13 Design and Run-time aspects of Secure Cyber Physical Systems

Apostolos P. Fournaris[1], Andreas Komninos[1], Aris Lalos[1], Athanasios P. Kalogeras[1], Christos Koulamas[1] and Dimitrios Serpanos[1]

Abstract: Cyber Physical Systems (CPS) combine computational and physical components enabling real world interaction. Digitization, decentralization and high connectivity as well as incorporation of various enabling technologies raise various security issues. These security concerns may affect safety, endangering assets and even human lives. This is especially true for CPS utilization in different sectors of great significance, including manufacturing or critical infrastructures, creating a need for efficiently handling relevant security issues. Including security as part of a software intensive technical system (i.e the CPS) that can be distributed and highly resilient highlights the need for appropriate security methodologies to be applied on the CPS from the engineering stage during CPS design. The efficient security related processes that are implemented at design time, have an impact on security monitoring during the CPS operational phase (at run time). Efficient and accurate security monitoring that follows security-by-design principles can be a potent tool in the hands of the CPS manager for detecting and mitigating cyber threats. Monitoring traffic and activity at the system boundaries, detecting changes to device status and configuration, detecting suspicious activity indicating attacks, detecting unauthorized activity that is suspicious or violates security policies, timely responding to security incidents and recovering from them, are issues that need to be efficiently tackled with by security monitoring. In the present chapter, we are exploring the various CPS cybersecurity threats and discus how adding security as a parameter at the CPS design phase can provide a well-structured and efficient approach on providing strong security CPS foundations. New technologies on CPS security Design are presented and emerging security directions are discussed. Furthermore, in the chapter, the different aspects of security monitoring are presented with a special emphasis on CPSs, discussing the various existing monitoring approaches that are followed in order to detect security issues at run time. Specific use-cases of CPSs in the manufacturing domain and with reference to critical infrastructures are also detailed and security requirements like confidentiality, integrity and availability are discussed

[1] Industrial Systems Institute, Research Center ATHENA, Patras Science Park, Patras, Greece

Keywords *Security by Design; Security Run-time Monitoring; Cyber Physical Systems Security, Cybersecurity, Digital Twins*

13.1 Introduction

Cyber physical systems (CPS) are characterized by the tight integration of computing, communication and control technologies (Rajkumar et al., 2010). They transverse a number of application areas ranging from manufacturing to transportation, to energy and healthcare, to name just a few. They are associated with several research domains including real time networking, real time computing, hybrid systems, wireless sensor networks, model driven development, and security (Kim and Kumar, 2012).

Cyber Physical Systems (CPS) constitute a disruptive technology applicable in many industrial domains and present strong impact on economies and social processes, on many fronts, including robotics, security, safety, and military, and across industries and applications (Serpanos, 2018), as they aim to bridge the cyber world of computing and communications with the physical world (Rajkumar et al., 2010). CPS represent complex engineered physical systems, centered around Information and Communication Technologies (ICT) to integrate, control, monitor and coordinate their operations. The advances of ICT towards interacting with the physical world actually makes CPS rather ICT systems integrated into the physical world processes and applications (Gollmann, 2012), (Humayed et al., 2017).

The application of CPSs is wide covering different domains. Such domains include critical infrastructure monitoring, control and protection (energy, water resources, communication systems, and transportation infrastructures), manufacturing, factory automation and control, building management and control, environmental monitoring, automotive systems, healthcare, and defense and military systems. Identification of end user needs, challenges and opportunities in the different application domains can advance research in CPS. Multidisciplinary collaborative research can lead towards high confidence systems characterized by compatibility, synergistic behavior and integration at all scales between cyber and physical designs (Baheti and Gill, 2011). This confidence cannot be attained, unless it stems from the careful integration of security and, by extension, reliability considerations in the design of such systems.

The inherent characteristics of CPSs raise significant security challenges. CPSs are characterized by wide geographical distribution comprising different types of sensing, actuating, computing, and control devices, usually without physical security. In several cases, those device are left unattended and unsupervised in “hostile” environment where they can be easily attacked. They have requirements with a need to react in a real time manner. There have different communication channels that may be exploited by adversaries due to the CPS feedback from the physical environment. They are characterized by distribution of management and control usually involving multiple parties. They present *System of Systems* control characteristics (Neuman, 2009a).

The peculiarities of CPS with reference to their characteristics and the high economic and societal impact stemming from potential security attacks, make reliability and security integral to their operation. In order to guarantee such properties in complex systems comprising heterogeneous devices interconnected with different communication technologies, it is imperative to have a deep understanding of related threats and vulnerabilities at the outset of the design phase. This approach leads to formal specifications describing the CPS implementation. However, this task is quite challenging as it needs to address both discrete and continuous CPS behavior. There is a strong need for security operations integration to CPS hardware structure as well as CPS software intensive operations beginning from the CPS engineering (design) phase and progressing to CPS operation phase in order to be able to achieve a high level of protection against cybersecurity attacks and to support a broad range of security activities. Applying security-by-design principles is the best approach in order to structure CPSs that have the capability to support intrinsically strong security at operational time with minimal number vulnerabilities. This also reduces the fault tolerance of the CPS system thus making it more resilient to support safety critical applications.

This security-by-design approach that lead to designs with a several supported security features, however, has no impact on security if such features are not put in good use throughout the full lifecycle of the CPS. More specifically, at the CPS regular operation, security protocols must be efficiently established at the communication level and specialized security mechanisms must be deployed and executed in order to monitor the system for cyber-threats.

Run time security monitoring utilizes program monitors in order to examine deviations between the expected and real behavior of a system. The formal specification describing the CPS behavioral model as well as machine learning algorithms can be used for the determination of the expected “good behavior” of the system. Building such program monitors is quite complex and challenging when it comes to large scale CPS of high complexity involving physical processes. Enabling technologies in the context of Industry 4.0 and the Industrial Internet of Things can be used to enhance such monitoring mechanisms. Digital Twin (Tao et al., 2018) by representing a digital model that accurately mimics its physical counterpart and evolves, is continuously updated to reflect changes in the physical world (Maurer, 2017) becomes fertile ground for detecting deviations between the physical twin and its digital counterpart.

In this chapter, we discuss emergent security and reliability challenges pertinent to CPSs focusing on the mechanisms, approaches and solutions on achieving security-by-design goals in CPS in order to support security-based software intensive functions at the CPS operational level. We discuss what are the design principles that need to be applied in order to protect communications and to achieve resilience, robustness in CPSs. We also present and analyze existing and emerging approaches on how to constantly monitor the CPS security level through appropriate cyber-threat and anomaly detection schemes considering also novel concepts like the Digital Twins technology. Finally, we also present a use case scenario that targets security in industrial/critical infrastructure CPSs and give some future research directions on the chapter's concepts.

The rest of this chapter is organized as follows: Section 2 presents CPS security challenges and deals with reliability and security by design, Section 3 addresses Run Time Security Monitoring in CPS. Section 4 deals with the use case of Industrial Control Systems (ICS), a subclass of CPS, their threats and vulnerabilities. Finally, Section 4 presents relevant research challenges and concludes the chapter.

.2 Reliability and Security by Design

Reliability and Security are two equivalent aspects of a system's operation. They are related with two different domains of thinking. While security ensures that a system is doing the *right thing*, system reliability safeguards that the system is doing the thing *in the right way*. To be characterized as secure, a system must exhibit the following three properties: confidentiality, integrity and availability. Confidentiality represents a set of rules limiting access to information. It thus ascertains that sensitive information does not reach inappropriate parties, while guaranteeing that it is accessible by the right parties. Integrity is the assurance of information consistency, trustworthiness and accuracy throughout its entire lifecycle. It thus implies that information is only modified by authorized parties. Availability is a guarantee of authorized parties' reliable access to information.

13.2.1 Threats to Cyber-Physical Systems

In the past, CPS safety and reliability have been the foremost concern, even though their security had always been recognized as important. Despite this acknowledgement, CPS security has traditionally been treated as an “afterthought” in the engineering phases of systems (Mouratidis et al., 2003), owing mostly to the fact that traditional CPS security dangers were limited, since these systems were operated in isolation from the rest of the world. The use of open networks, wireless technologies, the internet and the IoT, and the cloud, has terminated the isolation of CPS resulting to Internet based attacks being the majority of attacks after 2001 (Byres and Lowe, 2004). Unfortunately, despite the increasing connectivity of systems and realization that the connectedness requirements of modern systems also imply an increased need for system security, the application of securityconscious design practices in industrial systems remains fragmented. Modern systems are still designed without a consistent integration of security practices or requirements into the core functional requirements identified through design (Ruiz et al., 2015). In a sense, this is not overly surprising – after all, with the ever-changing nature of cyber threats, it would be impossible to design a completely secure system that could be robust against any future type of attack. However, to ensure CPS reliability and security, it is imperative that these systems are designed from the outset with a thorough understanding and consideration of the threats and challenges that at least a *current* adversary may pose.

Although the precise form of a CPS security threat may differ, and indeed, even though it is impossible to predict the novel and ingenious types of attack that may emerge in the future, the fundamental nature of a threat remains unchanged. A security threat represents a set of circumstances with the potential to cause loss or harm (Pfleeger and Pfleeger, 2006). The US National Institute of Standards & Technology defines threats more analytically as "*Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.*"(Ross et al., 2006). In a CPS, system vulnerabilities that can cause such circumstances or events, might be discerned into cyber, physical or cyber physical. Physical vulnerabilities include physical sabotage of equipment or jamming but also fault injection and side channel attacks (Fournaris et al., 2017b). Cyber vulnerability types include communications and communication protocols, software, and web-based attacks. Cyber physical vulnerabilities include interconnected devices, insecure protocols, insecure Operating Systems, Software, Replay and Injection attacks.

Summarizing, a threat is a situation where a given system is vulnerable to one or multiple *attackers*, attempting to gain access to one or more components of the *system*, in order to carry out an unauthorized *objective*. Through a review of existing literature, Lei et al. (Lei et al., 2018) propose a concise taxonomy of threats to the security (and by extension, reliability) of CPS, in which a specific attack can be categorized from three major perspectives, namely its origination, purpose and target. Furthermore, in (Humayed et al., 2017), we also find a consideration about the *consequences* of a successful attack.

These definitions may seem somewhat abstract, but understanding the fundamental components of a threat, allows a system engineer to begin to integrate security aspects as an intrinsic part of the requirements capture process and subsequent design choices. As a result, while designing around the requirements of a system, an engineer might begin to consider security by asking *Who* (i.e. which entities), *Why* (i.e. what objective these persons might have) and *How* (i.e. what tools or exploits will they attempt to use). The estimated consequences of an attack (we could call this the *What* aspect*),* allows us an engineer to identify the individual components that merit security considerations, classify the severity of each threat and therefore prioritize and inform the design work required to counter these threats (Figure 13.1). Such consequences may affect the operation of a system asset (technical consequence), business aspects (e.g. loss of confidential information, inability to respond to customer demands) and also psychological consequences both within the organization operating the CPS (in-house morale) and outside it (e.g. investor relationships, public sentiment).

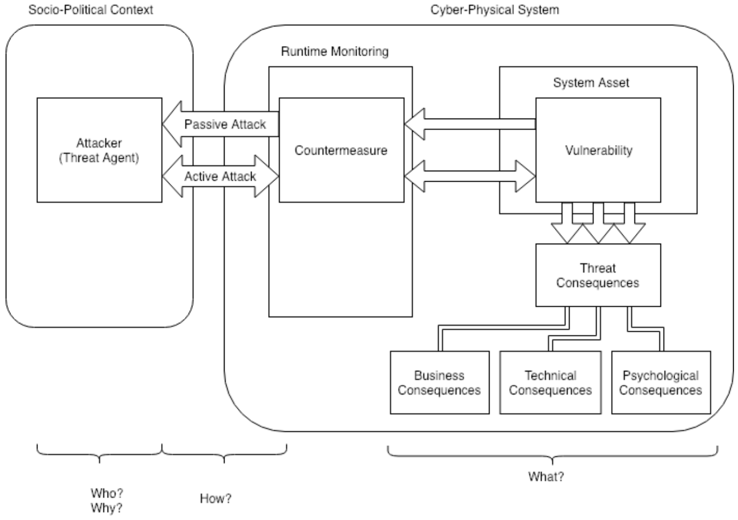

Fig. 13.1. Overview of CPS security and runtime monitoring environment. At the design phase, engineers should factor the Who, Why, How and What aspects in the specification of system requirements. Answers to these questions feed into the design and implementation of runtime countermeasures.

Concerning the "*Who?*" aspect, a CPS engineer should be aware that attacks can be adversarial, initiated by malicious insiders and outsiders, i.e. entities (persons or organizations) with authorized or unauthorized (illegitimate) access to the CPS. We often think of other humans as "attackers" of a system, and indeed this is a significant source of anxiety since we cannot always predict who might be interested in initiating an attack, but humans are not the only source of threats. Accidental damage caused by legitimate components (e.g. a buggy software update that may inadvertently wipe, or corrupt data under very specific and rare contexts), environmental factors including natural or man-made disasters ("act-of-God" phenomena), or simple failures, can also be the source of a threat that can affect security and reliability. Great heterogeneity of CPS components is also a source of vulnerability as they generally come from different vendors, following different paths of specification, design, implementation and integration, involving different entities. This greater level of integration of components that CPS are composed of, brings forth the inherent vulnerabilities of all constituent components (Ericsson, 2010). Thankfully, these non-human sources are easier to predict (and thus guard against).

In respect to the "*Why*" aspect, the motivation of attackers can be classified into cybercrime, cyber espionage, cyber terrorism or cyber war (Setola, 2011). Further threats move beyond the realm of the technical and into the sociopolitical domain (e.g. societal and legal reactions towards increased automation, surveillance, data gathering etc.). As such, motives might include personal factors (e.g. disgruntled employees), political factors (e.g. protest against government actions, retaliation against physical attacks) and socio-cultural factors (e.g. anniversaries of important or historic events). Answers to the "why" question help a system engineer understand the context of attacks, and perhaps even anticipate their timings (Gandhi et al., 2011).

The "*How*" aspect defines the type of attack, through the medium, tools and techniques that are used to carry out the attack. This is more formally defined as the *attack vector*. Attacks can be categorized as passive or active, in the sense that passive attacks attempt to obtain access to sensitive information without affecting the operation of the CPS, while active attacks explicitly aim to affect the CPS resources and modify the operation of the system (e.g. bringing nodes offline, or maliciously modifying their behaviour). Alternatively, an attack can be classified according to the system asset that it targets. Interception attacks aim to obtain unauthorized access to information (e.g. via keylogging, packet sniffing, side channel leakage exploitation etc.) and are, by their very nature, passive. Modification attacks target the system's integrity, by modifying control signals or sensor values for example. Interruption attacks target the system resource availability (e.g. Denial of Service Attacks), by disrupting communication, modifying software or deleting data. Finally, fabrication attacks target the system's operation by inserting non-authenticated operations, e.g. fake control signals and transactions. These three latter types of attack are all considered as active attacks. These active attacks can further be viewed under two typical scenarios, random attacks (in which, for example, an intruder blindly injects or modifies information into the CPS) and targeted attacks, where the attacker attempts to affect specific components or states of the CPS (Ding et al., 2017).

13.2.2 Approaches to preventing and detecting CPS attacks

As discussed previously, compared to pure software systems, the design of a CPS needs to be concerned with a mixture of physical, cyber and cyber-physical threats. To mitigate the effects of such attacks, a CPS may be designed so as to implement preventative measures at both the physical level (e.g. restricting physical access to certain assets, monitoring employee, contractor or asset presence, maintaining backup copies in physically remote locations) and also the cyber levels (e.g. enforcing strong cryptography, software certification and user/service authentication measures). Some measures involve a hybrid cyber-physical approach, for example, securing physical access to assets can be implemented using both physical equipment (e.g. gates, door locks) and cyber equipment (e.g. RFID employee badges, smart locks, sensor equipment). Other examples might include the physical location of networking equipment, which in turn, dictates the type of networking equipment that can be used, and thus the type of cyber security mechanisms that need to be implemented on top of the network layer. For example, at the design phase, the engineer may have to choose between wired and wireless communication technologies for certain assets, depending not just on the physical properties of the environment where these assets might be located, but also on the likelihood of potential types of attack that exploit weaknesses of the network types. In the next sections, we discuss a few core considerations during the design of secure CPSs and present common approaches that the engineer might consider during the design phase.

13.2.2.1 Prevention strategies

Communication networks are perhaps the most obvious attack target for connected CPSs. The obvious approach is to prevent such attacks by strictly securing all communications, data and access to various subcomponents, but this approach introduces problems at scale, given the required complexity and generated overheads. This is because many of the participating IoT devices (such as sensors) do not have the computing capability required to implement such measures, or at the very least, not with the required lack of delay (Zhang et al., 2016). Even more, given the heterogeneity of large-scale CPSs, the different operating characteristics and configurations (often left at “defaults”) of commercial-off-the-shelf components that make up parts of a CPS can be easily exploited for attacks, or equally may make it difficult to integrate them into a comprehensive security framework (Humayed et al., 2017). To circumvent the problem, it has been demonstrated that securing intelligently selected subsets of data, or strategically introducing secure infrastructure at key areas of a CPS, like hardware security tokens (Fournaris et al., 2017a), may act as a strong deterrent, especially in large scale systems such as power grids, since such approaches make it extremely hard and impractical for an attacker to effect more than small and inconsequential compromises to the system’s integrity (Kim and Poor, 2011).

13.2.2.2 Detection strategies

The design and integration of preventive strategies can add a strong deterrent to the security of a networked CPS, but it is not enough to guarantee reliable operation of a system. For example, an insider attack from a corrupted employee can completely override any network security measures and is not preventable with such means. Organized, motivated and well-funded attackers may be able to find ways to penetrate the security of a CPS network, but there is a range of attacks which do not require penetration of a network’s security (e.g. DoS through flooding), and which are able to wreak havoc on even the most heavily secured system. To secure a CPS and ensure its reliability, engineers must integrate detection mechanisms at the design phase, to continuously monitor the operation of the system and identify potential breaches of security, or threats to the reliable operation of the system, ideally in real-time.

In the design of CPS, mathematical modelling of the effects of various attack types can be useful in order to derive monitoring schemes that can detect these attacks in real time. For example, network performance degradation, such as observed in denial of service attacks, is well studied in terms of its modelling and is therefore easily detectable (Pang et al., 2011)(Amin et al., 2013)(Befekadu et al., 2015). Other types of network attack are harder to detect and defend against, for example, in (Lee et al., 2014), a framework is presented for detecting wormhole attacks. These attacks target the control signals relayed between distant sites over wireless networks, by establishing links that appear to be authentic, but in reality, work by delaying signals, or replaying previous signals. This type of attack is not preventable by cryptography, since the delayed or replayed messages are all valid.

When attackers are able to bypass network security and gain access to a networked CPS, there are still measures that can be implemented from the design phase, which can help ensure reliable operation of the system. A frequent objective of attack after penetration is to enact interruption or fabrication types of attack, threatening the integrity of CPS data. It is possible to implement simple yet effective detectors of bad data, either using simple thresholds (which the attacker cannot know in advance) or by detecting significant deviations from the expected reported states (Mo et al., 2010). Even with such measures, small changes effected by attackers may incrementally mount to large consequences in the operation of CPSs, and still, an attacker might adopt conservative strategies to minimize the chances of being detected. As such, it is apparent that real-time detection should ideally be paired with longitudinal monitoring of system behaviour, in order to detect such cumulative effects on the system.

13.2.2.3 Implementing CPS security strategies

More recently, the rise of popularity (and accessibility) of machine learning tools has led to the recommendation for applying such techniques (especially deep learning neural networks) to detect reliability or security issues (Kriebel et al., 2018). One drawback of these approaches is that although a trained classifier can work in real-time to detect threats, on the cloud, fog, or even edge level (Mamdouh et al., 2018), the training process has to be performed typically offline, and particularly so when the training data consists a large volume. Hence such classifiers cannot be re-trained online and require multi-tier architectures (Khorshed et al., 2015) for their implementation (e.g. online for detection, near-line for model tuning, off-line for training).

Other approaches, such as (Singh et al., 2016) are more concerned not with detecting attacks, but mitigating the results of attacks by implementing a better control mechanism that is robust to a variety of network effects, whether these effects occur naturally (e.g. dynamic inadvertent changes in network operating conditions) or occur maliciously (e.g. certain types of attack).

The introduction of trusted computing as part of the security-by-design approach can also provide a proactive countermeasure against possible attacks on CPS devices. Latest processor technologies provide trusted execution environment (TEE) generation that can be used for security sensitive software execution. Such execution environments cannot be accessed by attackers to install malicious code or alter existing software code since all activities are monitored. For example, ARM offers the ARMTrustzone TEE for all its cortex A and in some of its cortex M processor family. Dedicated hardware tokens can also be placed in non-embedded system CPS devices like Trusted Platform Modules (TPMs) in order to instill security and trust on control management subsystems of a CPS (Fournaris and Sklavos, 2014).

13.2.3 Challenges to securing Cyber-Physical Systems

There are several open problems and research challenges associated with security of CPS. First of all, there is a need to consider CPS security aspects throughout their lifecycle. This means that security should be an intrinsic property of CPS right from its design phase, rather than an issue to be dealt with at a later stage, e.g. for mitigating a potential attack. Secondly, CPS security design has to consider both cyber and physical system aspects. The combination of cyber security and physical system theoretic security, results in better guaranteeing CPS security as both approaches are incomplete and present drawbacks (Mo et al., 2012). Yet, this approach requires a change in design principles so as to address both the cyber and physical worlds. Thirdly, the real time nature of CPS security needs to be addressed. Real timeliness of decision making is critical in ascertaining survivability of CPS in the case of an attack. Ensuring attack resilience of CPS mandates taking into account the physical and cyber world interactions, during the design phase (Cárdenas et al., 2011). A fourth research challenge is associated to CPS change management. CPS represent complex systems that comprise a big number of components and systems. Changes in CPS, for instance changes in hardware, device mobility, software updating and patching or addition of capabilities, often result to different overall systems. It is a challenge to ascertain that no new vulnerabilities are introduced and that the CPS security assumptions remain valid as well as ascertain that updates are not tampered and are transmitted to CPS devices in a secure way.

Despite technical progress such as reported above, and as can generally be found in the domain of cryptography, system security and network security, a CPS introduces fundamental challenges to the holistic aspect of security and reliability, given its very nature. Starting off with the fact that we should not forget the *physical* aspect of CPSs, it is imperative that physical security measures (e.g. gates, barriers, locks, security personnel, access protocols and policies) are designed, tested and implemented. These physical security measures are by no means perfect and undefeatable, and their compromise can quickly lead to mounting threats in the *cyber* aspect of CPSs (e.g. manual installation of rootkits that compromise cyber-security). Still, one challenge to be faced in the transition from classical industrial control systems to dynamic CPSs, is the fact that older CPSs assumed isolation from the external world (hence burden was more heavily focused on physical, rather than cyber security). These legacy systems will inadvertently form part of extended, interconnected CPSs (until such time at least as they are upgraded or replaced) and therefore present an inherent security risk, since they rely on unsecure software and connection protocols (since the principle of isolation did not mandate more sophisticated approaches) (Humayed et al., 2017).

In this regard, it should be considered that a CPS may be subject to not just single types of attack, but needs to be secured against multiple types of simultaneous attacks. Given its scale, it is plausible that attackers might chose to adopt a low-detectability, low-consequence form of attack, whose effects might however amount to large cumulative problems in the operation of CPSs. Scale also aggravates the problem of attack detection, since network effects and the nonlinear dynamics of networked, adaptive and self-configurable operations might signal unexpected deviations from normal operating parameters, which might be confused for attacks. The latter aspect of networked, adaptive and self-configurable operation of CPSs (especially in the context of Industry 4.0 and distributed manufacturing), demonstrates the difficulty in developing robust, accurate and integral formal models for the simulation of such systems. Quite simply, the dynamic configuration of available components, operating modes and context-sensitive goals of CPSs, make the modelling of any non-trivial such system a daunting prospect.

On the positive side, this dynamic complexity of adaptive and self-configurable CPSs, makes it hard for attackers to carry out targeted attacks, since this typically requires an intimate knowledge of the system’s configuration and operations (Cárdenas et al., 2011). This aspect can work well to the advantage of defending such systems, since it has been demonstrated that prior knowledge about a system’s configuration (physical model) can significantly help in identifying the most critical sensors and attacks (Cárdenas et al., 2011), and of course, a-priori knowledge of this configuration can help to dynamically identify and control security aspects during operation.

13.2.4 Designing security and reliability into CPSs using Digital Twins

At this time, given the relative infancy of large-scale networked CPSs, it is not surprising that attempts at defining engineering and design processes for incorporating security into the design of such systems, are very limited. In (Neuman, 2009a) it is argued that for CPSs, all communication channels within an application are enumerated and analysed for security constraints, under a domain-specific understanding that includes both physical (sensor) and external (human operator or third-party control) process channels. This analysis feeds into the relatively modern concept of a "Digital Twin": an online, cyber model of the physical processes taking place in a CPS, which can be used to simulate outcomes at decision points, or monitor the operation of a CPS in real time. In this sense, a digital twin can incorporate many of the on-line threat detection and prevention measures discussed above. To become a successful tool towards this end, a digital twin cannot be statically defined; instead, it needs to be dynamically generated and adapted using environment specification. Multiple such twins can be produced for various purposes: control, monitoring and testing, the latter particularly important for security engineers, who can exploit these models to test CPS resilience against a variety of attack vectors, without fear of risking the real CPS in operation (Eckhart and Ekelhart, 2018a). Even better, the Digital Twin is something that can evolve along with the design of a CPS. As various options are explored in the design phases, they can be modelled and trialled in a partial Digital Twin model, thereby informing the design process and help engineers make better decisions in the context of security and reliability, as they work towards designing the full system.

Although the concept of a Digital Twin is promising, considering the assistance it may offer to engineers throughout the whole lifecycle of a CPS (design, operation, maintenance), there are some issues that need to be considered. The main challenge in producing a Digital Twin for a CPS, is that engineers need to fully document the sub-systems that comprise the CPS using a parseable formal specification language (e.g. AutomationML). Changes to existing components must also be reflected in their specification, to ensure model validity. This challenge is significant, as it requires considerable additional effort at the design stage of a CPS, and also there is no guarantee that components (e.g. third-party services, off-the-shelf assets) that are later added to the CPS will come with such documentation, therefore it might be challenging to integrate these new components to an existing Digital Twin model. Even if a Digital Twin model was able to fully capture the operation of a CPS, the model cannot display complete fidelity to the real world, since observed effects in real-world operation are often non-linear and chaotic, hence may not be so accurately modelled (e.g. signal noise, network latency, power grid fluctuations etc.).

Further from preliminary steps towards the specific use of digital twins as tools for security and reliability analysis, little other progress has been made in generalizing the use of Digital Twins as engineering design and lifecycle monitoring tools, as acknowledged in (Bécue et al., 2018). However, it is envisioned that digital twins that act as a testbed for real large-scale CPSs can enhance the potential of identifying and correctly applying strategic security approaches, possibly through carrying out adversarial exercises (red team vs blue team, attackers vs. defenders) across the security engineers and personnel involved in the operation of a CPS. In that sense, the CPS users can be trained on the security aspects of their CPS using trial and error without affecting the reliability of the actual system. Also, CPS Digital Twins can be used in order to test new security policy effectiveness before such policies are applied to the real system thus considerably helping the construction of a tailored-made, concrete security policy for each CPS at hand.

The digital twin concept will undoubtedly feedback results of such exercises into the design of CPSs, helping to improve their configuration. At a time where it is uniformly accepted that our knowledge about how to best configure and secure large-scale adaptive CPSs is rather limited, the digital twin can be seen not just as a tool for post-hoc evaluations of system security and reliability, but potentially as an integral part in the incremental design process of robust CPSs, from their very core conceptual phases.

13.3 Runtime Security Monitoring

Considering the security threats and challenges that CPS have, as those are described in the previous section, it becomes obvious that there is a considerable need to continuously monitor a CPS during its regular operation for security anomalies that can result to some security attack. Typical ICT systems have a series of well-developed tools that by combining a wide range of technologies and methods can detect, respond and mitigate security attacks. The generic category of run-time monitoring systems may comprise of various components like intrusion detection systems (IDS), zero-vulnerability malware detectors and anomaly detectors that are all interconnected under a security information and event management (SIEM) system. The SIEM is usually responsible for the correlation between various events and logs to extract security alerts and make attack mitigation suggestions. However, a CPS runtime security monitoring system must consider the CPS specificities that, in several cases, are distinctly different than those of a typical ICT system.

According to (Mitchell and Chen, 2014), there are four basic characteristics that distinguish CPSs from typical ICT systems in terms of runtime security intrusion detection: physical process monitoring, Machine to Machine communications, heterogenity and legacy system interactions. Due to their connection between the cyber and the physical world, the CPS devices measure physical phenomena and perform physical processes that are governed by the laws of physics. Thus, a CPS security monitoring system must perform physical process monitoring, using physical laws as a control mechanism to model and predict valid instructions and outcomes. Furthermore, many CPSs application scenarios are highly focused on automation and time driven processes that realize closed control loops, that do not require human intervention (and its associated unpredictability). This kind of behavior focused on Machine to Machine communications, increases the regularity and predictability of the CPSs activities. The CPS security monitoring system should be able to monitor regularly closed control loops. Thirdly, the attack surface of a CPS is considerably broader than that of an ICT system. CPSs consist of many heterogeneous subsystems and devices while they follow a broad range of different, not ICT related, control protocols like ISA 100, Modbus, CAN etc. Some of these devices and protocols have proprietary software or standards that may constitute ICT attacks unfitting. This characteristic, along with the fact that a successful CPS attack has high impact and thus high payoff, attracts very skilled attackers that can mount very sophisticated attacks on CPSs (Fournaris et al., 2017b). Such attacks are usually very hard to discover and document since typical ICT intrusion detection software cannot identify them (e.g. the attacks may not be IT related but rather OT related). Attackers exploit CPS zero-day vulnerabilities which would render many ICT security monitoring toolsets useless (e.g. knowledge-based ones (Mitchell and Chen, 2014)).

Lastly, many CPS include legacy hardware that is difficult to modify or physically access. Such components may be partially analog, have very limited installed software resources and be dictated by physical processes. The challenge here is how to install security monitoring sensors on such devices and how to predict/model their behavior correctly in order to detect possible anomalies. It needs also to be considered that legacy devices do not have many computational resources and it becomes hard for the monitoring system to retain its real-time responsiveness when collecting security metrics from them.

Runtime Security monitoring in the CPS domain, considering the above specificities can take various forms. However, they all rely on two core functions, the collection of data from various CPS sources and the analysis of data in a dedicated runtime security monitoring subsystem. To achieve appropriate data collection, the security monitoring system must deploy security agent software/hardware (Fournaris et al., 2018) on the monitored CPS devices, or introduce virtual entities (Virtual Machines) for data collection (Eckhart and Ekelhart, 2018a) within the CPS infrastructure. Examples of collected data can be Syslog log events, system call logs, traffic recordings from network interfaces, reputation scores, processing loads, connection/communication failures etc. All collected data are analyzed in the CPS runtime security monitoring system that uses data mining, machine learning, pattern recognition or statistical data analysis to extract metrics on security issues that may take place inside the CPS at runtime. Such issues may be possible incidents detected via data that can be binarily characterized as bad/good, or continuously characterized by a specific significance grade. The performance of the security monitoring system is measured by the False Positive Rate (FPR), the False Negative Rate (FNR) and the True Positive Rate (TPR). The system is also measured in term of incident detection latency and consumed resources number, computational overhead, excessive network traffic and power consumption (Mitchell and Chen, 2014).

To better understand the monitoring/detection approach that runtime security monitoring systems follow, we can broadly identify two approach categories, knowledge-based detection, and behavioral based approaches. In a knowledge-based security monitoring system runtime, features that are extracted from collected data are matched with a specific profile pattern or model. Alarms are raised when there is a behavior mismatch with the existing profiles or models. This approach may lead to low FPR, but needs a very well described profile or model to be effective (e.g. an attack dictionary, a CPS device functionality pattern) since it relies on identifying a specific pattern/model.

On the other hand, behavior-based security monitoring systems do not rely on a specific prescribed knowledge but rather look for runtime features that seem out of the ordinary and act as outlier values to the expected behavior of a CPS. Supervised, semi-supervised or unsupervised machine learning algorithms can be employed on this approach. As expected, in supervised and semi-supervised algorithms a predefined training set must be constructed in such a way that it reflects accurately the expected CPS behavior. Given the CPS specificities, this is a non-trivial task. It takes a lot of time an effort to structure such a dataset (e.g. using state-of-the-art feature analysis, discovery and engineering techniques) and still the behavior-based monitoring may result in high FPR. Unsupervised behavior-based monitoring does not need a pre-structured training set and creates the dataset using CPS live data (Mitchell and Chen, 2014). The behavior pattern that the above approaches evaluate can be a deviation from good behavior or a match to bad behavior (Khan et al., 2016). Bad behavior matching monitors detect attacks by building profile(s) of known bad system behavior, such as statistical profiles of attacks (Hodge and Austin, 2004) (Paxson, 1998). Such monitors are robust since machine learning techniques tend to generalize from the presented data (Khan et al., 2016). On the other hand, good behavior deviation monitors build a statistical profile of normal (good) behavior and detect deviations from this profile (Lakhina et al., 2005) (Watterson and Heffernan, 2007). Their robustness is better than that of bad behavior monitors since their employed machine learning techniques do not rely on historical knowledge of possible attacks (Khan et al., 2016).

There are several CPS security monitoring systems that consider some of distinguishing CPS characteristics in their design, like the works in (Kane, 2015) (Koopman and Wagner, 2016), which are focused on closed control loop monitoring in autonomous computing systems and on traditional network traffic monitoring. Specifically, for industrial network runtime security monitors, there are solutions that take advantage of the physical process measuring taking place in an industrial site as well as the closed control loop processes (Qin, 2012), but they still use techniques based on traditional network traffic monitoring. For example, the ARMET (Khan et al., 2018) system can identify good behavior deviations in a reliable way that has very low FPR and FNR. ARMET can observe at runtime an application's execution, compares it against the predicted execution behavior and identifies deviations.

When it comes to security runtime monitoring based on knowledge-based approaches using models, there is a need for some model description language that can take into account CPS characteristics like real-time responsiveness (Blum and Wasserman, 1994). Barnett et al. in (Barnett and Schulte, 2003) propose the use of AsmL as an executable specification language for run-time monitoring. AsmL, an extension of Abstract State Machines (ASM), is based on the formalism of a transition system whose states are first order algebras (Börger and Stärk, 2012). In

(Chupilko and Kamkin, 2013) a full framework for executing specifications of real-time systems is proposed. This proposal can be used for security runtime monitoring in CPS timed systems.

There are very few CPS security runtime monitoring systems that provide efficiency metrics as are specified at the beginning of this section (Kane, 2015). There exist works where such results are provided but only for CPS monitoring subsystems like IDSs (Khan et al., 2016).

What needs also to be mentioned, is the fact that existing solutions on CPS security monitoring are primarily focused on detecting computational and network security incidents happening in a CPS. However, since a CPS implements closed control loops that rely on collected data for autonomic decision making, malicious attacks on the collected data can also constitute a very serious threat. Recently, effort has been invested in detecting false data injection (FDI) attacks that aim to maliciously alter the CPS control loops. Research works aiming to provide protection against FDI are focused on making efficient vulnerability analysis like the work in (Khan et al., 2016) where vulnerability to FDI is expressed as a satisfiability problem and solved using a solver that supports functions over real numbers (Gao et al., 2013) or focused on utilizing appropriately FDI fault diagnosis techniques (Rigatos, 2016) (Rigatos, 2015).

It is important here to note that the full potentials of a runtime security monitoring system are not unrelated with the security-by-design principles described in the previous section. A very important aspect of any monitoring tool is the mechanism that provides input to such a tool. As mentioned, in security monitoring tools inputs are provided by event data collection points (security agents or sensors) that are installed in various parts of a CPS. It is of prime importance that these security sensors are designed and realized in the CPS architecture during engineering phase (design time) and that they are fully integrated with the CPS architecture. Only then can such sensors maximize their efficiency (in terms of speed but also in terms of impact) on collecting all security related information that may trigger runtime security anomalies.

As will be explored in detail in the following chapters of this book, Digital Twins can enhance CPS security monitoring mechanisms. It has already been proposed as a tool to provide additional security in a CPS system by testing security components in complex CPSs (Eckhart and Ekelhart, 2018b) (Tauber and Schmittner, 2018) (Damjanovic-Behrendt, 2018). A framework providing a security-aware environment for Digital Twins is described in (Eckhart and Ekelhart, 2018b) (Eckhart and Ekelhart, 2018a), demonstrating, among others, how security and safety rules can be monitored in security-relevant use cases. The framework is extended in (Eckhart and Ekelhart, 2018c) by a specification-based, physical device state replication approach, by passively monitoring their inputs and outputs, showing successful detections of attacks against a CPS test-bed. However, there is still no concrete proposal on how to use Digital Twins of a CPS as part of a runtime security monitoring system since Digital Twins is a relatively new modelling approach and it has considerable complexity making it hard to be integrated in a monitoring tool. However, conceptually, Digital Twins can be an integral part of security monitoring since it provides a trusted environment for testing real inputs without the possible presence of malicious entities. In that sense, a Digital Twin of a CPS can act as a trusted replica of the actual system where good behavior can be modeled and evaluated as well as device patterns can be described and trusted. Following the knowledge-based or behavior-based CPS runtime monitoring approaches we can use the Digital Twin virtual environment to match collected data and its behavioral patterns with the known good behavior of the Digital Twin. Alternatively, Digital Twins can be used in order to construct in a safe, virtual environment, training data sets for machine learning algorithms that are used during CPS runtime security monitoring.

The industrial sector is beginning to understand the security benefits and potentials of Digital Twins in the industrial CPSs. As such, large companies have announced their plans to launch relevant products based on the Digital Twins concept. The announcement of General Electric's (GE) Digital Ghost which is a combination of GE's Digital Twin efforts and industrial control technologies, as a mean to thwart cyber attacks (Dignan, 2017) constitutes an indication of the above interest.

13.4 CPS Security Use cases

In this section we focus more specifically on a CPS use case, targeting Industrial Control Systems (ICS) and describe state-of-the-art approaches in the application of CPS security and reliability design for such systems, in order to concretely demonstrate the integration of the aspects discussed in the preceding chapters. An ICS is a subclass of CPS that is associated primarily with the manufacturing sector, yet is increasingly utilized for control and management of critical infrastructures. It thus covers a wide area of application use cases such as energy smart grids, transport systems, water management systems apart from the pure industrial manufacturing domain as its name implies. ICS constitute the infrastructure of the so called Operational Technology (OT), comprising control equipment (Programmable Logic Controllers (PLCs), Network Controllers (NCs), and robot controllers), supervisory control and data acquisition systems (SCADA), and their industrial networking infrastructure. OT has a different path of evolution with reference to Information Technology (IT) systems, as it addressed quite different end user needs. The interoperable convergence of OT and IT is an emerging challenge, as ICS are viewed in the context of the emergence on the Industrial Internet of Things (IIoT) both in the pure industrial manufacturing domain, and other application domains taking advantage of OT like critical infrastructures or healthcare (Serpanos and Wolf, 2017).

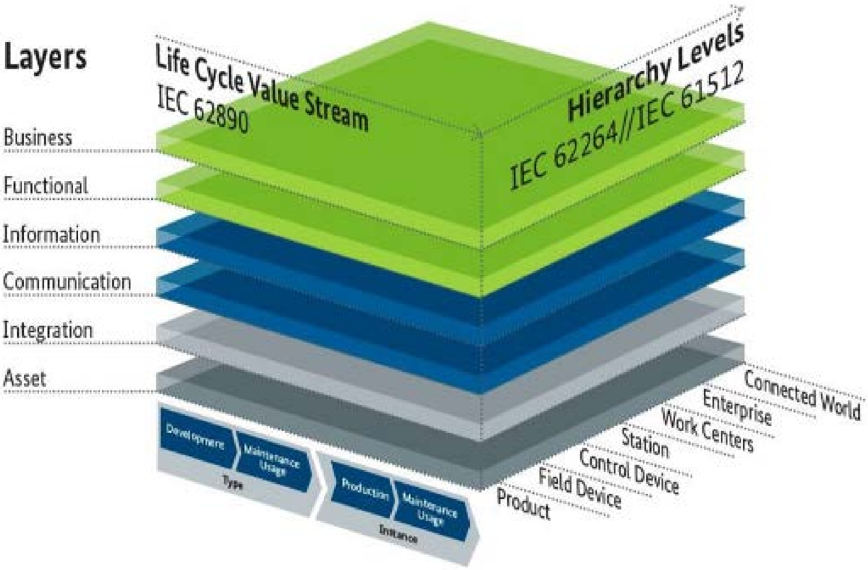

Fig. 13.2. Reference Architectural Model Industry 4.0 (RAMI 4.0) (Schweichhart, n.d.)

Industry 4.0 (Schweichhart, n.d.) is a high tech strategy of the German government, that promotes computerization in manufacturing. It tries to bridge the two worlds of OT and IT to enable higher, more flexible and efficient productivity in the manufacturing sector and more services. Its reference architecture RAMI 4.0 comprises three axes related to Hierarchy, Architecture and Product Lifecycle (Figure 13.2). The hierarchical axis actually dissolves the traditional multilevel hierarchy in the manufacturing domain to a flat and flexible hierarchy that distributes functionalities to devices and equipment in a Smart Factory producing smart products and being connected to the world.

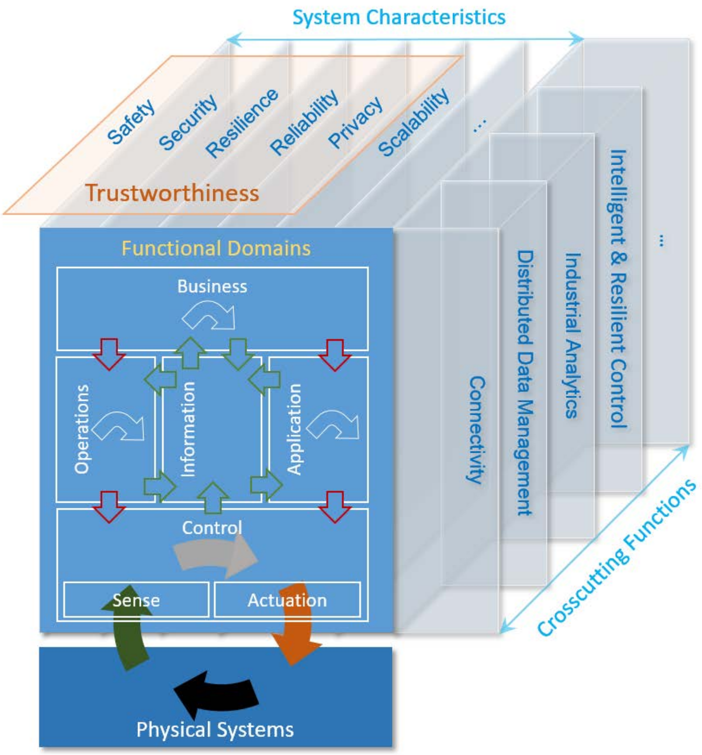

Fig. 13.3. Industrial Internet Reference Architecture (IIRA) (Lin et al., 2017a)

The Industrial Internet Consortium (Lin et al., 2017a) has also developed a reference architecture, the Industrial Internet Reference Architecture (IIRA)(Figure 13.3). IIRA is applicable to ICS that are related to different domains ranging from manufacturing to transportation to energy and healthcare. IIRA Functional Viewpoint focuses on functional components, structure and interrelation, interfaces and interactions. There are efforts for mapping between IIRA and RAMI 4.0 recognizing the commonalities between the two reference architectures (Lin et al., n.d.) (Figure 13.4).

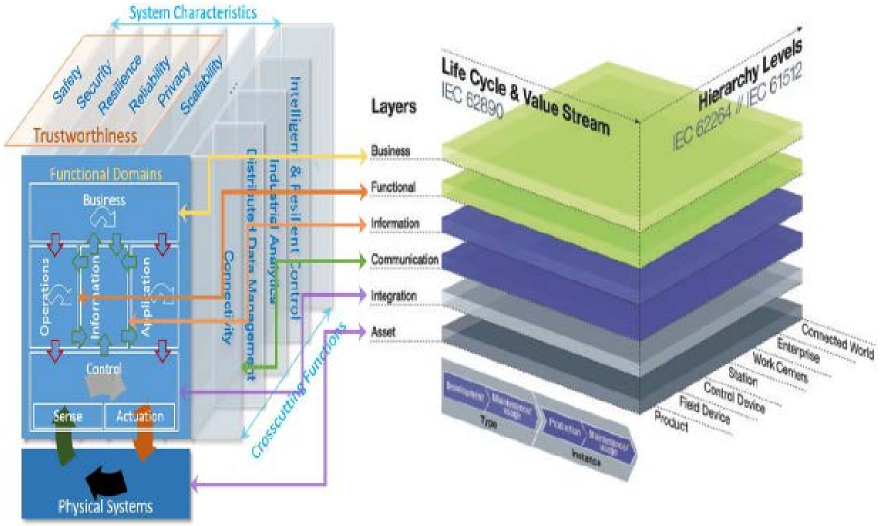

Fig. 13.4. Architecture Alignment between RAMI 4.0 and IIRA (Lin et al., n.d.)

Different types of threats are relevant for ICSs: criminal, financial, potilical, and physical. For each threat five factors may be identified, namely source, target, motive, attack vector, and consequences (Humayed et al., 2017). A criminal threat has as a potential consequence ICS application disruption of operation through its remote control utilizing wireless connectivity (vector) by an attacker (source). A financial threat leads to financial losses (consequence) of a utility (target) through false data injection or tampering (vector) by a customer (source) (Turk and others, 2005). Critical infrastructures of a targeted nation are usually attacked by a hostile nation (source) in political threats utilizing to this end access to ICS devices or malware and leading to sabotage actions or environmental destruction (Ryu et al., 2009). A physical threat is of the type of sensor input spoofing like the spoofing of optical flow cameras (target) in UAVs possible even for not sophisticated adversaries (source) (Davidson et al., 2016).

ICS present different *cyber*, *physical*, and *cyber physical* vulnerabilities. The geographically distributed nature of ICS especially with reference to Critical Infrastructure applications, e.g. road infrastructure, and physical exposure of its different components creates a physical vulnerability, as inability to physically secure each and every component makes them vulnerable to tampering or sabotage.

ICS cyber vulnerabilities include communication and software vulnerabilities. Communication vulnerabilities are related to specific protocols utilized in ICS and their inherent vulnerabilities, as well as the communication physical medium. Use of open protocols, like TCP/IP, not intended to be secure by design, raises security issues (Bellovin, 1989). Remote procedure call (RPC) protocol vulnerabilities have contributed to the Stuxnet attack (Langner, 2011). Man in the Middle attacks are a vulnerability of both wired and wireless systems (Hwang et al., 2008) (Francia III et al., 2012). False data injection, capturing traffic, attacking employee probably unsafe personal devices connected to ICS network are just a few of the vulnerabilities. A taxonomy for wireless communication vulnerabilities is presented in (Welch and Lathrop, 2003). Software vulnerabilities include SQL injection (Halfond et al., 2006) and email based attacks (Fovino et al., 2009).

ICS cyber physical vulnerabilities include communication, OS and software vulnerabilities. Communication vulnerabilities comprise vulnerabilities of the protocols used for ICS component communications. For instance, widely used in ICS Modbus protocol lacks encryption, integrity checks, and authentication measures making it vulnerable to different types of attacks (Byres and Lowe, 2004). MTF-Storm fuzzer has evaluated different Modbus implementations and identified issues with all of them (Katsigiannis and Serpanos, 2018). Direct access to ICS devices, left with default passwords (Mo et al., 2012), or being directly connected to the Internet (Leverett, 2011), or through secondary emergency communication channels (Alcaraz and Zeadally, 2013), is also an ICS communication vulnerability. OS vulnerabilities comprise real-time operating system (RTOS), used by ICS devices, and general-purpose OS, where ICS applications run, vulnerabilities. Absence of access control mechanisms in RTOS makes them vulnerable (Igure et al., 2006) to authentication and confidentiality failures. General-purpose OS vulnerabilities are exploitable for attacks as proven in the case of Stuxnet attack, exploiting Windows Print Spooler Service and Server Service vulnerabilities (Chen and Abu-Nimeh, 2011). Software vulnerabilities are associated with ICS devices reduced computational resources that limit their capacity to enforce cryptographic measures (Langner, 2011). Existing backdoors in ICS devices further facilitate attacks (Santamarta, 2012).

A solution towards ascertaining that ICS applications are free from vulnerabilities while meeting their requirements is developing reliable and secure applications by design. The challenge is to derive in a formal way the ICS application implementation starting from a declarative specification. This challenge is quite ambitious as it has to take into account both continuous and discrete system behaviors. Different approaches in literature address security by design (Yang et al., 2012) (Zhang et al., 2015) (Martinelli and Matteucci, 2007) (Matteucci, 2007) or reliability by design (Soulat, 2014). An approach for reliable and secure by design ICS applications (Khan et al., 2018) based on deductive synthesis (Delaware et al., 2015) utilizes stepwise refinements of declarative specifications. In this context an initial non deterministic specification is refined into a fully deterministic efficient, correct and secure implementation. As an example, the Coq proof assistant (Barras et al., 1997) is used to encode ICS behavior declarative specification based on an abstraction relation specification (Hoare, 1978).

The reliable and secure by design ICS applications is coupled by run time security monitoring utilizing to this end both the specification and implementation of the ICS application (Khan et al., 2016). The specification models normal behavior of ICS resources, data and control flow between their submodules as well as misbehavior (bad behavior). The divergence between anticipated normal behavior and misbehavior of submodules of an ICS is an indication of potential attacks. Hypothetical attacks can be used for diagnostic reasoning increasing robustness of the approach.

13.5 Research Directions

Part of the current challenges in certain dimensions have been initially discussed above, referring mainly to the evolution path of core CPS engineering disciplines that should widely adopt and adapt modern model-based and formal methods that should incorporate security aspects from the system specification and design phases. Looking into the core aspects of existing approaches, there are two main research directions identified as of paramount importance.

The first direction involves the formal description methods for the principal, as well as for the emerging composable system properties and the exploitation of automatic executable model transformation and component code generation methods and tools in order to create security monitor(s) from a distributed system specification. While there is a significant progress on the available knowledge and tools, these are typically applicable to provable correct autonomous computing nodes for the detection of specific attack types (Khan et al., 2018). Their applicability though to complex distributed systems is not straightforward, considering the introduction of additional system properties that are contributing to the system-wide correctness condition, as well as the introduction of additional attack surfaces that need to be handled altogether, and not on an isolated basis, both contributing to an exponential complexity increase for the detection of multiple, concurrent attacks. Furthermore, the existence of other non-functional system properties like the timeliness constraints and critical dependencies in a distributed CPS, pose substantial challenges to the engineering of robust program synthesis and code generation techniques and modules. Needless to say, the unpredictability of human-in-the-loop scenarios (Folds, 2015) adds to the complexity of formally modeling CPSs, and while CPS literature has, so far, largely avoided the subject, it is hard to see how it can be completely disregarded in many CPS use-cases, where human operators play a vital role to critical and safety-sensitive systems' behavior (e.g. intelligent transportation and aviation).

Then, there is the quest for the consolidation, or even more, the standardization of basic runtime frameworks, component libraries and subsystem interfaces that will ease the deployment of interoperable customized components into generic, domain-specific solutions and architectural frameworks (Koulamas and Kalogeras, 2018). These include the prevalent reference architectures of Industry 4.0 and IIC, and the challenge is enlarged after considering the widening of the digital twins deployment alternatives, stemming from the expected evolution of AI capabilities in embedded devices at the edge (Koulamas and Lazarescu, 2018), in contrast to the typical cloud-based paradigm of today. That is, the security designing and runtime monitoring becoming a distributed system procedure on its own terms, opening then other research challenges on the necessary computing hardware and networking hardware and software support for the isolation of the two concurrently executed distributed systems, such as the exploitation and integration of results from trusted execution environments and trusted platform modules (TPM) or from time sensitive networking (TSN) research.

Acknowledgements

This work is supported by the project “I3T - Innovative Application of Industrial Internet of Things (IIoT) in Smart Environments” (MIS 5002434) implemented under the “Action for the Strategic Development on the Research and Technological Sector”, funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

References

Alcaraz, C., Zeadally, S., 2013. Critical control system protection in the 21st century. Computer 46, 74–83.

Amin, S., Schwartz, G.A., Shankar Sastry, S., 2013. Security of interdependent and identical networked control systems. Automatica 49, 186–192. https://doi.org/10.1016/j.automatica.2012.09.007

Baheti, R., Gill, H., 2011. Cyber-physical systems. Impact Control Technol. 12, 161–166.

Barnett, M., Schulte, W., 2003. Runtime verification of. net contracts. J. Syst. Softw. 65, 199–208.

Barras, B., Boutin, S., Cornes, C., Courant, J., Filliatre, J.-C., Gimenez, E., Herbelin, H., Huet, G., Munoz, C., Murthy, C., others, 1997. The Coq proof assistant reference manual: Version 6.1 (PhD Thesis). Inria.

Bécue, A., Fourastier, Y., Praça, I., Savarit, A., Baron, C., Gradussofs, B., Pouille, E., Thomas, C., 2018. CyberFactory#1 — Securing the industry 4.0 with cyber-ranges and digital twins, in: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS). Presented at the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1–4. https://doi.org/10.1109/WFCS.2018.8402377

Befekadu, G.K., Gupta, V., Antsaklis, P.J., 2015. Risk-Sensitive Control Under Markov Modulated Denial-of-Service (DoS) Attack Strategies. IEEE Trans. Autom. Control 60, 3299–3304. https://doi.org/10.1109/TAC.2015.2416926

Bellovin, S.M., 1989. Security problems in the TCP/IP protocol suite. ACM SIGCOMM Comput. Commun. Rev. 19, 32–48.

Blum, M., Wasserman, H., 1994. Software reliability via run-time result-checking, in: Journal of the ACM. Citeseer.

Börger, E., Stärk, R., 2012. Abstract state machines: a method for high-level system design and analysis. Springer Science & Business Media.

Paxson, V., 1998. Bro. A system for detecting network intruders in real-time, in: Proc. 7th USENIX Security Symposium.

Byres, E., Lowe, J., 2004. The myths and facts behind cyber security risks for industrial control systems, in: Proceedings of the VDE Kongress. Citeseer, pp. 213–218.

Cárdenas, A.A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., Sastry, S., 2011. Attacks Against Process Control Systems: Risk Assessment, Detection, and Response, in: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS ’11. ACM, New York, NY, USA, pp. 355–366. https://doi.org/10.1145/1966913.1966959

Chen, T., Abu-Nimeh, S., 2011. Lessons from stuxnet. Computer 44, 91–93.

Chupilko, M., Kamkin, A., 2013. Runtime verification based on executable models: On-the-fly matching of timed traces. ArXiv Prepr. ArXiv13031010.

Damjanovic-Behrendt, V., 2018. A Digital Twin Architecture for Security, Privacy and Safety. ERCIM NEWS 25–26.

Davidson, D., Wu, H., Jellinek, R., Singh, V., Ristenpart, T., 2016. Controlling UAVs with Sensor Input Spoofing Attacks., in: WOOT.

Delaware, B., Pit-Claudel, C., Gross, J., Chlipala, A., 2015. Fiat: Deductive synthesis of abstract data types in a proof assistant, in: ACM SIGPLAN Notices. ACM, pp. 689–700.

Dignan, L., 2017. GE aims to replicate Digital Twin success with security-focused Digital Ghost. ZDNet.

Ding, D., Wei, G., Zhang, S., Liu, Y., Alsaadi, F.E., 2017. On scheduling of deception attacks for discrete-time networked systems equipped with attack detectors. Neurocomputing 219, 99–106. https://doi.org/10.1016/j.neucom.2016.09.009

Eckhart, M., Ekelhart, A., 2018a. Towards Security-Aware Virtual Environments for Digital Twins, in: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, CPSS ’18. ACM, New York, NY, USA, pp. 61–72. https://doi.org/10.1145/3198458.3198464

Eckhart, M., Ekelhart, A., 2018b. Securing Cyber-Physical Systems through Digital Twins. ERCIM NEWS 22–23.

Eckhart, M., Ekelhart, A., 2018c A Specification-based State Replication Approach for Digital Twins, in: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. ACM, pp. 36–47.

Ericsson, G.N., 2010. Cyber security and power system communication—essential parts of a smart grid infrastructure. IEEE Trans. Power Deliv. 25, 1501–1507.

Folds, D.J., 2015. Human in the Loop Simulation, in: Modeling and Simulation in the Systems Engineering Life Cycle. Springer, pp. 175–183.

Fournaris, A.P., Lampropoulos, K., Koufopavlou, O., 2018. Trusted hardware sensors for anomaly detection in critical infrastructure systems, in: Modern Circuits and Systems Technologies (MOCAST), 2018 7th International Conference On. IEEE, pp. 1–4.

Fournaris, A.P., Lampropoulos, K., Koufopavlou, O., 2017a. Hardware Security for Critical Infrastructures-The CIPSEC Project Approach, in: 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). IEEE, pp. 356–361.

Fournaris, A.P., Pocero Fraile, L., Koufopavlou, O., 2017b. Exploiting hardware vulnerabilities to attack embedded system devices: a survey of potent microarchitectural attacks. Electronics 6, 52.

Fournaris, A.P., Sklavos, N., 2014. Secure embedded system hardware design–a flexible security and trust enhanced approach. Comput. Electr. Eng. 40, 121–133.

Fovino, I.N., Carcano, A., Masera, M., Trombetta, A., 2009. An experimental investigation of malware attacks on SCADA systems. Int. J. Crit. Infrastruct. Prot. 2, 139–145.

Francia III, G., Thornton, D., Brookshire, T., 2012. Cyberattacks on SCADA systems, in: Proc. 16th Colloquium Inf. Syst. Security Educ. pp. 9–14.

Gandhi, R., Sharma, A., Mahoney, W., Sousan, W., Zhu, Q., Laplante, P., 2011. Dimensions of Cyber-Attacks: Cultural, Social, Economic, and Political. IEEE Technol. Soc. Mag. 30, 28–38. https://doi.org/10.1109/MTS.2011.940293

Gao, S., Kong, S., Clarke, E.M., 2013. dReal: An SMT solver for nonlinear theories over the reals, in: International Conference on Automated Deduction. Springer, pp. 208–214.

Gollmann, D., 2012. Security for cyber-physical systems, in: International Doctoral Workshop on Mathematical and Engineering Methods in Computer Science. Springer, pp. 12–14.

Halfond, W.G., Viegas, J., Orso, A., others, 2006. A classification of SQL-injection attacks and countermeasures, in: Proceedings of the IEEE International Symposium on Secure Software Engineering. IEEE, pp. 13–15.

Hoare, C.A.R., 1978. Proof of correctness of data representations, in: Programming Methodology. Springer, pp. 269–281.

Hodge, V., Austin, J., 2004. A survey of outlier detection methodologies. Artif. Intell. Rev. 22, 85–126.

Humayed, A., Lin, J., Li, F., Luo, B., 2017. Cyber-Physical Systems Security—A Survey. IEEE Internet Things J. 4, 1802–1831. https://doi.org/10.1109/JIOT.2017.2703172

Hwang, H., Jung, G., Sohn, K., Park, S., 2008. A study on MITM (Man in the Middle) vulnerability in wireless network using 802.1 X and EAP, in: Information Science and Security, 2008. ICISS. International Conference On. IEEE, pp. 164–170.

Igure, V.M., Laughter, S.A., Williams, R.D., 2006. Security issues in SCADA networks. Comput. Secur. 25, 498–506.

Kane, A., 2015. Runtime monitoring for safety-critical embedded systems.

Katsigiannis, K., Serpanos, D., 2018. MTF-Storm: a high performance fuzzer for Modbus/TCP, in: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, pp. 926–931.

Khan, M.T., Serpanos, D., Shrobe, H., 2018. ARMET: Behavior-based secure and resilient industrial control systems. Proc. IEEE 106, 129–143.

Khan, M.T., Serpanos, D., Shrobe, H., 2016. A rigorous and efficient run-time security monitor for real-time critical embedded system applications, in: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum On. IEEE, pp. 100–105.

Khorshed, M.T., Sharma, N.A., Kumar, K., Prasad, M., Ali, A.B.M.S., Xiang, Y., 2015. Integrating Internet-of-Things with the power of Cloud Computing and the intelligence of Big Data analytics — A three layered approach, in: 2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE). Presented at the 2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), pp. 1–8. https://doi.org/10.1109/APWCCSE.2015.7476124

Kim, K.-D., Kumar, P.R., 2012. Cyber–physical systems: A perspective at the centennial. Proc. IEEE 100, 1287–1308.

Kim, T.T., Poor, H.V., 2011. Strategic Protection Against Data Injection Attacks on Power Grids. IEEE Trans. Smart Grid 2, 326–333. https://doi.org/10.1109/TSG.2011.2119336

Koopman, P., Wagner, M., 2016. Challenges in autonomous vehicle testing and validation. SAE Int. J. Transp. Saf. 4, 15–24.

Koulamas, C., Kalogeras, A., 2018. Cyber-Physical Systems and Digital Twins in the Industrial IoT. IEEE Comput.

Koulamas, C., Lazarescu, M.T., 2018. Real-Time Embedded Systems: Present and Future. MDPI Electron. 7.

Kriebel, F., Rehman, S., Hanif, M.A., Khalid, F., Shafique, M., 2018. Robustness for Smart Cyber Physical Systems and Internet-of-Things: From Adaptive Robustness Methods to Reliability and Security for Machine Learning, in: 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). Presented at the 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 581–586. https://doi.org/10.1109/ISVLSI.2018.00111

Lakhina, A., Crovella, M., Diot, C., 2005. Mining anomalies using traffic feature distributions, in: ACM SIGCOMM Computer Communication Review. ACM, pp. 217–228.

Langner, R., 2011. Stuxnet: Dissecting a cyberwarfare weapon. IEEE Secur. Priv. 9, 49–51.

Lee, P., Clark, A., Bushnell, L., Poovendran, R., 2014. A Passivity Framework for Modeling and Mitigating Wormhole Attacks on Networked Control Systems. IEEE Trans. Autom. Control 59, 3224–3237. https://doi.org/10.1109/TAC.2014.2351871

Lei, H., Chen, B., Butler-Purry, K.L., Singh, C., 2018. Security and Reliability Perspectives in Cyber-Physical Smart Grids, in: 2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia). Presented at the 2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), pp. 42–47. https://doi.org/10.1109/ISGT-Asia.2018.8467794

Leverett, E.P., 2011. Quantitatively assessing and visualising industrial system attack surfaces. Univ. Camb. Darwin Coll. 7.

Lin, S.-W., Crawford, M., Mellor, S., 2017a. The Industrial Internet of Things, Volume G1: Reference Architecture. Industrial Internet Consortium.

Lin, S.-W., Murphy, B., Clauer, E., Loewen, U., Neubert, R., Bachmann, G., Pai, M., Hankel, M., n.d. Architecture Alignment and Interoperability - An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper (No. IIC:WHT : IN3 : V1.0:PB : 2017120 5).

Mamdouh, M., Elrukhsi, M.A.I., Khattab, A., 2018. Securing the Internet of Things and Wireless Sensor Networks via Machine Learning: A Survey, in: 2018 International Conference on Computer and Applications (ICCA). Presented at the 2018 International Conference on Computer and Applications (ICCA), pp. 215–218. https://doi.org/10.1109/COMAPP.2018.8460440

Martinelli, F., Matteucci, I., 2007. An approach for the specification, verification and synthesis of secure systems. Electron. Notes Theor. Comput. Sci. 168, 29–43.

Matteucci, I., 2007. Automated synthesis of enforcing mechanisms for security properties in a timed setting. Electron. Notes Theor. Comput. Sci. 186, 101–120.

Maurer, T., 2017. What is a digital twin?, Siemens, https://community.plm.automation.siemens.com/t5/Digital-Twin-Knowledge-Base/What-is-a-digital-twin/ta-p/432960

Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. CSUR 46, 55.

Mo, Y., Garone, E., Casavola, A., Sinopoli, B., 2010. False data injection attacks against state estimation in wireless sensor networks, in: 49th IEEE Conference on Decision and Control (CDC). Presented at the 49th IEEE Conference on Decision and Control (CDC), pp. 5967–5972. https://doi.org/10.1109/CDC.2010.5718158

Mo, Y., Kim, T.H.-J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., Sinopoli, B., 2012. Cyber–physical security of a smart grid infrastructure. Proc. IEEE 100, 195–209.

Mouratidis, H., Giorgini, P., Manson, G., 2003. Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems, in: Eder, J., Missikoff, M. (Eds.), Advanced Information Systems Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 63–78.

Neuman, D.C., 2009a. Challenges in Security for Cyber-Physical Systems, in: Proc. DHS Workshop Future Directions Cyber-Phys. Syst. Security.

Pang, Z.H., Liu, G.P., Dong, Z., 2011. Secure Networked Control Systems under Denial of Service Attacks*. IFAC Proc. Vol., 18th IFAC World Congress 44, 8908–8913. https://doi.org/10.3182/20110828-6-IT-1002.02862

Pfleeger, C.P., Pfleeger, S.L., 2006. Security in Computing, Prentice Hall, 4th edition, 2006.

Qin, S.J., 2012. Survey on data-driven industrial process monitoring and diagnosis. Annu. Rev. Control 36, 220–234.

Rajkumar, R., Lee, I., Sha, L., Stankovic, J., 2010. Cyber-physical systems: the next computing revolution, in: Design Automation Conference (DAC), 2010 47th ACM/IEEE. IEEE, pp. 731–736.

Rigatos, G., 2016. Intelligent renewable energy systems: modelling and control. Springer.

Rigatos, G., 2015. Differential Flatness Approaches to Nonlinear Filtering and Control: Applications to Electromechanical Systems. Springer, New York.

Ross, R.S., Katzke, S.W., Johnson, L.A., 2006. Minimum security requirements for federal information and information systems.

Ruiz, J.F., Maña, A., Rudolph, C., 2015. An integrated security and systems engineering process and modelling framework. Comput. J. 58, 2328–2350.

Ryu, D.H., Kim, H., Um, K., 2009. Reducing security vulnerabilities for critical infrastructure. J. Loss Prev. Process Ind. 22, 1020–1024.

Santamarta, R., 2012. Here be backdoors: A journey into the secrets of industrial firmware. Black Hat USA.

Schweichhart, K., n.d. Reference Architectural Model Industrie 4.0 (RAMI 4.0) - An Introduction.

Serpanos, D., 2018. The Cyber-Physical Systems Revolution. Computer 51, 70–73.

Serpanos, D., Wolf, M., 2017. Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies. Springer.

Setola, R., 2011. Cyber Threats to SCADA Systems.

Singh, V.P., Kishor, N., Samuel, P., 2016. Load Frequency Control with Communication Topology Changes in Smart Grid. IEEE Trans. Ind. Inform. 12, 1943–1952. https://doi.org/10.1109/TII.2016.2574242

Soulat, R., 2014. Synthesis of correct-by-design schedulers for hybrid systems (PhD Thesis). École normale supérieure de Cachan-ENS Cachan.

Tao, F., Zhang, H., Liu, A., Nee, A., 2018. Digital Twin in Industry: State-of-the-Art. IEEE Trans. Ind. Inform.

Tauber, M., Schmittner, C., 2018. Enabling Security and Safety Evaluation in Industry 4.0 Use Cases with Digital Twins. ERCIM News.

Turk, R.J., others, 2005. Cyber incidents involving control systems. Citeseer.

Watterson, C., Heffernan, D., 2007. Runtime verification and monitoring of embedded systems. IET Softw. 1, 172–179.

Welch, D., Lathrop, S., 2003. Wireless security threat taxonomy, in: Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society. IEEE, pp. 76–83.

Yang, J., Yessenov, K., Solar-Lezama, A., 2012. A language for automatically enforcing privacy policies, in: ACM SIGPLAN Notices. ACM, pp. 85–96.

Zhang, H., Shu, Y., Cheng, P., Chen, J., 2016. Privacy and performance trade-off in cyber-physical systems. IEEE Netw. 30, 62–66. https://doi.org/10.1109/MNET.2016.7437026

Zhang, M., Duan, Y., Feng, Q., Yin, H., 2015. Towards automatic generation of security-centric descriptions for android apps, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 518–529.