 on *Sys* when attacking Eng^{R})**.** *Let Sys be the system defined in Example 3 with attacker's goal indicators $\mathcal{I}_{Sys}$ and detection policy $\mathcal{P}_{Sys}$, as defined in Example 5. Let $\mathcal{I} = \mathcal{I}_{Sys}$ and $\mathcal{P} = \mathcal{P}_{Sys}$. Let Att_3^{R} be $Att_3\{^{cool\mathrm{R}}/_{cool}\}$. Then,*

- $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(Sys \,\|\, Att_3^{\mathrm{R}}) = 0.7 \cdot \mathbf{FN}^n_{\mathcal{I}_{\mathrm{R}},\mathcal{P}_{\mathrm{R}}}(Eng^{\mathrm{R}} \,\|\, Att_3^{\mathrm{R}})$
- $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(Sys \,\|\, Att_3^{\mathrm{R}}) = 0.7 \cdot \mathbf{FP}^n_{\mathcal{I}_{\mathrm{R}},\mathcal{P}_{\mathrm{R}}}(Eng^{\mathrm{R}} \,\|\, Att_3^{\mathrm{R}})$.

Here, after 300 time instants, the metric $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}$ of the system *Sys* reaches the value $0.7 \cdot 0.9 = 0.63$. As expected, after a short transitory phase, false positives drops to 0.

Finally, we might think of a *fourth attack* $Att_4 = Att_1^{\mathrm{L}} \parallel Att_2^{\mathrm{R}}$ consisting of two *colluding attacks* focusing on the two engines, respectively: Att_1 tampering with the left engine, and Att_2 operating on the right engine. In this case, as expected, we would have both false negatives and false positives. However, the attacker's goal indicators of the two engines will let us know where these false alert signals are actually coming from.

Proposition 7 (Impact of Att_4 on *Sys* when attacking Eng^{L} and Eng^{R})**.** *Let Sys be the system defined in Example 3 with attacker's goal indicators $\mathcal{I}_{Sys}$ and detection policy $\mathcal{P}_{Sys}$, as defined in Example 5. Let $\mathcal{I} = \mathcal{I}_{Sys}$ and $\mathcal{P} = \mathcal{P}_{Sys}$. Let $n = 300$ and Att_4 be the attack $Att_1^{\mathrm{L}} \parallel Att_2^{\mathrm{R}}$. Then,*

- $\mathbf{FN}^n_{\mathcal{I},\mathcal{P}}(Sys \parallel Att_4) = \min(1, (0.7 \cdot \mathbf{FN}^n_{\mathcal{I}_{\mathrm{L}},\mathcal{P}_{\mathrm{L}}}(Eng^{\mathrm{L}} \parallel Att_1^{\mathrm{L}}) + 0.7 \cdot \mathbf{FN}^n_{\mathcal{I}_{\mathrm{R}},\mathcal{P}_{\mathrm{R}}}(Eng^{\mathrm{R}} \parallel Att_2^{\mathrm{R}})))$
- $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(Sys \parallel Att_4) = \min(1, (0.7 \cdot \mathbf{FP}^n_{\mathcal{I}_{\mathrm{L}},\mathcal{P}_{\mathrm{L}}}(Eng^{\mathrm{L}} \parallel Att_1^{\mathrm{L}}) + 0.7 \cdot \mathbf{FP}^n_{\mathcal{I}_{\mathrm{R}},\mathcal{P}_{\mathrm{R}}}(Eng^{\mathrm{R}} \parallel Att_2^{\mathrm{R}})))$.

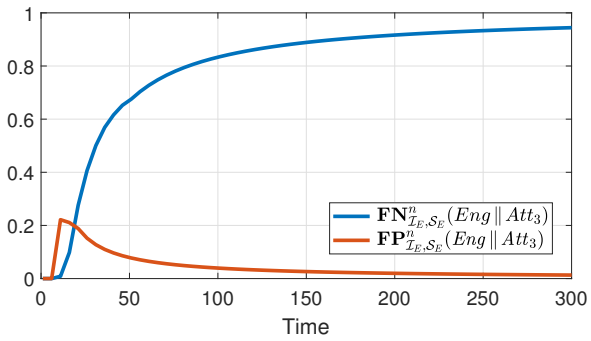

Figure 5. Impact of the third attack on the Eng system.

Here, after 300 time instants, the average false negatives approaches the value $0.7 \cdot 0.5 + 0.7 \cdot 0.18 = 0.476$, due to the malicious activity on the left engine Eng^{L}. Furthermore, the average false positives reaches the value $0.7 \cdot 0.1 = 0.07$ (and then stabilises), due to the malicious activity on Eng^{R}.

V. Conclusions and Related Work

We defined two probabilistic metrics, $\mathbf{FN}^{n}_{\mathcal{I},\mathcal{P}}$ and $\mathbf{FP}^{n}_{\mathcal{I},\mathcal{P}}$, to estimate the impact of (possibly stealthy) cyber-physical attacks in terms of the average false negatives and the average false positives generated by an arbitrary IDS. Our metrics build on the following concepts: (i) a timed probabilistic labelled transition semantics to formally describe the dynamics of (possibly compromised) CPSs; (ii) a set $\mathcal{I}$ of weighted attacker's goal indicators whose runtime values denote how close is the attacker to reach these goals (equipment or production damage, compliance violations); (iii) a detection policy $\mathcal{P}$ that, given an alert signal (raised by the IDS) and an attacker's goal indicator in $\mathcal{I}$, returns a statically-determined estimate on the progresses of the detected attack in achieving the goal with that indicator.

Our impact metrics can be computed in a compositional manner on those CPSs whose sub-systems may physically interact only if the associated logical components agree on when and how that interaction should happen. Examples of such kind of systems have beed discussed in the Introduction.

We showed the computational feasibility of our metrics with the help of a non-trivial use case and four significant cyber-physical attacks, both implemented in MATLAB [15].

As *future work*, we plan to improve the computational efficiency of our metrics by selecting execution traces of the CPS under analysis via Monte Carlo simulation methods [18]. Of course, simulation-based solutions do not ensure results with a 100% confidence. However, the number of executions that our simulator should perform to reach a desired confidence can be determined using well-known statistical techniques [19]. This will allow us to scale our impact metrics on larger CPSs that are not necessarily compositional, according to our definitions.

Related Work: Different methodologies for assessing the impact of attacks targeting CPSs have been proposed. Here, we discuss and compare the papers closest to our work.

Urbina et al. [13] addressed the problem of detecting greedy attacks to sensors or actuators, but not both, as we do. Based on a strong adversary model that is always able to bypass *stateful* IDSs (relying on cumulative sums - CUSUM), they introduced an evaluation metric for attack-detection algorithms that quantifies the negative impact of stealthy attacks and the inherent trade-off with false positives. In practice, in order to define the impact of stealthy attacks the authors select one (or more) variable(s) of interest in the process they want to control. Then, the impact of the undetected attack will measure how much can the attacker drive that value towards its intended goal per unit of time. More precisely, their metric measures the tradeoff between the maximum deviation per time unit imposed by stealthy attacks and the expected time between false positives. Their metric relies exclusively on observable physical variables without explicitly addressing equipment damage, production damage, or compliance violations [6]. We do that by introducing attacker's goal indicators $\mathcal{I}$ to capture manipulations on (unobservable) physical variables whose evolution is beyond the control of IDSs.

Umsonst et al. [20] provided the mathematical theory to expand the analysis of [13] based on CUSUM IDSs. In particular, they considered attacks that can tamper with both sensors and actuators at the same time, and can optimise their malicious patterns over a window of time. However, the works of [13] and [20] neglect the influence of the uncertainty of the models and do not propose substitutes for their infinity-norm-based metrics to be used in stochastic systems.

Huang et al. [21] proposed a risk assessment method that uses a Bayesian network to model the attack propagation process and infers the probabilities of sensors and actuators to be compromised. These probabilities are fed into a stochastic hybrid system model to predict the evolution of the physical process being controlled. Then, the security risk is quantified by evaluating the system availability with the model.

More recently, Milošević et al. [22] overcome the limitations of the infinity-norm-based metrics of [13], [20] by proposing two metrics to measure the impact of stealthy cyber-attacks in stochastic linear networked control systems. The first metric returns the probability that some of the critical states leave a safety region; the second metric measures the deviation of such critical states with respect to the expected values. Their metrics relies on two tuning parameters: $N \in \mathbb{N}$, denoting the time window over which the impact is estimated (similar to our parameter n), and $\epsilon \geq 0$, *i.e.*, the stealthiness level based on the Kullback-Leibler divergence. Somehow, these two metrics together provide an information similar to what we obtain with our metric $\mathbf{FN}^{n}_{\mathcal{I},\mathcal{P}}$, except for the fact that we rely on the detection policy $\mathcal{P}$ to distill stealthiness instead of using Kullback-Leibler divergence. Unlike us, the authors do not take into consideration false positives introduced by the attacker to mislead system engineers.

Other works on impact evaluation of attacks targeting CPSs are the following. Genge et al. [23] introduced a methodology, inspired by research in system dynamics and sensitivity analysis, to compute the covariances of the observed variables before and after the execution of specific interventions involving the control variables. Bilis et al. [24] proposed five metrics derived

from network theory to assess the impacts of cyber attacks on power systems. Sgouras et al. [25] evaluated the impact of attacks on a simulated smart metering infrastructure. Sridhar and Govindarasu [26] evaluated the impact of attacks on wide-area frequency control applications in power systems.

As regards *formal methodologies*, Lanotte et al. [11], [10] defined hybrid process calculi to model both CPSs and cyber-physical attacks; our calculus is a probabilistic simplification of theirs. The paper [10] proposes a formalisation of physical impact of an attack in terms of the *potential perturbation* that might be introduced by the attacker during a compromised execution of the system. However, the model in that paper does not support probabilities to represent system uncertainty. Thus, the notion of impact in [10] does not take into account the attacker's chances of achieving an attack with that physical impact/damage: it just looks at the existence of a malicious execution trace reaching that impact/damage; it says nothing about the likelihood of such an execution actually taking place. More recently, Lanotte et al. [27] have used a discrete-time generalisation of Desharnais et al.'s *weak bisimulation metric* [28] to estimate the impact of attacks targeting sensor devices of IoT systems; however, their impact metric takes into consideration only the logical effects on (communication channels of) the IoT system under attack, but not the physical ones. The objective of this paper is to bring the ideas from these two papers together by defining impact metrics that take into account both the physical impact and the likelihood to successfully completing an attack with that physical impact.

Finally, we wish to underline that *formal methodologies* are increasingly used to lay theoretical foundations to reason about cyber-physical systems and their vulnerabilities (see, *e.g.*, [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39]).

Acknowledgment

We thank the anonymous reviewers for their comments. Massimo Merro has been partially supported by the project "Dipartimenti di Eccellenza 2018–2022", funded by the Italian Ministry of Universities and Research (MUR).

References

[1] R. Rajkumar, I. Lee, L. Sha, and J. A. Stankovic, "Cyber-physical systems: the next computing revolution," in *DAC*. ACM, 2010, pp. 731–736.
[2] Y. Huang, A. A. Cárdenas, S. Amin, Z. Lin, H. Tsai, and S. Sastry, "Understanding the physical and economic consequences of attacks on control systems," *IJCIP*, vol. 2, no. 3, pp. 73–83, 2009.
[3] N. Falliere, L. Murchu, and E. Chien, "W32.Stuxnet Dossier," 2011.
[4] J. Slowik, "Anatomy of an attack: Detecting and defeating CRASHOVERRIDE," *VB2018, October*, 2018.
[5] A. Di Pinto, Y. Dragoni, and A. Carcano, "TRITON: The First ICS Cyber Attack on Safety Instrument Systems," in *Black Hat USA*, 2018.
[6] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, and A. Winnicki, "Cyber-Physical Systems Security: Experimental Analysis of a Vinyl Acetate Monomer Plant," in *CPSS*. ACM, 2015, pp. 1–12.
[7] R. Milner, J. Parrow, and D. Walker, "A Calculus of Mobile Processes," *Inf. Comput.*, vol. 100, no. 1, pp. 1–40, 1992.
[8] L. Cardelli and A. Gordon, "Mobile Ambients," *Theoretical Computer Science*, vol. 240, no. 1, pp. 177–213, 2000.
[9] M. Abadi, B. Blanchet, and C. Fournet, "The applied pi calculus: mobile values, new names and secure communication," *JACM*, vol. 65:1-41, 2018.
[10] R. Lanotte, M. Merro, A. Munteanu, and L. Viganò, "A formal approach to physics-based attacks in cyber-physical systems," *ACM Trans. Priv. Secur.*, vol. 23, no. 1, pp. 3:1–3:41, 2020.
[11] R. Lanotte, M. Merro, and S. Tini, "A probabilistic calculus of cyber-physical systems," *Information and Computation*, no. 104618, 2020.
[12] R. Segala, "Modeling and Verification of Randomized Distributed Real-Time Systems," Ph.D. dissertation, MIT, 1995.
[13] D. I. Urbina, J. A. Giraldo, A. A. Cárdenas, N. O. Tippenhauer, J. Valente, M. A. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," in *CCS*. ACM, 2016, pp. 1092–1105.
[14] A. A. Cárdenas, J. S. Baras, and K. Seamon, "A Framework for the Evaluation of Intrusion Detection Systems," in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2006, pp. 63–77.
[15] MATLAB, *9.7.0.1190202 (R2019b)*. US: The MathWorks Inc., 2018.
[16] M. Hennessy and T. Regan, "A Process Algebra for Timed Systems." *Information and Computation*, vol. 117, no. 2, pp. 221–239, 1995.
[17] R. M. Keller, "Formal verification of parallel programs." *Communications of the ACM*, vol. 19, pp. 371–384, 1976.
[18] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
[19] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *The Annals of Mathematical Statistics*, vol. 23, no. 4, pp. 493–507, 12 1952.
[20] D. Umsonst, H. Sandberg, and A. A. Cárdenas, "Security analysis of control system anomaly detectors," in *ACC*. IEEE, 2017, pp. 5500–5506.
[21] K. Huang, C. Zhou, Y. Tian, S. Yang, and Y. Qin, "Assessing the Physical Impact of Cyberattacks on Industrial Cyber-Physical Systems," *IEEE Trans. Industrial Electronics*, vol. 65, no. 10, pp. 8153–8162, 2018.
[22] J. Milošević, H. Sandberg, and K. H. Johansson, "Estimating the Impact of Cyber-Attack Strategies for Stochastic Networked Control Systems," *IEEE Trans. Control. Netw. Syst.*, vol. 7, no. 2, pp. 747–757, 2020.
[23] B. Genge, I. Kiss, and P. Haller, "A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures," *Int. J. Critical Infrastructure Protection*, vol. 10, pp. 3–17, 2015.
[24] E. I. Bilis, W. Kröger, and N. Cen, "Performance of Electric Power Systems Under Physical Malicious Attacks," *IEEE Systems Journal*, vol. 7, no. 4, pp. 854–865, 2013.
[25] K. I. Sgouras, A. I. Birda, and D. L. Labridis, "Cyber attack impact on critical Smart Grid infrastructures," in *PES ISGT*. IEEE, 2014, pp. 1–5.
[26] 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, 2014.
[27] R. Lanotte, M. Merro, and S. Tini, "Towards a formal notion of impact metric for cyber-physical attacks," in *iFM*, ser. LNCS, vol. 11023. Springer, 2018, pp. 296–315.
[28] J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden, "The Metric Analogue of Weak Bisimulation for Probabilistic Processes," in *LICS*. IEEE Computer Society, 2002, pp. 413–422.
[29] V. Nigam, C. Talcott, and A. Urquiza, "Towards the Automated Verification of Cyber-Physical Security Protocols: Bounding the Number of Timed Intruders," in *ESORICS*, ser. LNCS, vol. 9879:450-470, 2016.
[30] M. Rocchetto and N. Tippenhauer, "On attacker models and profiles for cyber-physical systems," in *ESORICS*, ser. LNCS, vol. 9879:427-449, 2016.
[31] B. Bohrer and A. Platzer, "A Hybrid, Dynamic Logic for Hybrid-Dynamic Information Flow," in *LICS*. ACM/IEEE, 2018, pp. 115–124.
[32] V. Nigam and C. L. Talcott, "Formal Security Verification of Industry 4.0 Applications," in *ETFA*. IEEE, 2019, pp. 1043–1050.
[33] C. Bodei, S. Chessa, and L. Galletta, "Measuring security in IoT communications," *Theor. Comput. Sci.*, vol. 764, pp. 100–124, 2019.
[34] M. Barrère, C. Hankin, N. Nicolaou, D. G. Eliades, and T. Parisini, "Measuring cyber-physical security in industrial control systems via minimum-effort attack strategies," *J. Inf. Secur. Appl.*, vol. 52, 2020.
[35] L. Lin, Y. Zhu, and R. Su, "Synthesis of covert actuator attackers for free," *Discret. Event Dyn. Syst.*, vol. 30, no. 4, pp. 561–577, 2020.
[36] R. Lanotte, M. Merro, and A. Munteanu, "Runtime Enforcement for Control System Security," in *CSF*. IEEE, 2020, pp. 246–261.
[37] C. Bernardeschi, A. Domenici, and M. Palmieri, "Formalization and co-simulation of attacks on cyber-physical systems," *J. Comput. Virol. Hacking Tech.*, vol. 16, no. 1, pp. 63–77, 2020.
[38] D. Bresolin, P. Collins, L. Geretti, R. Segala, T. Villa, and S. Z. Gonzalez, "A computable and compositional semantics for hybrid automata," in *HSCC*. ACM, 2020, pp. 18:1–18:11.
[39] I. Jahandideh, F. Ghassemi, and M. Sirjani, "An actor-based framework for asynchronous event-based cyber-physical systems," *Software and Systems Modeling*, pp. 1–25, in press.

APPENDIX

A. Proofs of Section III

Proof of Proposition 1. Given a trace $M_1 \xrightarrow{\alpha_1 \ldots \alpha_n} M_{n+1}$, with $M_1 = \mathit{Eng}$, for $j \in 1..n{+}1$, we let t_j, c_j, sp_j and s_j denote the value of the variables *temp*, *cool*, *speed* and *stress*, respectively, in the physical state of M_j. We recall that $\mathcal{I}_{\mathrm{E}}^- = \{\mathit{stress}\}$, $\mathcal{P}_{\mathrm{E}}(\overline{\mathit{warn}}\langle \mathrm{ID}, \mathsf{hot}\rangle, \mathit{stress}) = 1$ and $\mathcal{P}_{\mathrm{E}}(\alpha, \mathit{stress}) = 0$, for all $\alpha \neq \overline{\mathit{warn}}\langle \mathrm{ID}, \mathsf{hot}\rangle$. Thus, in order to derive the result it is sufficient to show that any trace $M_1 \xrightarrow{\alpha_1 \ldots \alpha_{n-1}} M_n$ satisfies: 1) $\alpha_j \neq \overline{\mathit{warn}}\langle \mathrm{ID}, \mathsf{hot}\rangle$, for $j \in 1..n{-}1$; and 2) $s_j = 0$, for $j \in 1..n$. Indeed, $\mathbf{FP}^n_{\mathcal{I}_{\mathrm{E}}, \mathcal{P}_{\mathrm{E}}}(\mathit{Eng}) = 0$ follows by 1), and $\mathbf{FN}^n_{\mathcal{I}_{\mathrm{E}}, \mathcal{P}_{\mathrm{E}}}(\mathit{Eng}) = 0$ follows by 2). We prove these two properties separately.

Case 1). Since $\overline{\mathit{warn}}\langle \mathrm{ID}, \mathsf{hot}\rangle$ can be emitted uniquely by the process *IDS* when $(z_{temp} > 100 \wedge z_{cool} = \mathsf{off})$ is true, it is sufficient to prove that the condition $(t_j > 100 \wedge c_j = \mathsf{off})$ is never satisfied, for any $j \in 1..n$. We know that $t_1 = 95$ and $c_1 = \mathsf{off}$. Moreover, the temperature rises by at most 1.2 degrees per time unit. Therefore, at some instant k the temperature exceeds the threshold, *i.e.*, $100 < t_k < 101.2$. The controller *Ctrl* immediately turns on the cooling system, which remains active for 5 time instants, namely $c_j = \mathsf{on}$ for $j \in k..k{+}4$. This ensures that $(t_j > 100 \wedge c_j = \mathsf{off})$ remains false for $j \in k..k{+}4$. In this time interval, the cooling system lowers the temperature by at least $0.8 \cdot 5 = 4$ degrees. As a consequence, we have $t_{k+5} < 97.2$. This implies that $(t_j > 100 \wedge c_j = \mathsf{off})$ remains false at instant $k + 5$. The reasoning can now be reiterated to prove the required result.

Case 2). We will show that for any $j \geq 0$ we have: (a) if $t_j > 100$ then $t_{j+1} \leq 100$ or $t_{j+2} \leq 100$; (b) if $t_j > 100$ and $t_{j+1} \leq 100$, then $t_{j+1+k} \leq 100$ for $k \in 1 \ldots 4$. From (a) and (b) it follows that for any arbitrary time interval $i..i{+}5$ at most two values in $\{t_i, \ldots, t_{i+5}\}$ may be above the threshold 100. This ensures that *stress* is never incremented and maintains the initial value 0. In order to show (a), we note that the temperature cannot rise above $100 + 1.2 = 101.2$ degrees. Hence $t_j \in (100, 101.2]$. The cooling system is turned on implying $t_{j+1} \in (100 - 1.2, 101.2 - 0.8] = (98.8, 100.4]$ and $t_{j+2} \in (98.8{-}1.2, 100.4{-}0.8] = (97.6, 99.6]$. This gives (a). In order to show (b), from $t_j > 100$ and $t_{j+1} \leq 100$ we infer $t_{j+1} \in (100{-}1.2, 100] = (98.8, 100]$. The cooling system operates for further 4 time units and $t_{j+1+k} \in (98.8{-}k \cdot 1.2, 100 - k \cdot 0.8]$, namely $t_{j+1+k} \leq 100$, for $k \in 1..4$. □

In order to prove Theorem 1, we need a preliminary result to formalise the relationship between the attackers' progresses associated with the traces of a CPS $M_1 \uplus M_2$, and those associated with the traces of the two sub-systems M_1 and M_2.

Lemma 1. *Let M_1 and M_2 be two CPSs with disjoint sets of weighted attacker's goal indicators $\mathcal{I}_1 = \{(i^1_1, r^1_1), \ldots, (i^1_{k_1}, r^1_{k_1})\}$ and $\mathcal{I}_2 = \{(i^2_1, r^2_1), \ldots, (i^2_{k_2}, r^2_{k_2})\}$, and detection policies $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Assume that for any $i, j \in \{1, 2\}$, $i \neq j$, whenever $\overline{c}m \in \mathsf{out}(M_i)$ and $cm \in \mathsf{inp}(M_j)$ then $\mathcal{P}_i(\overline{c}m, h) = 0$, for any $h \in \mathcal{I}_i^-$. Then, for two tuples of reals $\overline{q^1} = q^1_1, \ldots, q^1_{k_1}$ and $\overline{q^2} = q^2_1, \ldots, q^2_{k_2}$, the following two properties are equivalent:*

1) $M_1 \uplus M_2 \xrightarrow{\overline{q^1}, \overline{q^2}}_{\mathcal{P}} \gamma$, with $\mathcal{P} = p_1 \cdot \mathcal{P}_1 \uplus p_2 \cdot \mathcal{P}_2$;

2) $M_1 \xrightarrow{\overline{q^1}}_{\mathcal{P}_1} \gamma_1$, $M_2 \xrightarrow{\overline{q^2}}_{\mathcal{P}_2} \gamma_2$ and $\gamma = \gamma_1 \uplus \gamma_2$,

where $\gamma_1 \uplus \gamma_2$ is the distribution over CPSs such that $(\gamma_1 \uplus \gamma_2)(N_1 \uplus N_2) = \gamma_1(N_1) \cdot \gamma_2(N_2)$, for all CPSs N_1 and N_2.

Proof: We show that property 1) implies 2), the other implication is similar. Let $M_1 = \langle \varepsilon_1; \xi_{\mathsf{V}_1}\rangle \bowtie P_1$ and $M_2 = \langle \varepsilon_2; \xi_{\mathsf{V}_2}\rangle \bowtie P_2$, for evolution laws ε_1 and ε_2, state functions ξ_{V_1} and ξ_{V_2} and processes P_1 and P_2. Let $N^0 = M_1 \uplus M_2$. By Definition 9, the transition in 1) entails:

- $N^0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} N^n$, where $\alpha_j \neq \mathsf{tick}$, for $j \in 1..n$
- $N^n \xrightarrow{\mathsf{tick}} \gamma$.

All systems N^j, with $j \in 0..n$, have the form $N^j = \langle \varepsilon_1 \uplus \varepsilon_2; \xi_{\mathsf{V}_1} \uplus \xi_{\mathsf{V}_2}\rangle \bowtie P^j_1 \parallel P^j_2$, for some processes P^j_1 and P^j_2. The proof of this statement follows by induction on j and by inspection on the transition rules of Tables I and II. Then, the transitions $N^j \xrightarrow{\alpha_{j+1}} \overline{N^{j+1}}$, for $j \in 0..n{-}1$, can be derived only because $P^j_1 \parallel P^j_2 \xrightarrow{\beta_{j+1}} P^{j+1}_1 \parallel P^{j+1}_2$, for some β_{j+1}.

The transitions $N^n \xrightarrow{\mathsf{tick}} \gamma$ can be derived only because $P^n_1 \parallel P^n_2 \xrightarrow{\mathsf{tick}} P^{n+1}_1 \parallel P^{n+1}_2$ by the application of either rule (ReadIDS) or rule (Time). Thus, the distribution γ of the transition $N^n \xrightarrow{\mathsf{tick}} \gamma$ has the form $\gamma = \gamma_1 \uplus \gamma_2$, where $\gamma_1 = \varepsilon_1(\xi_{\mathsf{V}_1}) \bowtie P^{n+1}_1$ and $\gamma_2 = \varepsilon_2(\xi_{\mathsf{V}_2}) \bowtie P^{n+1}_2$, for some processes P^{n+1}_1 and P^{n+1}_2.

Thus, let $N^j_1 = \langle \varepsilon_1; \xi_{\mathsf{V}_1}\rangle \bowtie P^j_1$ and $N^j_2 = \langle \varepsilon_2; \xi_{\mathsf{V}_2}\rangle \bowtie P^j_2$, for $j \in 1..n$, the transitions of CPSs N^j, for $j \in 0..n$, follow from transitions of the associated processes. There are 4 cases.

Case (i). $P^j_1 \parallel P^j_2 \xrightarrow{\beta_{j+1}} P^{j+1}_1 \parallel P^{j+1}_2$ is derived by an application of rule (Par) in Table I because $P^j_1 \xrightarrow{\beta_{j+1}} P^{j+1}_1$ and $P^{j+1}_2 = P^j_2$. Now, whatever is the rule in Table II used to derive $N^j \xrightarrow{\alpha_{j+1}} N^{j+1}$ from $P^j_1 \parallel P^j_2 \xrightarrow{\beta_{j+1}} P^{j+1}_1 \parallel P^{j+1}_2$, this rule allows us to derive $N^j_1 \xrightarrow{\alpha_{j+1}} N^{j+1}_1$ as well, from $P^j_1 \xrightarrow{\beta_{j+1}} P^{j+1}_1$. As $\mathcal{P} = p_1 \cdot \mathcal{P}_1 \uplus p_2 \cdot \mathcal{P}_2$, we have $\mathcal{P}_1(\alpha_{j+1}, i^1_l) = \mathcal{P}(\alpha_{j+1}, i^1_l)$, for $\alpha_{j+1} \in \mathsf{out}(M_1)$ and $l = 1..k_1$.

Case (ii). $P^j_1 \parallel P^j_2 \xrightarrow{\beta_{j+1}} P^{j+1}_1 \parallel P^{j+1}_2$ is derived by applying rule (Par) because $P^j_2 \xrightarrow{\beta_{j+1}} P^{j+1}_2$. Similarly to case (i) we have $\mathcal{P}_2(\alpha_{j+1}, i^2_l) = \mathcal{P}(\alpha_{j+1}, i^2_l)$, for $\alpha_{j+1} \in \mathsf{out}(M_2)$ and $l = 1..k_2$.

Case (iii). $P^j_1 \parallel P^j_2 \xrightarrow{\beta_{j+1}} P^{j+1}_1 \parallel P^{j+1}_2$ is derived by rule (Comm) in Table I because $P^j_1 \xrightarrow{\beta^1_{j+1}} P^{j+1}_1$ and $P^j_2 \xrightarrow{\beta^2_{j+1}} P^{j+1}_2$. Without loss of generality, we assume $\beta^1_{j+1} = \overline{c}m$ and $\beta^2_{j+1} = cm$. By rules (Receive) and (Send), we infer $N^j_1 \xrightarrow{\alpha^1_{j+1}} N^{j+1}_1$ and $N^j_2 \xrightarrow{\alpha^2_{j+1}} N^{j+1}_2$, with $\alpha^1_{j+1} = \beta^1_{j+1}$ and $\alpha^2_{j+1} = \beta^2_{j+1}$. Since $\alpha^1_{j+1} = \overline{c}m \in \mathsf{out}(M_1)$ and $\alpha^2_{j+1} = cm \in \mathsf{inp}(M_2)$, by applying the hypothesis of the lemma we have $\mathcal{P}_1(\alpha^1_{j+1}, i^1_l) = 0$. Thus, $\alpha^2_{j+1} \notin \mathsf{out}(M_2)$ and $\alpha_{j+1} = \tau \notin \mathsf{out}(M)$.

Case (iv). $P^n_1 \parallel P^n_2 \xrightarrow{\mathsf{tick}} P^{n+1}_1 \parallel P^{n+1}_2$ is derived by an application of rule (Timepar) in Table I because $P^n_1 \xrightarrow{\mathsf{tick}} P^{n+1}_1$ and $P^n_2 \xrightarrow{\mathsf{tick}} P^{n+1}_2$. In this case, we have: $N^n_1 \xrightarrow{\mathsf{tick}}$

γ_1 with $\gamma_1 = \varepsilon_1(\xi_{\mathsf{V}_1}) \bowtie P_1^{n+1}$, and $N_2^n \xrightarrow{\mathsf{tick}} \gamma_2$ with $\gamma_2 = \varepsilon_2(\xi_{\mathsf{V}_2}) \bowtie P_2^{n+1}$.

Now, from the form of the transitions originating from CPSs N_i^j and N_i^n, for $j \in 0..n{-}1$ and $i \in 1..2$, we derive the following five facts:

(a) $M_1 = N_1^0 \xrightarrow{\delta_{1,1}} \ldots \xrightarrow{\delta_{1,n_1}} N_1^{n_1} = N_1^n \xrightarrow{\mathsf{tick}} \gamma_1$;
(b) for all $j \in 1..n_1$ and $l \in 1..r_1$, either $\mathcal{P}_1(\delta_{1,j}, i_l^1) = \mathcal{P}(\alpha_j, i_l^1)$ or both $\delta_{1,j} \notin \mathsf{out}(M_1)$ and $\alpha_j \notin \mathsf{out}(M)$;
(c) $M_2 = N_2^0 \xrightarrow{\delta_{2,1}} \ldots \xrightarrow{\delta_{2,n_2}} N_2^{n_2} = N_2^n \xrightarrow{\mathsf{tick}} \gamma_2$;
(d) for all $j \in 1..n_2$ and $l \in 1..r_2$, either $\mathcal{P}_2(\delta_{2,j}, i_l^2) = \mathcal{P}(\alpha_j, i_l^2)$ or both $\delta_{2,j} \notin \mathsf{out}(M_2)$ and $\alpha_j \notin \mathsf{out}(M)$;
(e) $\delta_{1,j_1} \neq \mathsf{tick}$ and $\delta_{2,j_2} \neq \mathsf{tick}$, for $j_1 \in 1..n_1$, $j_2 \in 1..n_2$.

From (a) and (e), by applying Definition 9 we derive the transition $M_1 \xrightarrow{r_1^1,\ldots,r_{k_1}^1}_{\mathcal{P}_1} \gamma_1$, with $r_l^1 = \min(1, \sum_{j=1}^{n_1} r_l^{1,j})$; here, $r_l^{1,j} = \mathcal{P}_1(\delta_{1,j}, i_l^1)$ if $\delta_{1,j} \in \mathsf{out}(M_1)$, and $r_l^{1,j} = 0$ otherwise. We recall that by Definition 9, the transition in item 1) of the hypothesis of the lemma entails $q_l^1 = \min(1, \sum_{j=1}^{n} q_l^{1,j})$, where $q_l^{1,j} = \mathcal{P}(\alpha_j, i_l^1)$ if $\alpha_j \in \mathsf{out}(M_1)$, and $q_l^{1,j} = 0$ otherwise. Hence, by fact (b) it follows that $r_l^1 = q_l^1$. This implies $M_1 \xrightarrow{\overline{q^1}}_{\mathcal{P}_1} \gamma_1$, for $\overline{q^1} = q_1^1, \ldots, q_{k_1}^1$. By applying a similar reasoning, from facts (c), (d) and (e) we derive $M_2 \xrightarrow{\overline{q^2}}_{\mathcal{P}_2} \gamma_2$. The result follows because $\gamma = \gamma_1 \uplus \gamma_2$. ■

Now, everything is in place to prove Theorem 1.

Proof of Theorem 1. We give the proof for the metric **FP**. The proof for the metric **FN** is similar. Moreover, we consider the case $k = 2$, the extension to $k > 2$ is straightforward. We reason by induction on n. The base case, $n = 0$, is immediate. We move to the inductive case, $n > 0$. Let $M = M_1 \uplus M_2$. Assume $\mathcal{I}_1 = \{(i_1^1, r_1^1), \ldots, (i_{k_1}^1, r_{k_1}^1)\}$ and $\mathcal{I}_2 = \{(i_1^2, r_1^2), \ldots, (i_{k_2}^2, r_{k_2}^2)\}$. Let $\overline{q^1}$ denote a tuple of reals $q_1^1, \ldots, q_{k_1}^1$ and $\overline{q^2}$ a tuple of reals $q_1^2, \ldots, q_{k_2}^2$. Then, we have:

$$
\begin{aligned}
&\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)\\
&= \min\left(1, \max_{M \xrightarrow{\overline{q^1},\overline{q^2}}_{\mathcal{P}}\gamma} \frac{\sum_{j=1}^{k_1} p_1 \cdot r_j^1 \cdot \max(0, q_j^1 - i_j^1(M)) + \sum_{j=1}^{k_2} p_2 \cdot r_j^2 \cdot \max(0, q_j^2 - i_j^2(M)) + (n-1)\sum_{M' \in \mathsf{supp}(\gamma)} \gamma(M') \cdot \mathbf{FP}^{n-1}_{\mathcal{I},\mathcal{P}}(M')}{n}\right)\\
&= \min\left(1, \max_{\substack{M_1 \xrightarrow{\overline{q_1}}_{\mathcal{P}_1}\gamma_1\\ M_2 \xrightarrow{\overline{q_2}}_{\mathcal{P}_2}\gamma_2}} \frac{\sum_{j=1}^{k_1} p_1 \cdot r_j^1 \cdot \max(0, q_j^1 - i_j^1(M)) + \sum_{j=1}^{k_2} p_2 \cdot r_j^2 \cdot \max(0, q_j^2 - i_j^2(M)) + (n-1)\sum_{\substack{M_1' \in \mathsf{supp}(\gamma_1)\\ M_2' \in \mathsf{supp}(\gamma_2)}} \gamma_1(M_1') \cdot \gamma_2(M_2') \cdot \mathbf{FP}^{n-1}_{\mathcal{I},\mathcal{P}}(M_1' \uplus M_2')}{n}\right)\\
&= \min\left(1, \max_{\substack{M_1 \xrightarrow{\overline{q_1}}_{\mathcal{P}_1}\gamma_1\\ M_2 \xrightarrow{\overline{q_2}}_{\mathcal{P}_2}\gamma_2}} \frac{\sum_{j=1}^{k_1} p_1 \cdot r_j^1 \cdot \max(0, q_j^1 - i_j^1(M)) + \sum_{j=1}^{k_2} p_2 \cdot r_j^2 \cdot \max(0, q_j^2 - i_j^2(M)) + (n-1)\cdot\sum_{\substack{M_1' \in \mathsf{supp}(\gamma_1)\\ M_2' \in \mathsf{supp}(\gamma_2)}} \gamma_1(M_1') \cdot \gamma_2(M_2') \cdot \min(1, p_1 \cdot \mathbf{FP}^{n-1}_{\mathcal{I}_1,\mathcal{P}_1}(M_1') + p_2 \cdot \mathbf{FP}^{n-1}_{\mathcal{I}_2,\mathcal{P}_2}(M_2'))}{n}\right)\\
&= \min\left(1, \max_{\substack{M_1 \xrightarrow{\overline{q_1}}_{\mathcal{P}_1}\gamma_1\\ M_2 \xrightarrow{\overline{q_2}}_{\mathcal{P}_2}\gamma_2}} \frac{p_1 \cdot \sum_{j=1}^{k_1} r_j^1 \cdot \max(0, q_j^1 - i_j^1(M_1)) + (n-1)\sum_{M_1' \in \mathsf{supp}(\gamma_1)} \gamma_1(M_1') p_1 \mathbf{FP}^{n-1}_{\mathcal{I}_1,\mathcal{P}_1}(M_1') + p_2 \cdot \sum_{j=1}^{k_2} r_j^2 \cdot \max(0, q_j^2 - i_j^2(M_2)) + (n-1)\sum_{M_2' \in \mathsf{supp}(\gamma_2)} \gamma_2(M_2') p_2 \mathbf{FP}^{n-1}_{\mathcal{I}_2,\mathcal{P}_2}(M_2')}{n}\right)\\
&= \min\left(1, \max_{\substack{M_1 \xrightarrow{q_1}_{\mathcal{P}_1}\gamma_1\\ M_2 \xrightarrow{q_2}_{\mathcal{P}_2}\gamma_2}} \frac{(n-1)\sum_{M_2' \in \mathsf{supp}(\gamma_2)} \gamma_2(M_2') p_2 \mathbf{FP}^{n-1}_{\mathcal{I}_2,\mathcal{P}_2}(M_2')}{n}\right)\\
&= \min\left(1,\ p_1 \cdot \mathbf{FP}^n_{\mathcal{I}_1,\mathcal{P}_1}(M_1) + p_2 \cdot \mathbf{FP}^n_{\mathcal{I}_2,\mathcal{P}_2}(M_2)\right).
\end{aligned}
$$

Here, step 1 follows by the definition of $\mathbf{FP}^n_{\mathcal{I},\mathcal{P}}(M)$; step 2 by an application of Lemma 1; step 3 by inductive hypothesis; step 4 by immediate calculation; last step follows by the definitions of $\mathbf{FP}^n_{\mathcal{I}_1,\mathcal{P}_1}(M_1)$ and $\mathbf{FP}^n_{\mathcal{I}_2,\mathcal{P}_2}(M_2)$. □

It remains to prove both Proposition 2 and Theorem 2. We focus on the proof of Theorem 2; the proof of Proposition 2 is simpler. First, let us introduce some technical notations.

Notation 4. *Given a set of actions $\mathcal{A} \subseteq \mathbf{A}$, we write $\mathsf{chn}(\mathcal{A})$ to denote the set $\{c\colon cm \in \mathcal{A}$ or $\overline{c}m \in \mathcal{A}\}$ of the channels used in $\mathcal{A}$. Given an execution trace t, we write $\mathsf{restr}_{\mathcal{A}}(t)$ to denote the trace resulting by removing in t all actions which are not in $\mathcal{A}$. A set of actions $\mathcal{A} \subseteq \mathsf{out}(M)$ will be called a set of* alert signals *for the CPS M if it contains all and only output actions of M (and its derivatives) on channels which are never used in input; formally, $\alpha \in \mathcal{A}$ iff $\alpha \in \mathsf{out}(M)$ and $\mathsf{chn}(\alpha) \notin \mathsf{chn}(\mathsf{inp}(M))$. We say that a channel c is* fresh *with respect to a CPS M if c is never used in M. Finally, given a CPS M we define the set of* tick*-derivatives of M as $\mathsf{der}_{\mathsf{tick}}(M) = \{M' : M \overset{t}{\Longrightarrow} \xrightarrow{\mathsf{tick}} M'\} \cup \{M\}$.*

Now, we need a definition to formalise how to compare two detection policies in term of their aggressiveness.

Let us consider a CPS M and a supervisor process Sup. Assume two sets of alert signals $\mathcal{A} \subseteq \mathsf{out}(M)$ and $\mathcal{B} \subseteq \mathsf{out}(Sup)$ such that the signals in $\mathcal{A}$ are used for communications from M to the supervisor, while those in $\mathcal{B}$ are used for communications from the supervisor to the external observer. We also assume that the supervisor Sup collects all alert signals of $\mathcal{A}$ and converts them in alert signals of $\mathcal{B}$, on communication channels fresh with respect to M. More formally,

- for all $\widehat{Sup} \in \mathsf{der}_{\mathsf{tick}}(Sup)$, whenever $\widehat{Sup} \overset{t}{\Longrightarrow} \xrightarrow{\mathsf{tick}} \widehat{\widehat{Sup}}$, with $\mathsf{tick} \notin t$, then $t = c_1 m_1 \cdots c_h m_h \overline{d_1} m_1' \cdots \overline{d_l} m_l'$, for $\overline{c_j} m_j \in \mathcal{A}$ and $\overline{d_k} m_k' \in \mathcal{B}$; we call this property *signal conversion*;
- all channels in $\mathsf{chn}(\mathcal{B})$ are fresh with respect to M.

Now, if Sup is able convert all alert signals in $\mathcal{A} \subseteq \mathsf{out}(M)$ to fresh alert signals in $\mathcal{B} \subseteq \mathsf{out}(Sup)$ then we can compare the detection policies of the CPS M and the supervised CPS $(M \parallel Sup)\backslash\mathsf{chn}(\mathcal{A})$, in terms of their aggressiveness. Formally,

Definition 12 (Detection policy comparison)**.** *Let $M \in \mathcal{M}$ be a CPS with attacker's goal indicators $\mathcal{I}$, alert signals $\mathcal{A} \subseteq \mathsf{out}(M)$ and detection policy $\mathcal{P}$. Let Sup be a pure-logical process, with detection policy $\mathcal{P}_{Sup}$, and collecting all alert*

signals $\mathcal{A}$ converting them in alert signals $\mathcal{B} \subseteq \mathsf{out}(Sup)$, on fresh communication channels. Let $\sim$ be a binary operator to compare real numbers: $\sim \in \{\leq, =, \geq\}$.

We write $\mathcal{P}_{Sup} \sim \mathcal{P}$ when for all $(M' \parallel Sup')\backslash\mathsf{chn}(\mathcal{A}) \in \mathsf{der}_{\mathsf{tick}}((M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A}))$ and tick-derivatives M'' of M', for which $M' \stackrel{t}{\Longrightarrow} \xrightarrow{\mathsf{tick}} M''$, with $\mathsf{tick} \notin t$, there is a trace $Sup' \stackrel{t'}{\Longrightarrow} \xrightarrow{\mathsf{tick}} Sup''$, $\mathsf{tick} \notin t'$, such that:

- $\mathsf{restr}_{\mathcal{A}}(t) = \overline{c_1}m_1 \cdots \overline{c_h}m_h$, *for some* $h \in \mathbb{N}$
- $t' = c_1 m_1 \cdots c_h m_h \cdot \overline{d_1}m'_1 \cdots \overline{d_l}m'_l$, *for some* $l \in \mathbb{N}$
- $\min(1, \sum_{j=1}^{l} \mathcal{P}_{Sup}(\overline{d_j}m'_j, i)) \sim \min(1, \sum_{j=1}^{h} \mathcal{P}(\overline{c_j}m_j, i))$ *for all attacker's goal indicators* $i \in \mathcal{I}^-$.

In order to prove Theorem 2, we need a preliminary result, stating the relationships between the attackers' progresses associated with the traces of a CPS M and those associated with the traces of $(M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A})$, where process Sup collects all alert signals in $\mathcal{A}$ and converting them in alert signals in $\mathcal{B}$, on fresh communication channels.

Lemma 2. *Let $M \in \mathcal{M}$ be a CPS with attacker's goal indicators $\mathcal{I} = \{(i_1, r_1), \ldots, (i_k, r_k)\}$, alert signals $\mathcal{A} \subseteq \mathsf{out}(M)$, and detection policy $\mathcal{P}$. Let Sup be a pure-logical process with detection policy $\mathcal{P}_{Sup}$, collecting all alert signals in $\mathcal{A}$ and converting them in alert signals in $\mathcal{B}$, on fresh communication channels. Assume that $\mathcal{P}_{Sup} \sim \mathcal{P}$ for some operator $\sim \in \{\leq, =, \geq\}$. Then, for $\mathcal{P}' = (\mathcal{P} \parallel \mathcal{P}_{Sup}) \setminus \mathsf{chn}(\mathcal{A})$, the following two properties*

1) $M \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}} \gamma_1$

2) $(M \parallel Sup)\setminus\mathsf{chn}(\mathcal{A}) \xrightarrow{q'_1,\ldots,q'_k}_{\mathcal{P}'} \gamma_2$

are equivalent when $q'_m \sim q_m$, for $m \in 1..k$, and γ_1 and γ_2 are two distributions such that for any CPS M' there exists a pure logical process Sup' for which it holds that $\gamma_1(M') = \gamma_2((M' \parallel Sup') \setminus \mathsf{chn}(\mathcal{A}))$.

Proof. We prove that 1) implies 2); the proof of the other implication is similar. By Definition 9, the transition in 1) is derived from a trace of the form $M \stackrel{t}{\Longrightarrow} \xrightarrow{\mathsf{tick}} \gamma_1$, with $t = \alpha_1 \cdots \alpha_n$ and $\alpha_1, \ldots, \alpha_n \in \mathbf{A}\backslash\{\tau,\mathsf{tick}\}$, such that for all $m \in 1..k$ we have $q_m = \min(1, \sum_{j=1,\alpha_j \in \mathsf{out}(M)}^{n} \mathcal{P}(\alpha_j, i_m))$. Let us assume $\mathsf{restr}_{\mathcal{A}}(t) = \overline{c_1}m_1 \cdots \overline{c_h}m_h$ and $\mathsf{restr}_{(\mathbf{A}\setminus\mathcal{A})}(t) = \beta_1 \cdots \beta_{n-h}$. As $\mathcal{P}_{Sup} \sim \mathcal{P}$, it follows that:

1) there is an execution trace $Sup \stackrel{t'}{\Longrightarrow} \xrightarrow{\mathsf{tick}} Sup'$ such that $t' = c_1 m_1 \cdots c_h m_h \cdot \overline{d_1}m'_1 \cdots \overline{d_l}m'_l$;
2) for $m \in 1..k$, we have $\min(1, \sum_{j=1}^{l} \mathcal{P}_{Sup}(\overline{d_j}m'_j, i_m)) \sim \min(1, \sum_{j=1}^{h} \mathcal{P}(\overline{c_j}m_j, i_m))$.

Since processes evolve to point distributions, the execution trace $Sup \stackrel{t'}{\Longrightarrow} \xrightarrow{\mathsf{tick}} Sup'$ is actually the following: $Sup \stackrel{t'}{\Longrightarrow} \xrightarrow{\mathsf{tick}} \overline{Sup'}$. Thus, from $M \stackrel{t}{\Longrightarrow} \xrightarrow{\mathsf{tick}} \gamma_1$ we derive $(M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A}) \stackrel{t''}{\Longrightarrow} \xrightarrow{\mathsf{tick}} \gamma_2$, for a trace t'' of the form $\beta_1 \cdot \cdots \beta_{n-h}\overline{d_1}m'_1 \cdots \overline{d_l}m'_l$, and γ_2 satisfying $\gamma_1(M') = \gamma_2((M' \parallel Sup') \setminus \mathsf{chn}(\mathcal{A}))$. Then, by Definition 9 we derive $(M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A}) \xrightarrow{q'_1,\ldots,q'_k}_{\mathcal{P}'} \gamma_2$, for some $q'_1, \cdots q'_k$. In order to complete the proof, it remains to prove that $q'_m \sim q_m$, for all $m \in 1..k$. The result is obtained by an application of the following algebraic reasoning:

$$
\begin{aligned}
q'_m &= \min(1, \textstyle\sum_{j=1}^{l} \mathcal{P}'(\overline{d_j}m'_j, i_m) + \sum_{\substack{\beta_j \in \mathsf{out}((M \parallel Sup)\setminus\mathsf{chn}(\mathcal{A})) \\ j=1\ldots n-h}} \mathcal{P}'(\beta_j, i_m)) \\
&= \min(1, \textstyle\sum_{j=1}^{l} \mathcal{P}_{Sup}(\overline{d_j}m'_j, i_m) + \sum_{\substack{\beta_j \in \mathsf{out}(M) \\ j \in \{1,\ldots,n-h\}}} \mathcal{P}(\beta_j, i_m)) \\
&\sim \min(1, \textstyle\sum_{j=1}^{h} \mathcal{P}(\overline{c_j}m_j, i_m) + \sum_{\substack{\beta_j \in \mathsf{out}(M) \\ j \in \{1,\ldots,n-h\}}} \mathcal{P}(\beta_j, i_m)) \\
&= \min(1, \textstyle\sum_{\substack{\alpha_j \in \mathsf{out}(M) \\ j=1,\ldots,n}} \mathcal{P}(\alpha_j, i_m)) \\
&= q_m
\end{aligned}
$$

where: step 1 follows from $t'' = \beta_1 \cdot \cdots \beta_{n-h}\overline{d_1}m'_1 \cdots \overline{d_l}m'_l$; step 2 follows because $\mathcal{P}' = (\mathcal{P} \parallel \mathcal{P}_{Sup}) \setminus \mathsf{chn}(\mathcal{A})$, thus $\mathcal{P}_{Sup}(\overline{d_j}m'_j) = \mathcal{P}'(\overline{d_j}m'_j)$ and $\mathcal{P}(\beta_j, i_m) = \mathcal{P}'(\beta_j, i_m)$, where the last equality holds because no $\beta_j \in \mathsf{out}(M)$ uses channels in $\mathsf{chn}(\mathcal{A})$; step 3 follows by $\min(1, \sum_{j=1}^{l} \mathcal{P}_{Sup}(\overline{d_j}m'_j, i_m)) \sim \min(1, \sum_{j=1}^{h} \mathcal{P}(\overline{c_j}m_j, i_m))$; step 4 derives because $\mathsf{restr}_{\mathcal{A}}(t) = \overline{c_1}m_1 \cdots \overline{c_h}m_h$ and $\mathsf{restr}_{\mathbf{A}\setminus\mathcal{A}}(t) = \beta_1 \cdots \beta_{n-h}$; step 5 follows because $t = \alpha_1 \cdots \alpha_n$. □

Now, everything is in place to prove Theorem 2.

Proof of Theorem 2. We give the proof for the metric $\mathbf{FP}$. The proof for the metric $\mathbf{FN}$ is similar. We reason by induction on $n \in \mathbb{N}$. The base case, $n = 0$, is immediate. Let us consider the inductive step, $n > 0$. Here, we write N to denote the CPS $(M \parallel Sup) \setminus \mathsf{chn}(\mathcal{A})$, and N' for CPSs of the form $N' = (M' \parallel Sup') \setminus \mathsf{chn}(\mathcal{A})$, where $M' \in \mathsf{der}(M)$ and $Sup' \in \mathsf{der}(Sup)$. The desired result derives from the following algebraic reasoning:

$$
\begin{aligned}
& \mathbf{FP}^{n}_{\mathcal{I},\mathcal{P}'}(N) \\
= \quad & \min\left(1, \max_{N \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}'}\gamma} \frac{\sum_{j=1}^{k} r_j \cdot \max(0, q_j - i_j(N)) + (n-1)\cdot\sum_{N' \in \mathsf{supp}(\gamma)} \gamma(N') \cdot \mathbf{FP}^{n-1}_{\mathcal{I},\mathcal{P}'}(N')}{n}\right) \\
\sim \quad & \min\left(1, \max_{N \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}'}\gamma} \frac{\sum_{j=1}^{k} r_j \cdot \max(0, q_j - i_j(M)) + (n-1)\cdot\sum_{N' \in \mathsf{supp}(\gamma)} \gamma(N') \cdot \mathbf{FP}^{n-1}_{\mathcal{I},\mathcal{P}}(M')}{n}\right) \\
\sim \quad & \min\left(1, \max_{M \xrightarrow{q_1,\ldots,q_k}_{\mathcal{P}}\gamma} \frac{\sum_{j=1}^{k} r_j \cdot \max(0, q_j - i_j(M)) + (n-1)\cdot\sum_{M' \in \mathsf{supp}(\gamma)} \gamma(M') \cdot \mathbf{FP}^{n-1}_{\mathcal{I},\mathcal{P}}(M')}{n}\right) \\
= \quad & \mathbf{FP}^{n}_{\mathcal{I},\mathcal{P}}(M)
\end{aligned}
$$

where: step 1 follows by the definition of $\mathbf{FP}^{n}_{\mathcal{I},\mathcal{P}'}(N)$; step 2 follows because $i_j(M) = i_j(N)$ and by an application of the the inductive hypothesis; step 3 follows by Lemma 2; step 4 follows by the definition of $\mathbf{FP}^{n}_{\mathcal{I},\mathcal{P}}(M)$. □

B. Proofs of Section IV

The proofs of Propositions 4, 5, 6 and 7 follow from Theorem 1, Theorem 2 and Proposition 1.