Received November 22, 2020, accepted December 3, 2020, date of publication December 21, 2020,
date of current version January 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3045972

Analytical Review of Cybersecurity for Embedded Systems

ABDULMOHSAN ALOSEEL[1], HONGMEI HE[1], (Senior Member, IEEE), CARL SHAW[2], AND MUHAMMAD ALI KHAN[1]

[1]School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Bedford MK43 0AL, U.K.
[2]Cerberus Security Laboratories Ltd., Bristol BS34 8RB, U.K.

Corresponding author: Abdulmohsan Aloseel (abdulmohsan.aloseel@cranfield.ac.uk)

This work was supported by School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Bedford MK43 0AL, U.K, under Grant EMJ1001E.

ABSTRACT To identify the key factors and create the landscape of cybersecurity for embedded systems (CSES), an analytical review of the existing research on CSES has been conducted. The common properties of embedded systems, such as mobility, small size, low cost, independence, and limited power consumption when compared to traditional computer systems, have caused many challenges in CSES. The conflict between cybersecurity requirements and the computing capabilities of embedded systems makes it critical to implement sophisticated security countermeasures against cyber-attacks in an embedded system with limited resources, without draining those resources. In this study, twelve factors influencing CSES have been identified: (1) the components; (2) the characteristics; (3) the implementation; (4) the technical domain; (5) the security requirements; (6) the security problems; (7) the connectivity protocols; (8) the attack surfaces; (9) the impact of the cyber-attacks; (10) the security challenges of the ESs; (11) the security solutions; and (12) the players (manufacturers, legislators, operators, and users). A Multiple Layers Feedback Framework of Embedded System Cybersecurity (MuLFESC) with nine layers of protection is proposed, with new metrics of risk assessment. This will enable cybersecurity practitioners to conduct an assessment of their systems with regard to twelve identified cybersecurity aspects. In MuLFESC, the feedback from the system-components layer to the system-operations layer could help implement "Security by Design" in the design stage at the bottom layer. The study provides a clear landscape of CSES and, therefore, could help to find better comprehensive solutions for CSES.

INDEX TERMS Characteristics of embedded system, countermeasures, embedded system, cybersecurity of embedded system, MuLFESC, risk assessment.

I. INTRODUCTION

The embedded system (ES) concept, in its simplest form, is manifested when a processing unit is integrated into a larger physical system to steer its functions. For decades, ESs have gone through different stages of development until they have reached what they are today. The capabilities of ESs evolved in conjunction with several key technologies. The most common technologies are integrated circuits (ICs), such as Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs). The difference between FPGAs and ASICs lies in the fact that FPGAs are reconfigurable, whereas ASICs must be pre-configured for the purpose for which they are manufactured. During the course of embedded system development, the inclusion of Programmable Logic Controllers (PLCs), microcontrollers, and microprocessors played a vital role in the advancing of the capabilities of ESs, allowing them to be employed in a wide variety of applications. With the development of the Internet of Things (IoT), ESs have shown great potential in IoT network connected systems, and their capabilities have been increasingly improved, moving closer to those of traditional IT systems.

Technology is experiencing significant development because of the expansion of Cyber-Physical Systems (CPS), or IoT-enabled CPS. In all application domains of IoT-enabled CPS, such as Smart Cities, Supervisory Control, and Data Acquisition systems (SCADA), healthcare, transportation, communication, military, unmanned vehicles, smartphone, smart grids, gas distribution systems, avionics, and wearable devices, ESs have played significant roles in sensing, computing, and controlling.

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

The IoT and its cyber-physical environment bring great benefits by connecting people, processes, and data. However, IoT-enabled systems can be threatened by a wide variety of cyber-attacks from criminals, terrorists, and hacktivists [1]. Connecting all devices to the Internet and using off-the-shelf solutions is causing the vulnerabilities of CPS to grow [2]. If we look at what damage cyber-attacks can cause to traditional computer systems despite their computing capabilities, we will recognize the scale of the challenge that faces ESs, with their limited capabilities, when dealing with these cyber-attacks. The high profile WannaCry ransomware attacks in May 2017 showed how victims could be prevented from using their computers or accessing their data. The UK, Spain, Russia, Ukraine, and Taiwan were among the affected countries, with vital data, including confidential medical records, being held to ransom [3]. In another example, a malicious actor infiltrated a German steel facility in 2014. The adversary used a spear-phishing email to gain access to the corporate network and then moved into the plant network, resulting in massive physical damage [1], [4], [5]. Cyberattacks clearly have the potential to disrupt or damage physical systems in various application domains mentioned above [1], as the previous developers did not take cybersecurity into account in the design of ESs.

With connectivity to the Internet, ESs are more vulnerable to cyberattacks than ever before and with their limited resources, the problem is exacerbated. In addition, the many influencing factors and involved parties that should be taken into account makes it difficult to determine where deficiencies lie in security measures. Therefore, the study of CSES needs to consider the application context, and advanced and comprehensive ESs security solutions are necessary because of their crucial roles in a diversity of domains.

This survey aims to identify the security challenges and gaps in CSES by determining the influencing factors and related parties, thereby assessing the current status of countermeasures and security solutions against cyber-attacks. To appraise the factors that could affect cybersecurity, we need to understand the structure of the embedded system, its hardware and software components, security objectives, and the vulnerabilities that an attacker can exploit, as well the role of the related parties, including manufacturers, operators, users, and legislators. In this way we can draw the overall landscape of the CSES to help find better solutions.

The remaining part of this paper is divided into five main sections. Section II addresses the concept of ESs, their characteristics, and related terminologies, as well as the problems that are a result of the limitations imposed by those characteristics. In Section III, the security challenges facing ESs as a result of security requirements and, in contrast, their capabilities are discussed. In Section IV, cybersecurity is addressed in relation to security objectives, countermeasures, and risk management. In Section V, security risk metrics, involved parties in CSES and the factors in the cybersecurity industry and the Multiple Layers Feedback Framework of Embedded System Cybersecurity (MuLFESC) are presented. Finally, Section VI concludes the findings of the review.

II. EMBEDDED SYSTEMS

In the continuous pursuit of humankind to improve quality of life, techniques, and knowledge to meet the aspirations and needs of people, one of the most revolutionary aspects appeared in the field of technology when inventors tried to integrate computing operations into physical systems to enable predefined functionality—so-called ‘‘Embedded Systems.’’ During this evolutionary period, significant advances were made in various fields such as industry, health, aviation and communications. The difference between these systems and traditional computers and servers is that they are, as a subsystem, integrated into a larger physical system to perform a specific, essential function. In contrast, computers and servers are designed for multiple purposes, of which computational operations for data processing are the main purpose.

There are many definitions of ESs based on different perspectives. As Vahid and Givargis [6] stated, it is not easy to provide a precise definition of embedded computing systems, or simply embedded systems, and they stated that ‘‘an embedded system is a computing system built into a larger system, designed for dedicated functions. It consists of a combination of hardware, software, and optionally mechanical parts. Thus, the term refers to any computing systems other than general-purpose PC or mainframe computers.’’ [7]. It is noticeable that the main criterion in calling a system an embedded system is the embedding of a processing unit or the integration of computational functionality within a larger physical system to steer the functions of that CPS. Thus, naming an embedded system does not depend on a specific type of logic circuit, CPU, or architecture.

A. ARCHITECTURE OF AN EMBEDDED SYSTEM

Understanding the construction of the embedded system in terms of entry points and the attack surface leads us to predict which aspects should be protected from the risks of cyberattacks. The field of ESs is vast. Due to the widespread application of ESs in different technical domains, the design of the architecture of ESs in different applications is not limited to a particular form. Manufacturers seek to configure the design to fit the purpose it was designed for. An embedded system is typically comprised of CPU, RAM, ROM, and input/output ports [8]. Also, the embedded system CPU can be constructed with instruction cache and data cache or without the I/D caches to keep the CPU architecture simpler and less expensive. To support information exchange or communication, the bus system of an embedded system includes the system bus and the local bus. Figure 1 depicts the typical architecture of ESs. The CPU is the heart of an embedded system, but other components must be added, such as memory and peripheral interfaces, in order to construct the embedded system.

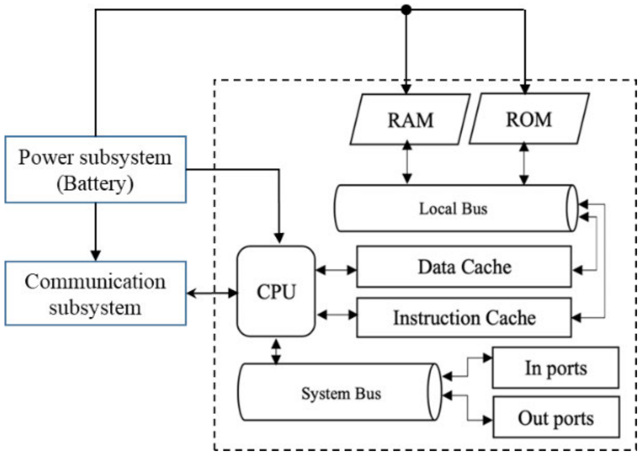

FIGURE 1. Typical architecture of embedded system.

Designs range from small electronic circuits, through microcontrollers with a small number of transistors and the capacity of 8 bits, to multiple core 64-bit microprocessors with speeds over 1 GHz. Various application-specific CPU implementations and architectures are also used, such as FPGA soft or hard cores, digital signal processors (DSPs), or even recent cores optimized for machine learning. This leads us to an important term, System on Chip (SoC): an integrated circuit (IC, also known as a ''chip'') that integrates all components of a computer, in addition to the digital and analogue interfaces. A SoC can be built based on a microcontroller or microprocessor, depending on the purpose of the application. An embedded system can exchange information with external devices through I/O ports. To exchange information remotely, an ES usually is connected to a communication subsystem. The power subsystem provides the power to the components of the ES, with many being battery-powered.

There are various types of embedded systems currently on the market. For example, **Raspberry Pi** is a series of small, inexpensive, single-board computers (SBCs) developed in the UK by the Raspberry Pi Foundation; **Beaglebone** is a low-power, open-source SBC produced by Texas Instruments in association with Digi-Key and Newark element14; and **Jetson Nano** is a small, powerful computer for embedded applications and AI IoT that delivers the power of modern AI in a module. These SBCs have been used widely in education, experimentation, and innovation projects. Süzen *et al.* [9] provided a benchmark analysis study addressing this category of systems.

B. THE ROLE OF ESs

When discussing embedded systems, it is necessary to know the relationship and difference between embedded systems and some terminologies, such as CPS and IoT. In 2006, the term ''CPS'' was coined by Helen Gill from the National Science Foundation (NSF). According to [10], ''The term of cyber-physical systems refers to the tight conjoining of and coordination between computational and physical resources''. One of the essential characteristics that shape the cyber-physical system concept is the ability to interact with the physical world via actuators or sensors. Whereas a cyber-physical system interacts with the external physical world, the responsibility of computational operations lies on the embedded system to steer the physical parts to perform its predefined functions. An embedded system is a co-design of hardware and software. The architecture of the hardware system is shown in Figure 1. The software system of an embedded system consists of the Operating System (OS) and applications [1], [11]. One of the differences between embedded systems and conventional computers is that they are designed to perform specific functions and they are integrated into a larger physical system. Sensors are used to sense the external environment, and actuators are used to steer the larger physical system. From this point of view, the embedded system is, as a computing unit, added to a physical system to shape the concept of the cyber-physical system [11], [2], [12], as shown in Figure 2.

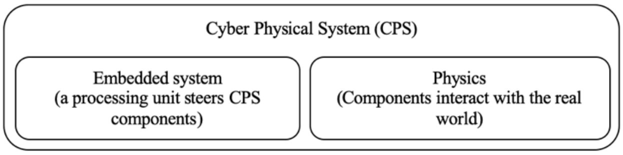
Cyber Physical System (CPS)
Embedded system
(a processing unit steers CPS components)
Physics
(Components interact with the real world)

FIGURE 2. Relationship between embedded systems and cyber-physical systems [11], [2], [12].

Therefore, CPS is the result of the combination of information processing and the physical environment by using computing processing. Similarly, in [5], [13], ''Cyber-physical systems are integrations of computation and physical processes.'' [12] The European Commission [14] also defined the concept of cyber-physical systems as ''the next generation of embedded ICT systems that are interconnected and collaborated through the Internet of things and provide citizens and business with a wide range of innovative applications and services.'' Another definition of CPS is given in [15]: ''A system is comprised of a set of interacting physical and digital components, which may be centralized or distributed, and provide a combination of sensing, control, computation and networking functions, to influence outcomes in the real world through physical processes.''

Industrial Automation and Control Systems (IACS) and Industrial Control Systems (ICS) are other types of CPS [15]. They are further associated with two concepts, Industry 4.0 and Industrial Internet of Things (IIoT), between which there exists an overlap. ''Industry 4.0'' was initially coined by the German government as part of its ''High-Tech Strategy 2020'' in 2010 and is all about connected value chains—connecting and automatically integrating things and processes to form cyber-physical systems [16]. Within the modular structured smart factories of Industry 4.0, CPS monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. With IoT technologies, CPSs communicate and cooperate with each other and with humans in real time. Via the Internet of Services (IoS), both intra- and inter-organizational services are offered and utilized by participants of the value chain [15]. IIoT, first mentioned by General Electric, is a subset of IoT. It leverages the power of smart machines and real-time analytics to take advantage of the data in industries such as manufacturing, transportation, energy and health care, thereby enhancing the productivity and reliability of communication and control in mission-critical applications for transformational business outcomes [15], [17].

In addition, there are two other terms: Distributed Control System (DCS) and Supervisory Control and Data Acquisition (SCADA). A DCS is a computerized control system for a process or plant, usually with many control loops, in which autonomous controllers are distributed throughout the system but there is no central operator supervisory control. SCADA is a system comprised of software and hardware to control and monitor a process or application. It allows an operator in a local center to monitor widely distributed processes (e.g., an oil or gas field, pipeline system, or hydroelectric generators), make set-point changes on distant process controllers (e.g., opening or closing valves or switching), observe alarms, and gather measurement information [4], [15], [18]. Figure 3 depicts the abstract relations between all the concepts mentioned above.

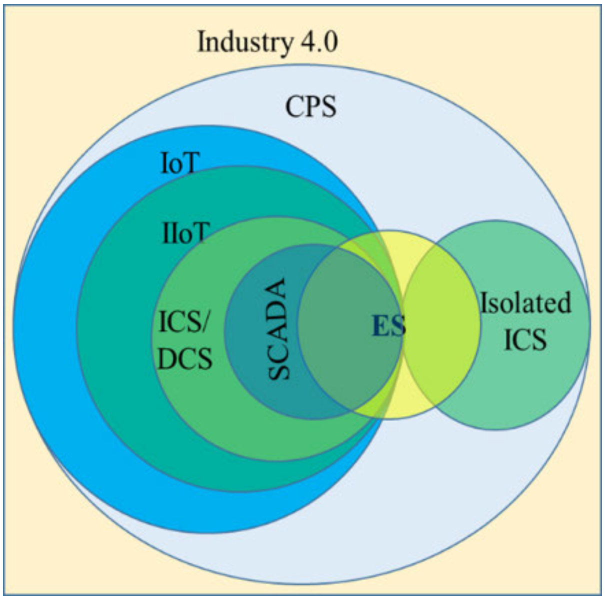

FIGURE 3. **The relations between all relevant concepts.**

From Figure 3, it can be seen that embedded systems (ES) are the core component in all of these concepts. Therefore, securing ESs is very important in all application domains.

C. CHARACTERISTICS AND LIMITATIONS OF ES

Embedded systems have been applied in broad fields. In addition to their use in the daily life of individuals, such as cell phones, tablets, and wearable products, ESs have been used in various application domains. For example, in smart home applications they are used to implement surveillance cameras, remote control, cooling systems and temperature control or thermostat systems. The applications of ESs can be extended for governments, organizations, companies, institutions, national infrastructures, in transportation, or to implement the future trend towards smart cities. In terms of their application conditions, embedded systems usually have such characteristics as low power consumption, small size, specific functionality, remote accessibility, unmanned operation, real-time performance, and low cost. These characteristics of ESs demonstrate their superiority over conventional computers in pervasive and ubiquitous computing. However, the connectivity of ESs to the Internet exposes them to the same cyber threats as conventional computers. ESs are characterized as remotely unmanned operation devices, and the nature of ESs, and the fact that they operate without human intervention, increases the chances of an attacker exploiting vulnerabilities to penetrate these systems. Sometimes, embedded systems are required to operate in harsh environmental conditions or under autonomous control where they are far from human supervision, increasing the potential for unauthorized physical access to these systems. This is a fundamental security problem for most IoT devices. Hence, if the attacker gains a fully unauthorized physical access to the system, the confidentiality, integrity, and availability of the system could be breached. As a result, a new challenge has emerged: how to ensure that the security goals of the system are maintained under these circumstances. The resource limitations of ESs poses tight constraints on both communication and computing capacity [19]. Moreover, the resource constraint problem [13] has given rise to many challenges in creating advanced security solutions for ESs and makes it difficult to meet their cybersecurity requirements. As stated by Meshram and Sasankar [20], ‘‘the limited processing power implies that an embedded system typically cannot run applications for defending against attacks as in conventional computer systems (e.g., virus scanner),’’ and the limited energy also prevents the implementation of advanced security measures. The limited computing resources of ES cannot support complex security schemes [21]. Several studies have addressed characteristics of ESs such as low power consumption and limited computing power in terms of CPU and memory data processing, not only with regard to the system performance requirements but also the problems and weaknesses of securing ESs [22], [23]. The study in [24] addressed the relationship between the characteristics and the problems of an embedded system in implementing IoT devices. Hence, the capabilities of ESs face the challenge of meeting the requirements of advanced security solutions. Table 1 summarizes some of the limitations of ESs due to their characteristics, and Table 2 presents some of the security problems resulting from these limitations.

TABLE 1. Limitations due to the characteristics of ESs.

References	Study Purpose	Processing Gap	Battery Gap	Memory Gap	Storage	Cost	Connectivity & Flexibility	Real Time	Implement Advanced Countermeasures
[25]	ES design principles	✓	✓	✓		✓	✓		✓
[19]	ESs' security from new dimension	✓	✓	✓	✓				✓
[26]	Security-aware and real-time ES		✓					✓	✓
[27]	Hardware enhancement of ES	✓	✓			✓			✓
[28]	Cybersecurity for cyber-physical system		✓	✓	✓		✓		
[21]	IoT and security measurements	✓		✓			✓		✓
[29]	Taxonomy of detection approaches	✓	✓				✓		
[30]	Data privacy and policy problems survey								✓
[13]	Security requirements and challenges in IoT								✓
[31]	Challenges, threats and solutions of IoT	✓		✓					✓
[32]	A roadmap for security challenges in the IoT	✓	✓						✓
[7]	A taxonomy of threats, vulnerabilities and attacks.								✓
[33]	A survey of cyber-physical systems security	✓	✓	✓			✓		
[34]	Security framework for IoT	✓	✓		✓				✓

TABLE 2. Security problems due to the limitations of the ES.

Limitation of ES	Problems
Limited processing capability	ES cannot run advanced solutions that are used for defenses against attacks as traditional computer or server systems do
Limited available power supply	ES can dedicate only limited power resources to provide system security
Operating in uncontrolled or harsh environment	ES is vulnerable to physical attack
Remote and unmanned operation	This feature imposes difficulties in physical access, monitoring, downloading updates and patches of vulnerabilities
Network connectivity via wireless or wired	This feature enables the attacker to perform remote attacks. The diversity and uncertainty of cyber-attacks make it difficult to predict how and when they occur, so many vulnerabilities can be exploited

III. SECURITY RISKS OF EMBEDDED SYSTEM

A. SECURITY PROBLEMS

Embedded systems are vulnerable to a wide range of attacks that might breach their security. For instance, an exhaustion attack could drain the power resource by increasing computational tasks or the use of peripherals or sensors. Also, ES is vulnerable to physical attacks: if attackers have physical access to the system, they might conduct a physical intrusion, tamper with the integrity of the system, and/or perform snoop attacks on the system bus, as well as possibly causing sensor or peripheral damage.

No matter what kind of system is under consideration, embedded or conventional, they all have generic security objectives: confidentiality, integrity, and availability. For example, malware (e.g., buffer overflow attacks [24]) can attack networked embedded systems. The stored data or cryptographic keys of an embedded system or electronic currency on smart cards are vulnerable to unauthorized access, and they must be protected to ensure the security of ESs. Also, the authenticity of an embedded system is vulnerable to numerous attacks, such as forged, malicious, or incorrect data or information produced by the system's sensors, an unauthorized user, or unauthorized reprogramming.

The processor's capabilities, which are the heart of the embedded system, are at the top of the list, and they are often unable to implement advanced security techniques such as data encryption. Furthermore, the CPUs themselves have insufficient hardware protection against logical and physical attacks. A more robust CPU could mitigate a lot of attacks, but these are more expensive and normally limited to use on smart cards or as dedicated secure elements in SoCs. Assuming that the performance of the processor has been improved in accordance with the requirements of advanced encryption, this creates a new problem: the need for a significant amount of energy, which may not be available in the case of portable systems. However, if these two problems are resolved, we will face a new problem, which is that of cost. A small increase in the cost of production, even if only a few cents, would be very expensive and would affect competitiveness if millions of units were manufactured [35].

Cybersecurity specialists often try to know the attacker's capabilities to prevent attacks. Indeed, the attacker's abilities depend on what is made available and unprotected in terms of entry points in the attack surface. Hardware components, such as Wi-Fi, Bluetooth, USB or other input/output interfaces, and software systems, such as operating systems or applications, increase the capabilities and flexibility of ESs, but may provide a greater attack surface for hackers; thus, the system becomes more vulnerable for cyberattacks. In other words, if the capabilities of ESs increase in terms of points of connection and input units, then attack surfaces increase, thereby increasing the probability that the system is hacked. Compounding this problem is the easy availability of advanced, low-cost physical attack tools such as

ChipWhisperer and ChipShouter that can be used to generate side-channel attacks (SCAs) or glitch attacks [36].

On the other hand, although imposing restrictions on entry points to the embedded system may contribute to reducing the attacker's chances, this conflicts with the importance of system flexibility. The cybersecurity problems of embedded systems are endless, and they differ according to the assessment perspective and technical domain of application. For example, several security problems are related to Internet connectivity; several studies have addressed this problem [2], [37], [38]. Furthermore, the problem of data privacy and policies was present in a study on smart cities [30], and the problem of resource constraint and the need to design lightweight encryption [39] and energy-efficient countermeasure strategies were discussed in [13], [19], [20]. The lack of a unified theoretical framework in the design of CPS is an problem that deserves attention [40]. Also among the problems is the operation of ESs in an unattended environment, which creates several security challenges and is easily accessible to the attacker, as well as the problem of the use of off-the-shelf solutions [19]. The initial design stages and their importance, as well as the neglect of the security requirements in the initial stages of design, are discussed in [38], which reinforces the urgent need to adopt a 'security by design' concept, not only for this reason but also because the embedded systems are designed for fixed purposes; thus, a successful attack on one sample of the embedded system's applications could facilitate the repetition of the attack on other embedded systems of the same type without additional cost or effort.

B. CLASSIC ATTACKS AND IMPACT ON ESs

Abomhara and Køien [24] addressed security risks in terms of four aspects: vulnerabilities, exposure, threat, and attacks. Vulnerabilities refer to weaknesses in a system, design deficiencies, or weaknesses in policies or procedures that might allow the attackers to have unauthorized access to data, execute illegitimate commands or conduct attacks. Furthermore, vulnerabilities might be found in different software layers: applications, operating systems, or communication protocol stacks [24], [41], [42]. In the context of cybersecurity, a vulnerability is a weakness that can be exploited by a cyberattack to gain unauthorized access to or perform unauthorized actions on a computer system. Vulnerabilities can allow attackers to run code, access a system's memory, install malware, and steal, destroy, or modify sensitive data.

Exposure risks refer to problems or mistakes in a system configuration that might be exploited by an intruder. Threats refer to the activities that take advantage of security weaknesses in a system to conduct a harmful impact [24], [43]. A cyber threat is a potential malicious act that might exploit a vulnerability to breach security and, therefore, cause possible harm. This threat can be an intentional action, accidental event, or an abnormal circumstance. Cyber threats include unstructured threats, which use existing hacking tools, and structured threats, e.g., Advanced Persistent Threats (APTs), conducted by an expert attacker [24], [28], [44], [45]. APT is a prolonged and targeted cyberattack in which an intruder gains access to a network and tries to remain undetected. An APT attack usually seeks to monitor network activity and breach the confidentiality of data rather than to cause direct damage to the network or organization. APT attacks are often preceded by planning, require tremendous experience, and are intended to spy for a longer term [46].

Cyber-attacks refer to the actions taken by an attacker to cause damage or harm to the system or disrupt normal operations by using different techniques or tools. There are many types of attacks: (a) physical attacks, (b) reconnaissance attacks, (c) denial-of-service (DoS), (d) access attacks, (e) attacks on privacy, (f) cyber-crimes, and (g) destructive [24], [47], [48], [49]–[52]. Several challenges were presented by [2], such as safety, security, and confidentiality, as well as reliability, reparability, and availability; this highlights the importance of cybersecurity defense countermeasures implementation in embedded systems. For network-connected systems, considering the four layers of Transmission Control Protocol/Internet Protocol (TCP/IP) in early stages of designing the embedded system will contribute to hardening the ES. The layers of TCP/IP are Application layer, Transport layer, Network layer, and Link and Physical layer [28]; these layers play an important role in terms of the security of the embedded system, as the weakness of the measures taken in these layers will create many vulnerabilities that the attacker can exploit, and in return, the implementing of best practices in these layers will enhance the stability of the system. The biggest challenge facing an embedded system is when it is connected to the public Internet, as it faces unpredictable cyber threats. Although embedded systems usually sit at the bottom layer—the physical layer of IoT systems—it is crucial to take into account the four layers of the TCP/IP model to consider cyber-threats from upper layers and the direct threat at the physical layer in the design stage of embedded systems. Ali *et al.* [4] extensively discuss the security threats and vulnerabilities according to the relationship between cyber-physical systems and the TCP/IP model. Fitz *et al.* [53] discuss the effect of network topography on the stability of cyber-physical systems' connectivity. Networks can be categorized into six types: Star, Bus, Linear, Ring, Tree, and Mesh. Mesh design, which could be partially or fully connected, is the highest cost among these topologies to maintain the connectivity of CPS and its sensors. The International Organization for Standardization (ISO) defines a 7-layer reference network model. The physical layer accounts for an important proportion of energy consumption, in addition to the existence of a lack of unified specific standards for designers and developers of CPS to mitigate cyber risks. Figure 4 summarizes classic cyberattacks on the TCP/IP layers.

Several valuable documents in the field of research were published by the National Institute of Standards and Technology (NIST) and related to NIST standards [54]–[60]. The impact of cyber-attacks on the functions of a cyber-physical system was presented in [61] by reviewing a case study of a rail transport system. The case study was performed based on Hardware-in-the-Loop (HIL) simulation to avoid

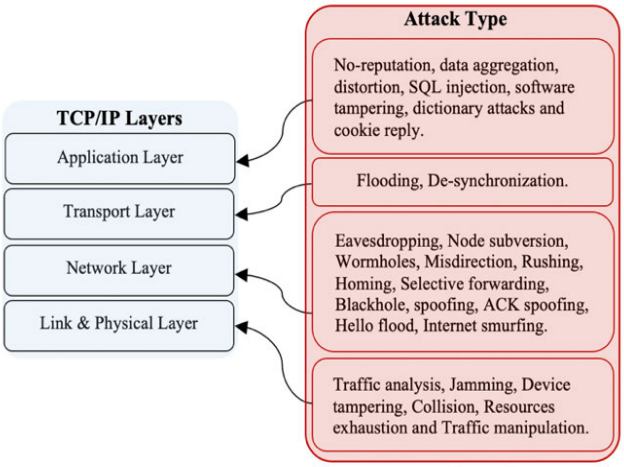
Attack Type
No-reputation, data aggregation, distortion, SQL injection, software tampering, dictionary attacks and cookie reply.
Flooding, De-synchronization.
Eavesdropping, Node subversion, Wormholes, Misdirection, Rushing, Homing, Selective forwarding, Blackhole, spoofing, ACK spoofing, Hello flood, Internet smurfing.
Traffic analysis, Jamming, Device tampering, Collision, Resources exhaustion and Traffic manipulation.
TCP/IP Layers
Application Layer
Transport Layer
Network Layer
Link & Physical Layer

FIGURE 4. TCP/IP layers and attacks [4].

any severe damage or danger for humans in the real world. HIL acts as an interface platform between the physical components (sensors in this case) and an embedded system, acting as the computational part. The effect of a cyber-attack on the efficiency of the functions of CPS was studied. It was shown that digital attacks can affect the effectiveness of the functions of CPS, causing significant delays transiting a signal between a sender and a receiver when the infrastructure is attacked.

Prevention of cyber-attacks is a critical challenge due to the diversity of attacks and the constraints of ESs. One of the most critical stages in dealing with a cyberattack is the detection of the attack itself. Late detection of an attack could allow severe consequences, such as system damage, to occur, whereas early detection allows for a suitable response. However, detection may require significant resources and limit the functionality of the system. An inappropriate response itself could also contribute to an attack and even be used by an attacker. For example, suppose a system disables network connectivity to prevent an attack. In this case, it could be used to direct the re-connection to a fake access point, not to mention the loss of availability during the disconnection time. Some unfinished or unnecessary responses could also conflict with the operation of a device, for example, safety-critical operation, real-time or necessity of immediate functional reaction and synchronization. Hence, an appropriate strategy for detection and response is needed to avoid possible damage but maintain the required functionality of a system, and this is often a compromise.

Each embedded system has its own requirements and, therefore, requires its own security methods. The absence of specific manufacturing standards has exacerbated the problem of creating a unified and comprehensive security solution, [40] although the essence and architecture of embedded systems are, in general, similar. As each system is designed for its own purpose, specific security mechanisms are required to support the functional requirements of the system.

C. CHALLENGES OF IMPLEMENTING CSES

IoT-enabled cyber-physical systems greatly increase the diversity of ES applications. Conversely, the cybersecurity vulnerabilities of ESs open the doors to countless types of cyber-attacks, and this is one of the biggest challenges facing ESs enabled by IoT technology [2], [37], [38]. Another challenge for the cybersecurity of ESs lies in the fact that ESs could work in a non-controlled environment [19], in a stand-alone and independent manner. Industry 4.0 raises the new concept of security by design. Cybersecurity should be considered when designing an embedded system, with specific regard to the security challenges caused by the characteristics of that system.

One of the most challenging aspects of embedded system security design is having to implement the security objectives within the system's capabilities and to do this without focusing on a specific aspect and neglecting to take into account other aspects, or providing non-comprehensive solutions. Habibzadeh *et al.* [30] found that the problem of existing cybersecurity research lies in the focus on a single component in a system, and they suggested that a robust CPS should have the cybersecurity capabilities of all of its components and that the security of a system is typically determined by its weakest link. Hence, Habibzadeh *et al.* considered ensuring the overall security of the system is the weakest link that needs to be addressed to have a robust CPS. The "weakest link" concept has been discussed at length in [62], and we can define it in this context as "A guardian is an entity in the system that the attacker could try to pass to gain access to an asset. The cost of passing a guardian determines the negative utility for the attacker when deciding to pass. The cost is typically dependent on the entities an attacker already has access to, such as keys or passwords" [62].

It is important to clearly understand the terminology of security risks, security threats, cyberattacks, vulnerabilities, and exposure risks [24], [7]. The distinction between these terms contributes to clarifying the vision of the nature of the risks facing ESs and their applications and, thus, facilitates the diagnosis of the problem and the finding of an appropriate security solution. A detailed explanation of these terms is provided in Section B.

The challenges facing ESs begin at the initial design stages and continue up to the final operational phase. During these stages, some of the most obvious challenges hinder the design of highly efficient countermeasures against cyberattacks, such as processing gap, battery gap, flexibility, tamper resistance, assurance gap, and cost [11], [19]. The design process of an ES is influenced predominantly by cost; in terms of the time factor, ES industries always pursue a fast development cycle for market competition. This adversely affects manufacturers in applying high standards for the development of advanced security solutions. Another critical challenge facing ESs is the consumption and measurement of energy that supports the functions of IoT components [63]. The optimization of energy is demanded due to the constraints on embedded system resources. The battery gap or the power consumption optimization constitutes one of the most challenging design factors [11], [19]. Table 3 summarizes the research directions on the security risks domain of ES.

TABLE 3. Summary table of the research directions on security risks aspects of the embedded systems.

Researches Directions		Addressed Aspects	References
Risks Taxonomy	Attacks	Actions taken by attacker (techniques or tools).	[24], [47], [48] [49]- [52]
	Threats	Unstructured threats, Structured threats Threats by Humans or Nature (Unauthorized physical access, Natural Disasters, Earthquake, Hurricanes or flood).	[4], [19] [24],[28] [44], [45] [46]
	Vulnerabilities	Weaknesses in the system design.	[4], [24] [41], [42]
	Exposure	Problems or mistakes in the system configuration.	[24], [43]
Attack elements	Methods of the attacks	Classification of the attack methods.	[64]
	Surface of the attack	Defining the attack surfaces to determine the entry points of the attack.	[65]
	Attacker capabilities and Tools	Attacker's skills, Availability of the attack tools such as ChipWhisperer and ChipShouter.	[36]
	Assets value	Data sensitivity, Privacy	[30], [61]
	Type of the Attacks	Categorizing security risks based on their nature whether threats, exposure, attack, or vulnerabilities.	[24]
	Connectivity	Network topography, Layers of TCP/IP (Application layer, Transport layer, Network layer, and Link and Physical layer)	[4],[28] [53], [2] [36], [37]
Challenges of Implementing CSES	Lightweight cryptographic.	Encryption requirements should consider power consumption.	[13]
	Denial of Service attack	DoS attacks can break the availability of the system.	[13], [24]
	Profiling and tracking Localization	Tracking and localization can be a challenge against privacy.	[13]
	Early detection of attacks	It needs the efficient use of system resources.	[13]
	Secure data transmission	The connectivity causes a large number of routing security problems.	[11], [40]
	Advanced Persistent Threats (APT)	An attack in which an unauthorized user gains access to a system or network and remains there for an extended period without being detected.	[24]
	Energy Challenge	It is important to concern the consumption and measurement of energy in improving the performance of IoT-enabled physical systems.	[63]
	Processing gap	Weak CPU capabilities versus encryption requirements.	[9], [18]
	Battery gap	One of the biggest challenges, affecting the lifespan of the system.	[26]
	Flexibility (different security protocols)	Compatibility problem due to the difference in the protocol used for data exchange due to flexibility in design as per requirement.	[11]
	Cost sensitivity	Cost is one of the biggest acute factors that constrain investors.	[9], [18]
	Physical attacks	Causing unpredicted damage.	[24]
	Reconnaissance attacks.	Scanning network ports, packet sniffers, traffic analysis, and queries about IP address.	[24]
	Access attacks	Due to the unattended nature of the CPS, might be vulnerable to physical access or unauthorized remote access.	[24]
	Attacks on privacy	Data mining, Cyberespionage, Eavesdropping, Tracking, and password-based attacks.	[24]
	Cyber-crimes	Intellectual property or identity theft or fraud.	[24]
	Destructive attacks	Terrorism and revenge attacks.	[24]
	Attacks against (SCADA)	As an embedded concept, it is subject to the same threats.	[24]

IV. CYBERSECURITY OF EMBEDDED SYSTEM

In this section, we will address three different aspects: A. security objectives, B. security countermeasures, and C. risk management and security incident response.

A. SECURITY OBJECTIVES

Cybersecurity, as defined in [66], [24], is a process to protect an object against physical damage, unauthorized access, theft, or loss, by maintaining high confidentiality and integrity of information about the object and making information about that object available whenever needed. To implement effective countermeasures, we need to state the security objectives clearly. The generic security objectives for all systems and services are Confidentiality, Integrity, and Availability, called the CIA triangle. Confidentiality means ensuring the information is not made available or disclosed to unauthorized entities; integrity aims to protect the accuracy and completeness, and the availability implies the information is accessible and usable by an authorized entity when it is demanded. In addition to the security triangle, according to [67], [68] and based on ISO/IEC 27001:2013, [69], embedded systems should also be designed with concerns of Authenticity, Accountability, Non-repudiation, Reliability, Access Control, Dependability, Safety, and Privacy. These security objectives are the cornerstone of the cybersecurity industry in ESs. From the aspect of attackers, the barriers implemented to meet the security objectives of embedded systems are the target elements of attacks; on the other hand, these security objectives are the goals to be achieved, maintained, and guaranteed by cybersecurity practitioners.

Many studies [19], [67], [68] have referred to the security triangle, and these security goals are breached by a number of different types of attacks. For example, confidentiality could be breached by side-channel attacks, authentication attacks, password attacks, packet sniffing, and session hijacking. Integrity could be breached by packet dropping or packet delay attacks and spoofing attacks. The availability could be breached by a buffer overflow or Denial of Service (DoS) attack, which target low memory capabilities and limited computation resources. Most CPS devices are vulnerable to such resource enervation attacks [24].

Industry 4.0 raises the concept of "Security by Design." Cybersecurity is a critical challenge for the success of industry 4.0. Taking security problems and challenges into account in the design stage of a cyber-physical system is the most efficient and effective solution. Also, it is considered the least expensive approach in the long term, compared to the post-processing of cyberattacks, and reduces the need for more modifications or improvements in the final product. Implementing the convergence between the capabilities of embedded systems and cybersecurity objectives is a dilemma: it conflicts with the current approach, where we always strive to design embedded systems with low cost, small size, and low energy consumption, compatible with mobility and dependability concepts, embeddable in larger CPS, and with efficient and sufficient processing capacity. On the other hand, the requirements of cybersecurity are inconsistent with what we seek, as advanced security measures such as sophisticated encryption or intrusion detection systems (IDS) require high computing capabilities which require increasing the transistor count, which in turn increases the cost, size, and power consumption. Even adopting smaller silicon geometries (e.g., 7nm chips) that could help in terms of size and power has a more expensive up-front cost and, therefore, require higher sales volumes to be profitable and cost-effective [70].

B. SECURITY COUNTERMEASURES OF ESs

Security countermeasures for embedded systems have been extensively studied, and most security solutions can be developed in the form of tools, methods, mechanisms, or approaches. Habibzadeh *et al.* [30] suggested that security countermeasures must be done in four dimensions of physical security: firmware-level, device-level, circuit-level, and energy-harvesting- and storage-level. Dibaji *et al.* [71] categorized defense mechanisms against cyberattacks into three types: prevention, resilience, and detection and isolation. The prevention mechanism is designed to counter disclosure attacks, such as Advanced Persistent Threats (APTs), and the two main methods are cryptography and randomization. Resilience refers to the ability of the system to continue to perform its function despite the effects caused by the cyber-attack. Several approaches to implement this mechanism can be applied, for instance, game theory, event-triggered control, mean subsequence reduced algorithms and trust-based approaches. The detection and isolation mechanism consists of five types: observer-based techniques, analytical consistency, watermarking, baiting, and learning-based anomaly detection. Ashibani and Mahmoud [72] highlighted that security measures should take place at three levels to achieve maximum protection: perception, transmission, and application layers.

Attack detection is one of the most important counteractions because it is critical for active countermeasures as it is directly associated with security countermeasures. We must discover the existence of an attack before dealing with it. There are different levels of defense: the first level of defense is to prevent the attack entirely by design, using techniques such as encryption and authentication [73]; the second level is to detect the attack early and deal with it immediately to stop any damage occuring by applying a detection mechanism such as an Intrusion Detection System (IDS) [73]; the third level is to prevent the recurrence of the attack again by taking the required countermeasures after knowing the type of attack and impact of the attack. These levels of defense against cyberattacks require more efficient techniques in different research areas [74], such as vulnerability identification, impact analysis, mitigation, cybersecurity metrics, data and model development, penetration testing, interoperability, and digital forensics. Trawczynski *et al.* [75] provided an approach to detect a DoS-type attack based on the failure of a single node communication interface. Intrusion Detection and Prevention Systems (IDPS) with multi-mode counteractions is also one of the suggested security solutions [76]. The counteraction technique is to block the attacker's IP address via a firewall, based on the number of packets exceeding the threshold limit in one second. In the case of failure, a remote stop of the corresponding service takes place as a third counteraction. In this context, it is important to note that the requirements of the IDS to inspect every packet requires a high resource consumption that is not generally compatible with the capabilities of ESs.

A comprehensive survey of physics-based attack detection techniques was provided by [77], where the researchers highlighted that physical components of cyber-physical systems (e.g., actuators or sensors) need to be monitored to detect the attack based on any abnormality in the performance of these physical components. Also, a discrete-time energy-based attack detection mechanism for a networked cyber-physical system was proposed by [78], where the detecting mechanism is based on the energy balance of the system. Among the existing solutions, Poongothai and Duraiswamy [73] applied a machine learning technique in an IDS to mobile ad hoc networks (MANET), which are not conventionally designed with an IDS, as an example of embedded system applications. Whereas encryption and authentication techniques work as the first line of defense, the IDS can work as a second line of defense. Also, problems such as lack of central points, co-operation, shared radio channel, limited resource availability, and the lack of a clear line of defense and secure communication in MANETs have been addressed in [73].

Gu *et al.* [79] suggested improving security by using a co-processor with the implementation of Mixed Integer Linear Programming (MILP) formulation. Although this approach may contribute to enhancing security aspects, it might conflict with the characteristics of ESs' resources or cost. Wang *et al.* [27] presented a hardware-enhanced protection method to maintain the confidentiality and integrity of data by using an AES stream encryption engine. Also, a combinational logic binding technique against cloning attacks for FPGA-based embedded systems is discussed in [80]. Negi *et al.* [81] discussed the embedded systems in the application field of networks, and the study conducted a test in the transfer of data based on Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The adoption of this protocol as a security protocol has shown promising results in terms of sending and receiving data securely on the level of embedded systems.

A survey of EU research efforts in the security solutions of ESs has been done in [82], where security solutions for ESs were classified to several levels: (i) Node-based security technologies, for example a physical unclonable function (PUF), a physically-defined "digital fingerprint" that serves as a unique identity [30]; and (ii) network-based security technologies, focusing on secure routing and Intrusion Detection System (IDS) for a distributed ES network, implemented with middleware and overlay technologies. Among the studies that dealt with security solutions is also a preemptive security mechanism, which is a thin-layer hypervisor-based memory introspection engine on ESs and was proposed by Lukacs *et al.* [83]. The concept of this mechanism is based on hardware virtualization technology, and this mechanism works on two different levels: privilege level and isolation; and hardware-level virtualization. This technique has been implemented on an x86 CPU, which paved the way for testing this mechanism on ARM Cortex A53 and A57 chips.

Evaluating security requirements to adopt appropriate security countermeasures based on different axes was presented by Elmiligi *et al.* [84]. According to Elmiligi *et al.*, the security requirements can be evaluated from 27 different angles and based on three main axes. These axes are the programmability level axis, integration level axis, and life-cycle phase axis. Figure 5 depicts these axes and their different angles [84].

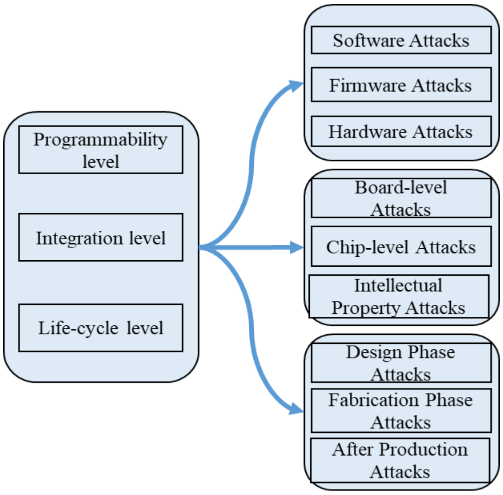

FIGURE 5. **The 27 different angles [84].**

It is also worth mentioning the criticality of a real-time embedded system (RTES). Specific security countermeasures for RTES can be implemented in two stages: (i) identify specific attacks that could threaten the systems; and (ii) implement security-guaranteed services, overcoming the challenges of real-time performance and energy consumption [26]. The study in [85] showed how security measures could be circumvented in different platforms. This demonstrates the importance of highly robust solutions and their availability for a long period.

A non-intrusive runtime monitoring technique for ensuring the safety and security of ESs was presented in [86]. It is based on the principle that finding and implementing solutions on one aspect may depend on another aspect, because the monitoring of non-intrusive runtime through power consumption has been used to enforce safety and security in ESs. This shows that some security solutions are indirect. In other words, enhancing specific security aspects of the system will reflect positively on other aspects.

A multi-metrics approach to ensure and evaluate Security, Privacy, and Dependability (SPD) in ESs is provided in [87], using a smart vehicle as a case study. Also, Mu *et al.* [88] presented a bottom-up approach for the information flow security of a verifiable embedded system based on Gate-Level Information Flow Tracking (GLIFT), at the early stages of designing ESs. The concept of this approach is based on applying restrictions to the information flow to allow only legitimate data to pass through. Liu [89] proposed a security kernel prototype system to support several security verification strategies—for example, multiple levels of security (MLS), Role-Based Access Control (RBAC), and Distribution Transforming Encoder (DTE) [89]. This security kernel prototype system focuses on the security kernel in an embedded system, and it is a very generic security prototype system.

"Security by design" is critically required by Industry 4.0. At the design stage of ESs, it is required to integrate security mechanisms into embedded systems according to Model-Based Development (MBD) [90]. Thayer [91] stated that the adversarial testing in the early stage of designing and developing the embedded system would increase the overall awareness of the threats posed to a system. Also, the high energy efficiency of systems supports the implementation of advanced cryptographic techniques [38]. While a stable and sufficient energy source is essential, thus, it is important to optimize the system for minimizing the energy consumption, so that there is enough energy for preventing attacks.

Hasler and Shah [92] addressed the security implications for the ultra-low energy consumption of SoC FPGA embedded systems.

A comprehensive framework for modeling and assessment for penetration testing of IoT systems, taking attack surface into account, is presented by [93], using a virtual prototype to validate the design of an IoT system. The adopted model uses virtual prototypes (VPs) as a concept, which is a method or technique implemented to validate a development design before any real implementation. The VP is used to develop a framework that aims to support security measures in the initial stages of designing embedded systems. Also, the authors in [94] presented a comprehensive experimental analysis of automotive attack surfaces. The experiment showed severe vulnerabilities that an attacker could exploit, and the results were shared with relevant industry and government stakeholders. Reducing the attack surface by lowering the attacker's access based on a permission-based security model is presented by [95] for Android applications. The suggested approach is designed for detecting permission gaps, using permission-based software.

In general, the adoption of a security solution depends on several factors: the purpose for which it was designed; the capabilities of ES to handle the solution; the nature of the risks that may be exposed; and technical implementation domains. It is also important to train operators with security awareness and relevant knowledge.

C. RISK MANAGEMENT AND SECURITY INCIDENT RESPONSE

Risk management and incident response are important for the cybersecurity of embedded systems, especially networked devices, because security threats cannot be eliminated entirely [73]. A NIST report in 2014 [54] presented a Cybersecurity Risk Management Framework (CRMF) for modern vehicles, where embedded systems are important components. Figure 6 depicts the security lifecycle presented by NIST [54].

Wilbanks [96] proposed a Cyber Risk Management Framework (CRMF) and Cyber Security Risk Indicator (CSRI). CRMF applied three principles: (i) integrating security countermeasures into the systems development lifecycle, (ii) monitoring and maintaining the status of the system, and (iii) interacting with the current situation by making a risk mitigation decision. CSRI measures the efficiency and effectiveness of the system by using quantitative criteria to assess the robustness of the system, [96], [54].

In terms of incident response, the term ‘incident response’ may be related to other terms, such as ‘incident handling’ and ‘incident management’. NIST does not give strict definitions of “incident handling” and “incident response”. CERT®/CC uses “incident handling” to refer to the overall process of incident detection, reporting, analysis, and response, while “incident response” refers to incident containment, recovery, and notification of others [97]. As stated in the Cyber Security Incident Response Guide from the Council for Registered Ethical Security Testers (CREST) [98], “There is no common understanding of what a cybersecurity incident is, with a wide variety of interpretations. With no agreed definition, many organizations adopt different views.” NIST breaks incident response down into four broad phases: (1) Preparation; (2) Detection and Analysis; (3) Containment, Eradication and Recovery; and (4) Post-Event Activity. Phase 2 and Phase 3 are interactive with each other. Dorofee *et al.* [99] classified incident management into five major steps: prepare, protect, detect, respond, and sustain. Incident response is not exclusive to administrative level. Some technical solutions can support early incident response, S. Sultana *et al.* [100] provided a security incident response and prevention system (Kinesis) for Wireless Sensor Networks (WSNs). This system can dynamically respond to anomalous events, based on a suspect's security status, and does not require any central authority to trigger an action.

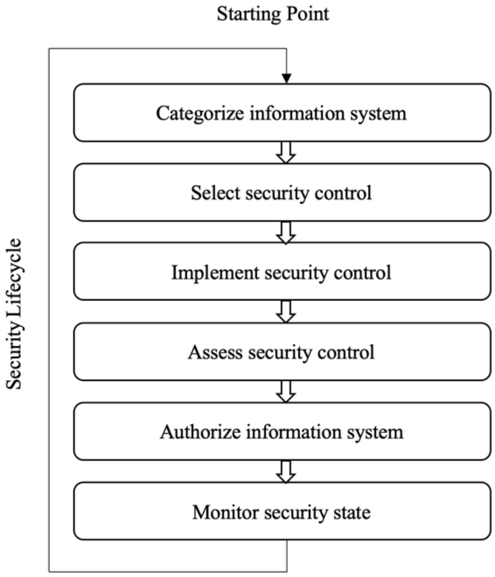

FIGURE 6. NIST security life cycle [54].

According to CREST [98], [101], cybersecurity incidents, particularly serious cybersecurity attacks (e.g., advanced persistent threats (APTs)) have been causing serious damage to organizations, governments, and international bodies. Computer Emergency Response Team (CERT) [99], [102], and NIST [54], [55] have made significant contributions in the subject of cybersecurity incident responses.

As long as the ESs are as essential as traditional computer systems and given the widespread applications of ESs in many domains and at different levels in governments, organizations, and individuals, the supervising parties must have plans to respond to possible incidents following recommended standards such as CRMF or CSRI. If all security measures at all levels fail to prevent and tackle a cyberattack, the responsible parties must at least be able to have a fast response to the incident caused by the attack, thereby reducing the damages and economic loss. They must learn from the incident to ensure that the attack does not recur in the future again. Figure 7 summarizes the research directions in cybersecurity requirements for the ESs.

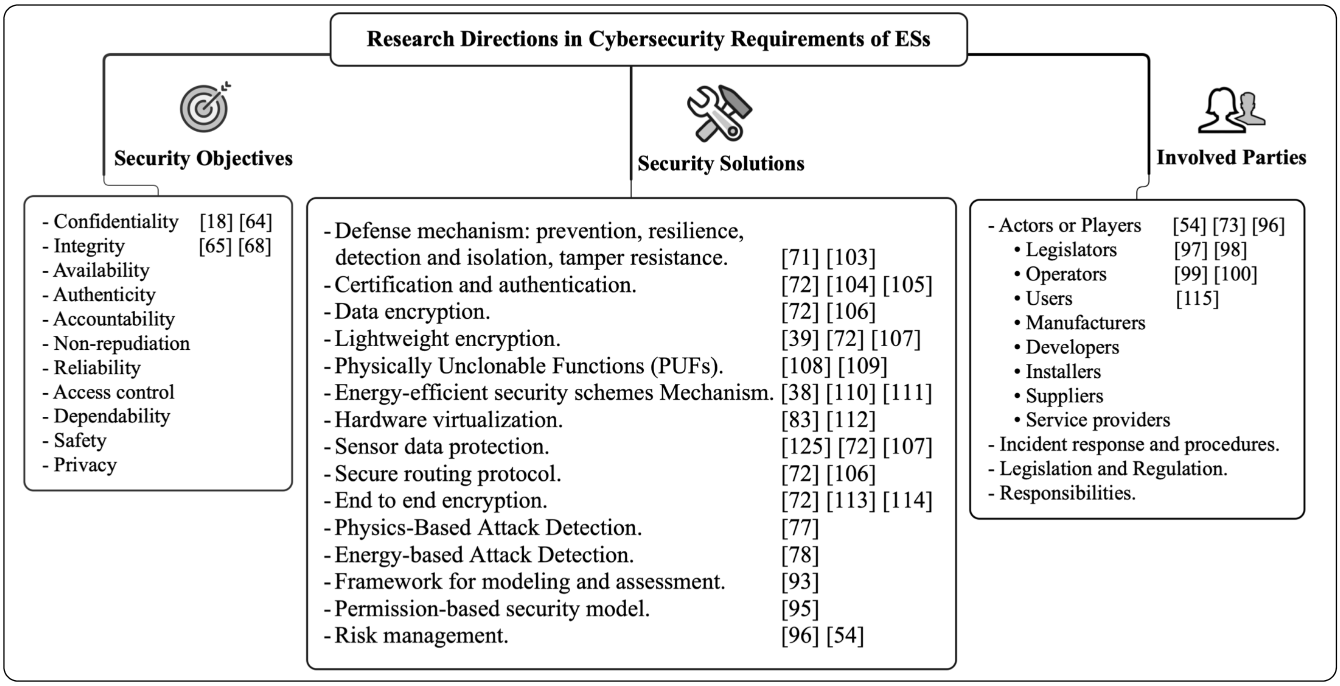
Research Directions in Cybersecurity Requirements of ESs
Security Objectives
Security Solutions
Involved Parties
- Confidentiality [18] [64]
- Integrity [65] [68]
- Availability
- Authenticity
- Accountability
- Non-repudiation
- Reliability
- Access control
- Dependability
- Safety
- Privacy
- Defense mechanism: prevention, resilience, detection and isolation, tamper resistance. [71] [103]
- Certification and authentication. [72] [104] [105]
- Data encryption. [72] [106]
- Lightweight encryption. [39] [72] [107]
- Physically Unclonable Functions (PUFs). [108] [109]
- Energy-efficient security schemes Mechanism. [38] [110] [111]
- Hardware virtualization. [83] [112]
- Sensor data protection. [125] [72] [107]
- Secure routing protocol. [72] [106]
- End to end encryption. [72] [113] [114]
- Physics-Based Attack Detection. [77]
- Energy-based Attack Detection. [78]
- Framework for modeling and assessment. [93]
- Permission-based security model. [95]
- Risk management. [96] [54]
- Actors or Players [54] [73] [96]
• Legislators [97] [98]
• Operators [99] [100]
• Users [115]
• Manufacturers
• Developers
• Installers
• Suppliers
• Service providers
- Incident response and procedures.
- Legislation and Regulation.
- Responsibilities.

FIGURE 7. Research directions in cybersecurity requirements of ESs.

V. ANALYTICAL RESULTS OF CSES

In this section, we will present the security risk metric; we will also identify the influencing factors of CSES to shape the MuLFESC framework, which can be used as an instrument of security assessment for the ESs.

A. SECURITY RISK METRICS

Based on the analysis of the previous studies in this area, multi-security risk metrics have been created and will be presented in this section as part of the contribution. Cybersecurity risk for a system is the probability of exposure or loss resulting from a cyberattack or data breach on the system. The sensitivity of data, its value, and the benefits attackers would gain from hijacking an asset are all large motivations; however, attackers' abilities are contingent on what has been made inadvertently available to them. More entry points mean more possibilities that a system can be exploited. In other words, if the entry points in the attack surfaces are reduced and the unnecessary services are disabled, the chances that a system is attacked are reduced. Many studies have considered security risks from different perspectives, and based on different criteria. According to NIST [56], [115], [116], security metrics are metrics based on IT security performance goals and objectives designed to assist decision-making and improve performance and accountability by collection and analysis of data against potential risk to take an appropriate countermeasure. Within the security risks metrics model, Figure 8, the cyberattacks have been addressed from seven different perspectives, providing the broadest coverage of attack probabilities. Thus, having a comprehensive perception of attack possibilities will lead to having a comprehensive awareness that will be reflected positively in the upcoming Multiple Layers Feedback Framework of Embedded System Cybersecurity, Figure 10. Optimal countermeasures can then be taken in each layer, according to the perspective of evaluation. For example, tackling attacks from the asset value angle will lead to the enactment of the necessary policies in layer 7 to protect the assets according to the sensitivity of the data. Also, addressing attacks based on the attack surface will help in disabling unnecessary entry points in layer 3. Besides, evaluating attacks from the targeted network layer perspective will help in adopting the most secure appropriate protocol, and so on. Therefore, the reflected feedback on layer one fed from other layers (Figure 10) as a continuous process will be enriched by considering these perspectives to achieve best practices and implement a compatible countermeasure, avoiding any conflict with requirements of the other layers.

We define security risks within a metrics of different criteria, based on the security triangle: Confidentiality, Integrity, and Availability (CIA) [19], [67], [68], as the backbone of the security risk metrics (CIA terms are explained in section IV.A). They are what attackers intend to breach ultimately, regardless of the methods or attack surface. Figure 8 depicts the adopted security risk metrics, while (X) refers to a security risk. Based on this risk metrics, a security risk can be addressed from 7 perspectives:

(1) A security risk X should be examined against the proposed Multiple Layers Feedback Framework of Embedded System Cybersecurity (MuLFESC), at the same time considering the twelve influencing factors, as illustrated in Figure 9. Figure 11 in Appendix also presents these factors with more details.

(2) Security risk can be assessed based on attack methods. According to [64], methods of attacks can be classified into three different typical methods: (1) physical method,

(2) logical or software-based method, and (3) Side-Channel attack method.

- A physical attack, whether non-invasive, semi-invasive, or invasive attack means an attacker's ability to access the cyber-physical system directly and this direct physical access is unauthorized and unauthenticated [64]. In this case, it is unpredictable to know what the attacker can do. Also, natural disasters fall within this type of security threat [24], [64]. A fault attack is an example of a physical attack that can be generated to attack an electronic device, and it can be executed by stressing a targeted device beyond its expected operational limits, causing errors [117]. These errors might lead to security failures on the system such as bypassing authentication checks or leaking sensitive information.
- The logical or software-based methods are often used to attack networked ESs through the Internet. The attacks are carried out by exploiting vulnerabilities or exposing errors in software [24], [64], whether in the operating system (OS), applications, protocols used for data transfer, or decryption of the encrypted data.
- In Side-Channel Attacks (SCA), an attacker studies the often unexpected, indirect physical effects of security operations. In this type of attack, the attacker monitors and analyzes system activities produced by its physical components such as electromagnetic emission, power consumption, timing, and cryptanalysis to gain access to protected data 67]. NIST defines a sid-channel attack as follows "An attack enabled by leakage of information from a physical cryptosystem. Characteristics that could be exploited in a side-channel attack include timing, power consumption, and electromagnetic and acoustic emissions" [55]. There are 12 possible attacks based on this method: (1) acoustic attack, (2) brute force attack, (3) cache-based attack, (4) electromagnetic analysis, (5) error message attack, (6) frequency-based attack, (7) glitch attack, (8) power analysis, (9) timing analysis, (10) safe error attack, (11) scan-based attack, and (12) visible light attack [64].

(3) Security risks can be classified into four different types based on their nature: Vulnerabilities, Exposure, Threat, and Attacks [24] (see section III.B).

(4) Attack surface is the sum of all possible security risk exposures. There are three types of attack surfaces [65]:

- The hardware surface can be any possible components in a cyber-physical system or an embedded system, such as sensors for receiving or sending signals, USB ports or Input/output units.
- The software surface, including the logicality of algorithms and protocols, can be in any levels of OS, firmware, protocol handlers or applications.
- Finally, the network-components surface.

As mentioned earlier, a smaller attack surface can help make the system less exploitable, reducing the risk; and a greater attack surface makes the system more vulnerable to attacks, which increases the risk. Cheng *et al.* [118] verify this security problem through using redundant controller architecture to avoid unpredictable mechanical failures, but unfortunately this technique increases the chance of exploiting the attack surface and lowers the sensitivity to respond to ongoing attacks.

(5) The security risk due to network connectivity can be assessed in terms of the four- layers TCP/IP model: Application layer, transport layer, network layer, and link and physical layer [4]. There could be different attacks in each layer; examples of these attacks were discussed in section III.B.

(6) The security challenges in terms of the limitations of the embedded system have been discussed by [25], including processing gap, battery gap, independence, flexibility, installation in an uncontrolled or harsh environment, remoteness and unmanned operation, connectivity to the network, the function's nature of the CPS, and cost. The cybersecurity risk often arises from the limited resources in embedded systems.

(7) Taking into account the attacker's capabilities and the value of the assets is essential for the assessment. A skilled attacker with significant resources poses a much higher risk than a low skilled attacker with few resources. The value of assets, including data, is important. The more sensitive the data, the more security measures are required to ensure its confidentiality, integrity, and availability. While evaluating the efficiency and effectiveness of the applied measures, taking attackers' capabilities and assets' values into account might help improve the adoption of appropriate security. The following figure depicts the suggested security risk metrics.

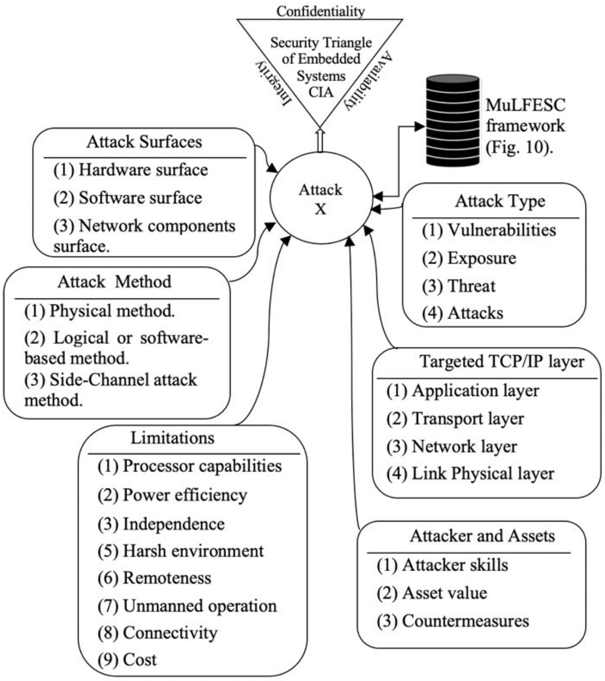

FIGURE 8. Security risk metrics.

B. KEY FACTORS OF THE MuLFESC FRAMEWORK

After presenting the multi-security risk metrics, which can predict the largest possible number of cyberattacks in terms of methods, types, surfaces, TCP/IP layer, and limitations of the ES's resources, this section addresses the key factors that affect the CSES. The determination of the influencing factors could help identify the gaps and weaknesses in the current countermeasures. Figure 9 represents these influencing factors that have been extracted from the existing research trends, while Figure 10 represents the MuLFESC framework layers with its involved parties and system components affected by these factors. The MuLFESC framework has been shaped by taking into account the following factors:

1) THE ARCHITECTURE OF ES IN CPS

Weak computing capabilities of ES could limit the ability to implement advanced security solutions regarding the components and architecture of ES in CPS. Failure to consider security requirements at the design stage of ESs could increase the complication of implementing cybersecurity objectives and requirements in a complicated CPS [13], [8], [119], [84], [88].

2) THE CHARACTERISTICS OF ES

Features of embedded systems and their flexibility in meeting the requirements of modern technologies earned them excellence over traditional computers. However, these characteristics pose many challenges for the cybersecurity of ESs [20], [56].

3) THE IMPLEMENTATIONS OF ES

''Embedding systems'' is a broad concept applied to form the computational part of a wide diversity of applications. Therefore, the diversity of ESs reflects positively on its applications [120], [81], [121], and the constraints of ESs' characteristics and the attack surfaces that can be exploited pose many security challenges. This requires developing market-appropriate security solutions whether in the field of healthcare, communications, military, etc.

4) THE TECHNICAL DOMAINS OF ES

Embedded systems are the core or the basic block of advanced technologies, such as DCS, SCADA, IACS, ICS, Industry 4.0, Industrial, IIoT, IoT, and CPS. The realization of an embedded system with security by design could therefore support the security, stability, and reliability of advanced systems [122], [123]. However, security requirements and solutions need to be considered in the context of different technical domains. The study and test of the embedded system should be done under the context of its real-world applications; thus, the goals of the applications as well as the performance in efficiency, reliability, and stability can be reached.

5) THE SECURITY OBJECTIVES OF ES

Embedded systems have the same objectives of cybersecurity as traditional computer systems, which include: confidentiality, integrity, availability, authenticity, accountability, non-repudiation, reliability, access control, dependability, safety, and privacy. The security requirements depend on the context of application domains. In contrast, the characteristics of ESs raise the challenges in the implementation of embedded system security to achieve these objectives [19], [67], [68].

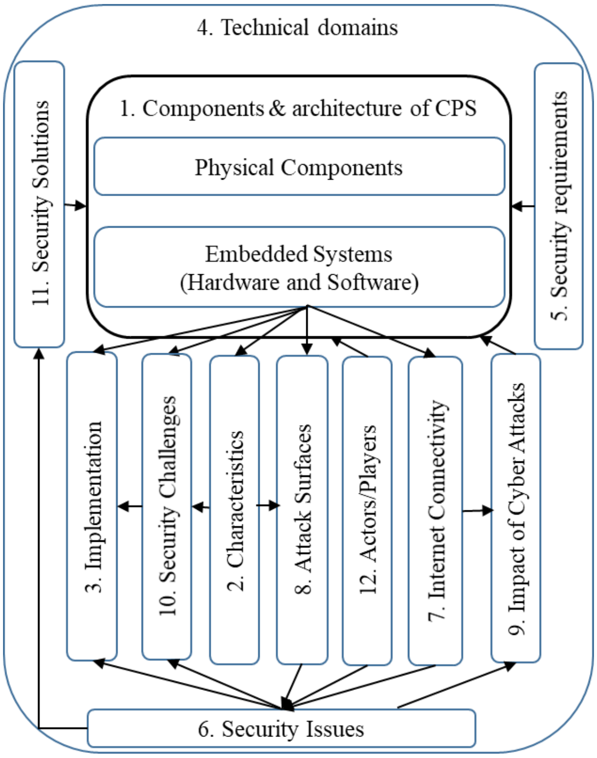

FIGURE 9. The relationship between the twelve factors.

6) THE DIVERSITY OF SECURITY PROBLEMS IN ES

Security risks of embedded systems are raised due to various factors, such as the existence of various attack surfaces in a CPS, the connectivity of the Internet, and human factors. Different application systems may have different security problems. For example, if a CPS is characterized as an autonomous system without human intervention, security problems are more likely referred to the physical security of ESs. If a CPS is characterized as an IoT-enabled system with the function of exchanging data or information with the Internet, a remote attack could threaten the CPS through the Internet. If a CPS is characterized as a system with a moveable power source and low energy utilization rate, the security problems may affect the sustainability of the energy source and the capacity of providing sufficient energy to meet the requirements of advanced applications. Embedded systems are lightweight and compact. The characteristics of ESs have given the limitations in the implementation of cybersecurity of ESs, which may require increasing the capability of the CPU to perform more complex operations, thus enabling ESs to apply complex encryption, other strong security algorithms, and so on [2], [37], [38], [37].

7) CONNECTIVITY AND TCP/IP MODEL

The connection of ESs to the Internet enables them to provide services and features that were not available without the connection. However, embedded systems could become vulnerable to a large number of cyber-attacks remotely [4]. For example, wireless connectivity can lead to the leakage of sensitive data [124].

8) ATTACK SURFACE AND CHANNELS

The main factors that could cause cyber threats or problems are: the attack surfaces which provide attackers with the entry points; network connection, which provides a way to remotely approach the system; and actors/operators, who provide opportunities to attackers for social engineering. The security problems and the characteristics of the embedded system make the implementation of a secure embedded system challengeable. Attacks are always carried out through one of the components of the targeted embedded system, such as Wi-Fi, Bluetooth, sensors or USB, as an attack surface. An attack could have different attack channels [64], [65]. Attacks on embedded systems can be carried out in different forms and on different attack surfaces and channels. Security vulnerabilities exist at different levels, and security threats come from the exploitation of existing vulnerabilities in a system.

9) IMPACT OF CYBER ATTACKS

Once an embedded system is attacked, the impact of the cyber-attack could affect the whole targeted system and the systems connected to the targeted system. Therefore, the requirements and solutions of cybersecurity should be considered from the design, implementation to responses, thus, to prevent and mitigate the damage and economic loss due to cyber-attacks. Figures 9 and 11 illustrate the relationship between the twelve factors with some examples.

Cyberattacks usually aim to damage or breach a security objective, such as confidentiality, integrity, and availability of assets or a combination of these security objectives. For example, when an attacker intends to monitor the traffic of data, violating the confidentiality of the data, the attacker also needs to breach the authentication of the connection; when an attacker gets unauthorized access to the storage of data and tampers with the stored data, this is violating the integrity and confidentiality of the data, and the impact of this attack might be on system resources in the form of increasing energy consumption or draining processor capabilities [86], [125]. A cyber-attack could produce severe consequences, e.g., the damage of all systems connected to the targeted system. Hence, it is important to consider the impacts and responses to potential cyber incidents, caused by attacks on these levels [92], [26], [85], [126], [123].

10) THE SECURITY CHALLENGES OF ES

The compromise between maintaining the characteristics of embedded systems and meeting cybersecurity requirements are the challenges of secure ESs [12], [72], [127], [128]. Therefore, there is an urgent need to find comprehensive and advanced security solutions while not draining the resources of the embedded system or conflicting with its properties. These solutions should be implemented, crossing all levels of the ES, as shown in Figure 10. In the nine levels of protection, the first level is to implement ''security by design,'' and the design should consider all cyber threats from level 2 to level 9. In section C, we will explain all these levels in more detail.

11) THE SECURITY SOLUTIONS OF ES

To ensure security solutions are effective and comprehensive, they must be compatible with the nature of the system characteristics and must be adaptive at all levels. In general, security solutions can be developed in the form of tools, methods, mechanisms or approaches. The best solution is to implement ''Security by Design'', regarding the security crossing all levels.

12) ACTORS OR PLAYERS

Manufacturers, suppliers, developers, installers, operators, and legislators play an important role to secure embedded systems. User behavior and the awareness of social-engineering-based attacks are also important [127], [129]–[133], [134]. As shown in Figures 9 and 11, the 12 aspects are strictly related to each other, and to design a protected CPS in an interconnected domain; it is important to secure ESs, as they are connected to each other and may be linked to the Internet.

C. MULTIPLE LAYERS FEEDBACK FRAMEWORK OF EMBEDDED SYSTEM CYBERSECURITY (MULFESC)

Taking the determined twelve factors into account in conjunction with the MuLFESC layers (Figure 10) will lead to building a robust and secure embedded system to the highest standards. The nine layers of exposed risks that need to be protected are depicted in Figure 10.

The MuLFESC framework consists of nine layers. These layers represent the involved entities and components in the CSES abstractly, and based on the nine layers, we can identify the vulnerabilities and cyber threats in each layer, which can be the inputs for improving the design of the system. The first layer is the initial design stage, and this layer is critical: it must be improved iteratively based on the feedback from the other eight layers to implement the ''Security by Design'' concept. The second layer (CPS) is the most abstract concept of the applications of ESs in different aspects of life. The third, fourth and fifth layers are the core of the MuLFESC framework and are the components targeted by the attacker. Therefore, better-adapted standards in the other layers will be reflected positively on these three layers. The communication layer with its protocols are the window to the outside world of CPS, and the gateway for remote exploitation. Layers 7 and 8 represent the required role of legislators, operators, users, manufacturers, developers, installers, suppliers, and service providers to set the appropriate policies that guarantee privacy, proper use, and right of access to the parties concerned in a manner that guarantees them the highest safety standards. Finally, the impact of the adopted security countermeasures taken in all layers will be monitored in the operational stage, which is considered as a real test layer of the strength and durability of security measures, and the feedback

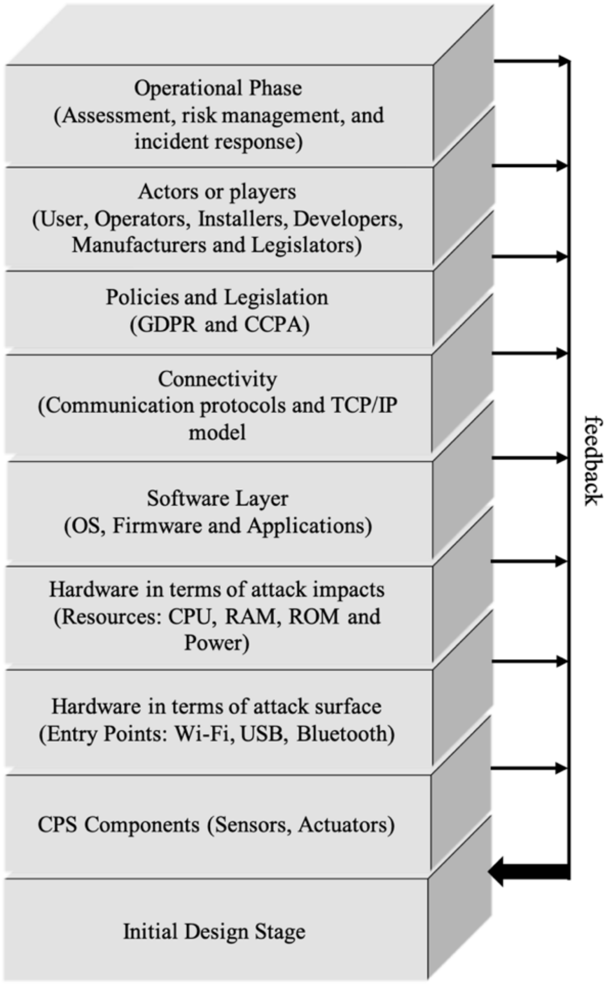
Operational Phase
(Assessment, risk management, and incident response)
Actors or players
(User, Operators, Installers, Developers, Manufacturers and Legislators)
Policies and Legislation
(GDPR and CCPA)
Connectivity
(Communication protocols and TCP/IP model
Software Layer
(OS, Firmware and Applications)
Hardware in terms of attack impacts
(Resources: CPU, RAM, ROM and Power)
Hardware in terms of attack surface
(Entry Points: Wi-Fi, USB, Bluetooth)
CPS Components (Sensors, Actuators)
Initial Design Stage
feedback

FIGURE 10. The MuLFESC framework.

must be positively reflected onto the first layer from which we set out to improve the security countermeasures.

From the protection perspective, security countermeasures should be integrated at the design stage (Layer 1) and take the properties at different layers of embedded systems into account. The ability to adopt more efficient generic security solutions means the ability to design more robust systems despite the different technical domains. Layer 2 is to secure the physical components of CPS that interact with the outside world, such as sensors and actuators; Layer 3 is to secure the attack surfaces, such as I / O modules, access points such as Wi-Fi, Bluetooth, and USB; Layer 4 is to secure the computation components that might be compromised as a result of the attack such as CPU, memory, and power source; Layer 5 is to secure the software layer, including the operating systems, firmware and applications that should be able to deal with various attacks and handle advanced security solutions; Layer 6 is to secure the Internet connection layer (TCP/IP model) by securing routes and adopting security protocols to ensure a secure transfer of data between a sender and a receiver. Therefore, it is necessary to provide the most secure communication protocols to reduce the capabilities of the attacker and to design ESs with capabilities compatible with the most secure protocols. Layer 7 is to ensure the designed embedded systems are compliant to the legislation and regulations, such as the General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA), thus ensuring the privacy and protection of users' data. Layer 8 is to regard the role of actors and players, including users, operators, manufacturers and legislators, etc.; therefore, their roles must be defined clearly to prevent cyber-attacks. Layer 9, the operational phase, should set up necessary risk management and assessment. The feedback from Layer 2 to Layer 9 can be used to improve the security design of ESs, and based on the feedback from the operational phase, the developers of CSES can refine the design of the system, thus eliminating cyber risks.

MuLFESC provides a guidance for ''Security by Design,'' which is required by industry 4.0. The design stage is essential for the implementation of CSES. The engineering cycle of CSES, such as implementation, test, and verification, are strictly required for the final security of a system, and the security of algorithms and protocols is especially critical. Optimal security solutions are the comprehensive solutions that cover all aspects at various levels. This is what was reached and extracted based on the analysis of previous studies and the extracted influencing factors. Based on that, the security framework shown in Figure 10 has been suggested as a comprehensive reference for comprehensive security assessment and solutions.

From the security risk perspective, these nine layers are exposed to many risks. The following table summarizes some security risks against each layer of the MuLFESC framework.

VI. CONCLUSION

In this paper, we conducted an analytical study in the field of cybersecurity for embedded systems in order to identify the deficiencies or gaps that need further research to improve the cybersecurity of ESs. The lack of compatible security solutions in line with the capabilities of embedded systems has provided the opportunities for attackers to find exploitable vulnerabilities and carry out various attacks. This is because the security of embedded systems is limited by their resource constraints, rather than the absence of advanced security solutions. Unfortunately, most of the advanced security solutions require a lot of computational resources and high-power consumption, so there is an urgent need to find effective and efficient solutions that do not drain the resources of the system.

Based on the architecture of ESs and the studies carried out in this field, we have identified the most critical factors that play an essential role in the cybersecurity industry for embedded systems. These factors draw the overall landscape of the cybersecurity industry for CSES, and they affect each other directly or indirectly. Also, we have reviewed the research on security risks and assessment methodologies regarding all aspects of cybersecurity of ESs and proposed a new assessment perspective within a metrics of risk assessment linked

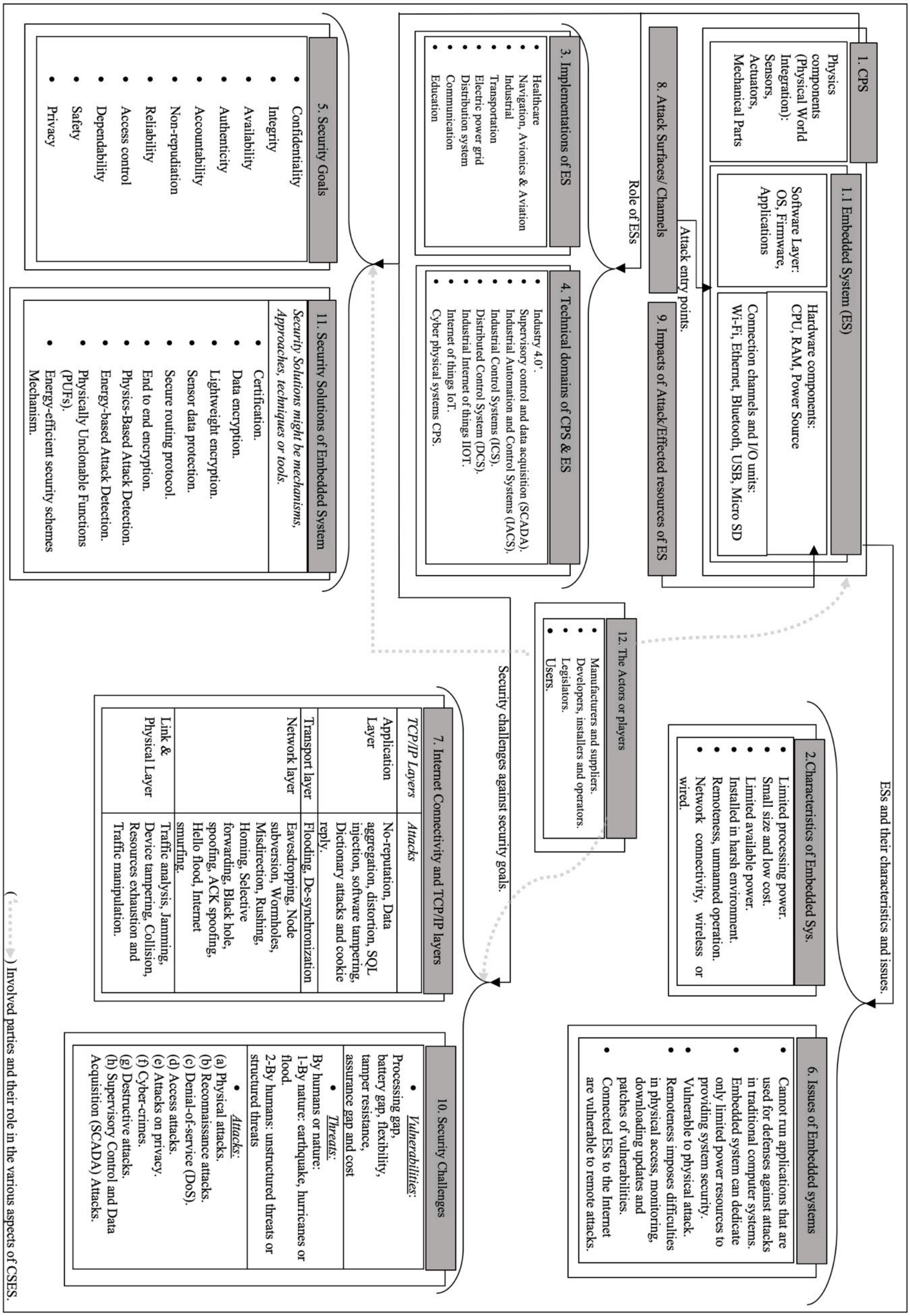
1. CPS
Physics components (Physical World Integration): Sensors, Actuators, Mechanical Parts
1.1 Embedded System (ES)
Software Layer: OS, Firmware, Applications
Hardware components: CPU, RAM, Power Source
Connection channels and I/O units: Wi-Fi, Ethernet, Bluetooth, USB, Micro SD
Attack entry points.
8. Attack Surfaces/ Channels
9. Impacts of Attack/Effected resources of ES
Role of ESs
3. Implementations of ES
• Healthcare
• Navigation, Avionics & Aviation
• Industrial
• Transportation
• Electric power grid
• Distribution system
• Communication
• Education
4. Technical domains of CPS & ES
• Industry 4.0'.
• Supervisory control and data acquisition (SCADA).
• Industrial Automation and Control Systems (IACS).
• Industrial Control Systems (ICS).
• Distributed Control System (DCS).
• Industrial Internet of things IIOT.
• Internet of things IoT.
• Cyber physical systems CPS.
5. Security Goals
• Confidentiality
• Integrity
• Availability
• Authenticity
• Accountability
• Non-repudiation
• Reliability
• Access control
• Dependability
• Safety
• Privacy
11. Security Solutions of Embedded System
Security Solutions might be mechanisms, Approaches, techniques or tools.
• Certification.
• Data encryption.
• Lightweight encryption.
• Sensor data protection.
• Secure routing protocol.
• End to end encryption.
• Physics-Based Attack Detection.
• Energy-based Attack Detection.
• Physically Unclonable Functions (PUFs).
• Energy-efficient security schemes Mechanism.
12. The Actors or players
• Manufacturers and suppliers.
• Developers, installers and operators.
• Legislators.
• Users.
ESs and their characteristics and issues.
2.Characteristics of Embedded Sys.
• Limited processing power.
• Small size and low cost.
• Limited available power.
• Installed in harsh environment.
• Remoteness, unmanned operation.
• Network connectivity, wireless or wired.
6. Issues of Embedded systems
• Cannot run applications that are used for defenses against attacks in traditional computer systems.
• Embedded system can dedicate only limited power resources to providing system security.
• Vulnerable to physical attack.
• Remoteness imposes difficulties in physical access, monitoring, downloading updates and patches of vulnerabilities.
• Connected ESs to the Internet are vulnerable to remote attacks.
Security challenges against security goals.
7. Internet Connectivity and TCP/IP layers
TCP/IP Layers
Attacks
Application Layer
No-reputation, Data aggregation, distortion, SQL injection, software tampering, Dictionary attacks and cookie reply.
Transport layer
Flooding, De-synchronization
Network layer
Eavesdropping, Node subversion, Wormholes, Misdirection, Rushing, Homing, Selective forwarding, Black hole, spoofing, ACK spoofing, Hello flood, Internet smurfing.
Link & Physical Layer
Traffic analysis, Jamming, Device tampering, Collision, Resources exhaustion and Traffic manipulation.
10. Security Challenges
• Vulnerabilities:
Processing gap, battery gap, flexibility, tamper resistance, assurance gap and cost
• Threats:
By humans or nature:
1-By nature: earthquake, hurricanes or flood.
2-By humans: unstructured threats or structured threats
• Attacks:
(a) Physical attacks.
(b) Reconnaissance attacks.
(c) Denial-of-service (DoS).
(d) Access attacks.
(e) Attacks on privacy.
(f) Cyber-crimes.
(g) Destructive attacks.
(h) Supervisory Control and Data Acquisition (SCADA) Attacks.
( ) Involved parties and their role in the various aspects of CSES.

FIGURE 11. Overall landscape of Cyber Security of Embedded Systems (CSES).

TABLE 4. Security risks against MuLFESC layers.

MuLFESC Layers	Related security risks and challenges
Operational layer	Absence of service desk, customer service, customer servers, after-sales services, warranty services, unmonitored and unmaintained system. Absence of network operations center (NOC), and security operations center (SOC), patches delay, absence of feedback-cycle system, and neglected complaints.
Actors or players (Involved parties)	Absence of unified standards, untrained operators and users, social engineering attack awareness, platform incompatibility problems, lack of regular updates.
Policies and legislations	Absence of regulations, policies, and legislation, such as GDPR, CCPA, unsigned responsibilities, undesignated roles, lack of manuals and procedures, and unclear job description.
Connectivity layer	Blackhole attacks, traffic analysis, jamming, device tampering, collision and traffic manipulation, misdirection, flooding, DoS attacks, eavesdropping, spoofing.
Software layer	SQL injection, software tampering, malicious software, remote code execution (RCE) attacks, password attacks, dictionary attacks, brute force attacks.
Hardware layer	Resources exhaustion, side-channel attacks (SCA), differential power analysis (DPA) attacks.
Attack surface	Unprotected entry points, enabling unnecessary ports, features, and services.
CPS components (Sensor, actuators)	Simple electromagnetic analysis (SEMA) attacks, differential electromagnetic analysis (DEMA) attacks, intentional electromagnetic interference (IEMI) attacks, frequency-constrained sensor and actuator attacks, unintended emissions or emanating spurious transmissions attacks.
Design stage	Neglecting security requirements, ignoring security-by-design principles, specs incompatible with purpose of use.

to MuLFESC. The determined influencing factors have been employed to shape the Multiple Layers Feedback Framework of Embedded System Cybersecurity (MuLFESC) in line with the security risk metrics model.

The proposed MuLFESC Framework could contribute to the implementation of comprehensive and effective ''Security by Design'' solutions by providing feedback to the design stage of CSES.

Overall, taking the identified key factors, the proposed MuLFESC, the risk assessment metrics, and all involved parties of CSES into account will facilitate the mission for security practitioners to carry out a comprehensive assessment. Thus, more efficient application-specific security solutions can be designed, thereby improving CPS cybersecurity.

APPENDIX

See Figure 11.

REFERENCES

[1] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, and B. Gabrys, ''The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 1015–1021, doi: 10.1109/CEC.2016.7743900.

[2] E. Levy, ''Crossover: Online pests plaguing the off line world,'' *IEEE Secur. Privacy*, vol. 1, no. 6, pp. 71–73, Nov. 2003, doi: 10.1109/MSECP.2003.1253573.

[3] M. Patrick. *How MCUs Actually Fight Security Attacks on Embedded Systems, New Electronics, the site for electronic design engineers*. Accessed: Nov.2, 2019. [Online]. Available: http://www.newelectronics.co.uk/electronics-technology/how-mcus-actually-fight-security-attacks-on-embedded-systems/176924/

[4] S. Ali, T. Al Balushi, Z. Nadir, and O. K. Hussain, ''ICS/SCADA system security for CPS,'' in *Cyber Security for Cyber Physical Systems* (Studies in Computational Intelligence), vol. 768. Muscat, Oman: Springer-Verlag, 2018, pp. 89–113. [Online]. Available: https://squ.pure.elsevier.com/en/publications/icsscada-system-security-for-cps, doi: 10.1007/978-3-319-75880-0_5.

[5] R. M. Lee, M. J. Assante, and T. Conway, ''German steel mill cyber attack,'' ICS Defense Use Case (DUC), SANS Ind. Control Syst. (ICS), Swansea, U.K., Tech. Rep., Dec. 2014. [Online]. Available: https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[6] F. V. and T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*. Hoboken, NJ, USA: Wiley, 2002.

[7] D. Papp, Z. Ma, and L. Buttyan, ''Embedded systems security: Threats, vulnerabilities, and attack taxonomy,'' in *Proc. 13th Annu. Conf. Privacy, Secur. Trust (PST)*, Jul. 2015, pp. 145–152, doi: 10.1109/PST.2015.7232966.

[8] P. V. Pham Van and N. N. Binh, ''Embedded system architecture design and optimization at the model level,'' *Int. J. Comput. Commun. Eng.*, vol. 1, no. 5, pp. 345–349, Nov. 2012, doi: 10.7763/ijcce.2012.v1.87.

[9] A. A. Suzen, B. Duman, and B. Sen, ''Benchmark analysis of jetson TX2, jetson nano and raspberry PI using deep-CNN,'' in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (HORA)*, Jun. 2020, pp. 1–5, doi: 10.1109/HORA49412.2020.9152915.

[10] *Cyber-Physical Systems (CPS) Nsf08611*. Accessed: Sep. 4, 2019. [Online]. Available: https://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm

[11] P. Marwedel and P. Marwedel, ''Embedded system hardware,'' in *Proc. Embedded Syst. Design*, 2011, pp. 119–175.

[12] E. A. Lee, ''Cyber physical systems: Design challenges,'' in *Proc. 11th IEEE Int. Symp. Object Component-Oriented Real-Time Distrib. Comput. (ISORC)*, May 2008, pp. 363–369, doi: 10.1109/ISORC.2008.25.

[13] S. Hameed, F. I. Khan, and B. Hameed, ''Understanding security requirements and challenges in Internet of Things (IoT): A review,'' *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–14, Jan. 2019, doi: 10.1155/2019/9629381.

[14] *Cyber-Physical Systems|Digital Single Market*. Accessed: Aug. 26, 2019. [Online]. Available: https://ec.europa.eu/digital-single-market/en/cyber-physical-systems

[15] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, ''The industrial Internet of Thing (IIoT): An analysis framework,'' *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018, doi: 10.1016/j.compind.2018.04.015.

[16] S. I. Tay, T. C. Lee, N. Z. A. Hamid, and A. N. A. Ahmad, ''An overview of industry 4.0: Definition, components, and government initiatives,'' *J. Adv. Res. Dyn. Control Syst.*, vol. 10, no. 14, pp. 1379–1387, 2018.

[17] S. Ntalampiras, ''Automatic identification of integrity attacks in cyber-physical systems,'' *Expert Syst. Appl.*, vol. 58, pp. 164–173, Oct. 2016, doi: 10.1016/j.eswa.2016.04.006.

[18] A. Pasquini, *Computer Safety, Reliability and Security*, vol. 1698. Berlin, Germany: Springer, 1999. [Online]. Available: https://link.springer.com/book/10.1007%2F3-540-48249-0, doi: 10.1007/978-3-319-24249-1.

[19] S. Ravi, P. Kocher, R. Lee, G. McGraw, and A. Raghunathan, ''Security as a new dimension in embedded system design,'' in *Proc. 41st Annu. Conf. Desigh Autom.*, 2004, p. 753, doi: 10.1145/996566.996771.

[20] V. H. MESHRAM and A. B. SASANKAR, "Security in embedded systems: Vulnerabilities, pigeonholing of attacks and countermeasures," in *Proc. IOSR J. Comput. Eng.*, 2016, pp. 11–15.

[21] C. Bodei, S. Chessa, and L. Galletta, "Measuring security in IoT communications," *Theor. Comput. Sci.*, vol. 764, pp. 100–124, Apr. 2019, doi: 10.1016/j.tcs.2018.12.002.

[22] S. Parameswaran and T. Wolf, "Embedded systems security—An overview," *Design Autom. Embedded Syst.*, vol. 12, no. 3, pp. 173–183, Sep. 2008, doi: 10.1007/s10617-008-9027-x.

[23] D. D. Hwang, P. Schaumont, K. Tiri, and I. Verbauwhede, "Securing embedded systems," *IEEE Secur. Privacy Mag.*, vol. 4, no. 2, pp. 40–49, Mar. 2006.

[24] M. Abomhara and G. M. Káien, "Cyber security and the Internet of Things: Vulnerabilities, threats, intruders and attacks," *J. Cyber Secur. Mobility*, vol. 4, no. 1, pp. 65–88, 2015, doi: 10.13052/jcsm2245-1439.414.

[25] P. Marwedel, *Embedded System Design*. Cham, Switzerland: Springer, 2018.

[26] H. Chai, G. Zhang, J. Zhou, J. Sun, L. Huang, and T. Wang, "A short review of security-aware techniques in real-time embedded systems," *J. Circuits, Syst. Comput.*, vol. 28, no. 2, Feb. 2019, Art. no. 1930002, doi: 10.1142/S0218126619300022.

[27] W. Wang, X. Zhang, Q. Hao, Z. Zhang, B. Xu, H. Dong, T. Xia, and X. Wang, "Hardware-enhanced protection for the runtime data security in embedded systems," *Electronics*, vol. 8, no. 1, p. 52, Jan. 2019, doi: 10.3390/electronics8010052.

[28] S. Ali, *Cyber Security for Cyber Physical Systems*, vol. 768. Springer, 2018, pp. 11–33, doi: 10.1007/978-3-319-75880-0_2.

[29] G. Loukas, E. Karapistoli, E. Panaousis, P. Sarigiannidis, A. Bezemskij, and T. Vuong, "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles," *Ad Hoc Netw.*, vol. 84, pp. 124–147, Mar. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870518307091?via%3Dihub, doi: 10.1016/j.adhoc.2018.10.002.

[30] H. Habibzadeh, B. H. Nussbaum, F. Anjomshoa, B. Kantarci, and T. Soyata, "A survey on cybersecurity, data privacy, and policy issues in cyber-physical system deployments in smart cities," *Sustain. Cities Soc.*, vol. 50, Oct. 2019, Art. no. 101660, doi: 10.1016/j.scs.2019.101660.

[31] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the Internet of Things: Challenges, threats and solutions," *Internet Things*, vol. 5, pp. 41–70, Mar. 2019, doi: 10.1016/j.iot.2018.11.003.

[32] A. Riahi Sfar, E. Natalizio, Y. Challal, and Z. Chtourou, "A roadmap for security challenges in the Internet of Things," *Digit. Commun. Netw.*, vol. 4, no. 2, pp. 118–137, Apr. 2018, doi: 10.1016/j.dcan.2017.04.003.

[33] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems Security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017, doi: 10.1109/JIOT.2017.2703172.

[34] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A security framework for the Internet of Things in the future Internet architecture," *Future Internet*, vol. 9, no. 3, p. 27, Jun. 2017, doi: 10.3390/fi9030027.

[35] P. Koopman, "Embedded system security," *Computer*, vol. 37, no. 7, pp. 95–97, Jul. 2004.

[36] C. O'Flynn and Z. Chen, "ChipWhisperer: An open-source platform for hardware embedded security research," in *Proc. Int. Workshop Constructive Side-Channel Anal. Secure Desigh*, 2014, pp. 243–260.

[37] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "Taxonomy for description of cross-domain attacks on CPS," in *Proc. 2nd ACM Int. Conf. High confidence Netw. Syst.*, 2013, pp. 135–142, doi: 10.1145/2461446.2461465.

[38] L. Khelladi, Y. Challal, A. Bouabdallah, and N. Badache, "On security issues in embedded systems: Challenges and solutions," *Int. J. Inf. Comput. Secur.*, vol. 2, no. 2, p. 140, 2008, doi: 10.1504/IJICS.2008.018515.

[39] G. Bansod, N. Raval, and N. Pisharoty, "Implementation of a new lightweight encryption design for embedded security," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 142–151, Jan. 2015, doi: 10.1109/TIFS.2014.2365734.

[40] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 1, pp. 27–40, Jan. 2017, doi: 10.1109/JAS.2017.7510349.

[41] J. M. Kizza, "Computer and Network Forensics," in *A Guide to Computer Network Security*, London, U.K.: Springer, 2009, pp. 299–328.

[42] E. Bertino, L. D. Martino, F. Paci, and A. C. Squicciarini, "Web services threats, vulnerabilities, and countermeasures," in *Security for Web Services Service-Oriented Architectures*. Berlin, Germany: Springer, 2009, pp. 25–44.

[43] H. G. Brauch, "Concepts of security threats, challenges, vulnerabilities and risks," in *Coping With Global Environmental Change, Disasters and Security*. Berlin, Germany: Springer, vol. 11. pp. 61–106. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-642-17776-7_2, doi: 10.1007/978-3-642-17776-7_2.

[44] K. Dahbur, B. Mohammad, and A. B. Tarakji, "A survey of risks, threats and vulnerabilities in cloud computing," in *Proc. Int. Conf. Intell. Semantic Web-Services Appl.*, 2011, pp. 1–6, doi: 10.1145/1980822.1980834.

[45] A. Yan. *Introduction to Information Sys—R. Kelly Rainer (1)*. Accessed: Aug. 25, 2019. [Online]. Available: https://www.academia.edu/28734440/Introduction_to_Information_Sys_-_R._Kelly_Rainer_1_

[46] F. Li, A. Lai, and D. Ddl, "Evidence of advanced persistent threat: A case study of malware for political espionage," in *Proc. 6th Int. Conf. Malicious Unwanted Softw.*, Oct. 2011, pp. 102–109, doi: 10.1109/MALWARE.2011.6112333.

[47] S. Ansari, S. G. Rajeev, and H. S. Chandrashekar, "Packet sniffing: A brief introduction," *IEEE Potentials*, vol. 21, no. 5, pp. 17–19, Dec. 2002, doi: 10.1109/MP.2002.1166620.

[48] M. de Vivo, E. Carrasco, G. Isern, and G. O. de Vivo, "A review of port scanning techniques," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 2, pp. 41–48, Apr. 1999, doi: 10.1145/505733.505737.

[49] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in SCADA networks," *Comput. Secur.*, vol. 25, no. 7, pp. 498–506, Oct. 2006, doi: 10.1016/j.cose.2006.03.001.

[50] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "SCADA security in the light of cyber-warfare," *Comput. Secur.*, vol. 31, no. 4, pp. 418–436, Jun. 2012, doi: 10.1016/j.cose.2012.02.009.

[51] C. Wilson. (2008). *Botnets, Cybercrime, and Cyberterrorism: Vulnerabilities and Policy Issues for Congress*. Accessed: Aug. 26, 2019. [Online]. Available: https://apps.dtic.mil/docs/citations/ADA477642

[52] I. Naumann and G. Hogben, "Privacy features of European eID card specifications," *Netw. Secur.*, vol. 2008, no. 8, pp. 9–13, Aug. 2008, doi: 10.1016/S1353-4858(08)70097-7.

[53] T. Fitz, M. Theiler, and K. Smarsly, "A metamodel for cyber-physical systems," *Adv. Eng. Informat.*, vol. 41, Aug. 2019, Art. no. 100930, doi: 10.1016/j.aei.2019.100930.

[54] NIST. (2014). *National Institute of Standards and Technology (NIST) Cybersecurity Risk Management Framework Applied to Modern Vehicles*. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/812073_natlinstitstandardstechcyber.pdf.

[55] *NIST*. Accessed: Oct. 16, 2019. [Online]. Available: https://www.nist.gov/

[56] D. S. Pallett, "National institute of standards and technology (NIST)," in *Proc. workshop Speech Natural Lang.*, 1989, p. 191, doi: 10.3115/100964.1138540.

[57] I. P. Draft. (2017). *NIST Guide to Supervisory and Data Acquisition-SCADA and Industrial Control Systems Security*. [Online]. Available: http://www.cyber.st.dhs.gov/docs/NIST

[58] E. Chew, "Performance measurement guide for information security," Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, NIST Special Publication 800-55 Revision 1, Jul. 2008. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-55r1.pdf

[59] *NIST Special Publication 800-181: National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework*, NIST, Gaithersburg, MD, USA, vol. 144, 2017, doi: 10.6028/NIST.SP.800-181.

[60] M. G. Williams, "A risk assessment on raspberry PI using NIST standards," *Int. J. Comput. Sci. Netw. Secur.*, vol. 15, no. 6, pp. 22–30, 2015.

[61] B. Potteiger, W. Emfinger, H. Neema, X. Koutosukos, C. Tang, and K. Stouffer, "Evaluating the effects of cyber-attacks on cyber physical systems using a hardware-in-the-loop simulation testbed," in *Proc. Resilience Week (RWS)*, Sep. 2017, pp. 177–183, doi: 10.1109/RWEEK.2017.8088669.

[62] W. Pieters, "Defining 'the weakest link': Comparative security in complex systems of systems," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2013, pp. 39–44, doi: 10.1109/CloudCom.2013.101.

[63] K. Georgiou, S. Xavier-de-Souza, and K. Eder, "The IoT energy challenge: A software perspective," *IEEE Embedded Syst. Lett.*, vol. 10, no. 3, pp. 53–56, Sep. 2018, doi: 10.1109/LES.2017.2741419.

[64] J. A. Ambrose, R. G. Ragel, D. Jayasinghe, T. Li, and S. Parameswaran, "Side channel attacks in embedded systems: A tale of hostilities and deterrence," in *Proc. 16th Int. Symp. Qual. Electron. Desigh*, Mar. 2015, pp. 452–459, doi: 10.1109/ISQED.2015.7085468.

[65] S. Bhunia and M. Tehranipoor, Eds., "Chapter 1—Introduction to hardware security," in *Hardware Security*. Cambridge, MA, USA: Morgan Kaufmann, 2019, pp. 1–20. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012812477200006X, doi: 10.1016/B978-0-12-812477-2.00006-X.

[66] J. M. Kizza, "Understanding computer network security," in *Guide to Computer Network Security* (Computer Communications and Networks). London, U.K.: Springer, 2013. [Online]. Available: https://link.springer.com/book/10.1007%2F978-1-4471-4543-1, doi: 10.1007/978-1-4471-4543-1_2.

[67] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Comput. Ind.*, vol. 100, pp. 212–223, Sep. 2018, doi: 10.1016/j.compind.2018.04.017.

[68] D. N. Serpanos and A. G. Voyiatzis, "Security challenges in embedded systems," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1s, pp. 1–10, Mar. 2013, doi: 10.1145/2435227.2435262.

[69] ISO. (2013). *ISO/IEC 27001:2013*. Accessed: Nov. 9, 2019. [Online]. Available: https://www.iso.org/standard/54534.html

[70] I. Hsu, C.-Y. Chen, S. Lin, T.-J. Yu, N. Cho, and M.-C. Hsieh, "7nm chip-package interaction study on a fine pitch flip chip package with laser assisted bonding and mass reflow technology," in *Proc. IEEE 69th Electron. Compon. Technol. Conf. (ECTC)*, May 2019, pp. 289–293, doi: 10.1109/ECTC.2019.00050.

[71] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," *Annu. Rev. Control*, vol. 47, pp. 394–411, Dec. 2019, doi: 10.1016/j.arcontrol.2019.04.011.

[72] Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security: Analysis, challenges and solutions," *Comput. Secur.*, vol. 68, pp. 81–97, Jul. 2017, doi: 10.1016/j.cose.2017.04.005.

[73] T. Poongothai and K. Duraiswamy, "Intrusion detection in mobile AdHoc networks using machine learning approach," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2014, pp. 1–5, doi: 10.1109/ICICES.2014.7033949.

[74] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 847–855, Jun. 2013, doi: 10.1109/TSG.2012.2226919.

[75] D. Trawczynski, J. Zalewski, and J. Sosnowski, "Design of reactive security mechanisms in time-triggered embedded systems," *SAE Int. J. Passenger Cars—Electron. Electr. Syst.*, vol. 7, no. 2, pp. 527–535, Apr. 2014, doi: 10.4271/2014-01-0341.

[76] R. M. Yousufi, P. Lalwani, and M. B. Potdar, "A network-based intrusion detection and prevention system with multi-mode counteractions," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Mar. 2017, pp. 1–6, doi: 10.1109/ICIIECS.2017.8276023.

[77] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Sep. 2018, doi: 10.1145/3203245.

[78] E. Eyisi and X. Koutsoukos, "Energy-based attack detection in networked control systems," in *Proc. 3rd Int. Conf. High Confidence Netw. Syst.*, 2014, pp. 115–124, doi: 10.1145/2566468.2566472.

[79] Z. Gu, G. Han, H. Zeng, and Q. Zhao, "Security-aware mapping and scheduling with hardware co-processors for FlexRay-based distributed embedded systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 3044–3057, Oct. 2016, doi: 10.1109/TPDS.2016.2520949.

[80] J.-L. Zhang, W.-Z. Wang, X.-W. Wang, and Z.-H. Xia, "Enhancing security of FPGA-based embedded systems with combinational logic binding," *J. Comput. Sci. Technol.*, vol. 32, no. 2, pp. 329–339, Mar. 2017, doi: 10.1007/s11390-017-1700-8.

[81] V. Negi, H. Verma, I. Singh, A. Vikram, K. Malik, A. Singh, and G. Verma, "Network security in embedded system using TLS," *Int. J. Secur. Appl.*, vol. 10, no. 2, pp. 375–384, Feb. 2016, doi: 10.14257/ijsia.2016.10.2.33.

[82] C. Manifavas, K. Fysarakis, A. Papanikolaou, and I. Papaefstathiou, "Embedded systems security: A survey of EU research efforts," *Secur. Commun. Netw.*, vol. 8, no. 11, pp. 2016–2036, Jul. 2015, doi: 10.1002/sec.1151.

[83] S. Lukacs, A. V. Lutas, D. H. Lutas, and G. Sebestyen, "Hardware virtualization based security solution for embedded systems," in *Proc. IEEE Int. Conf. Autom., Qual. Test., Robot.*, May 2014, pp. 1–6, doi: 10.1109/AQTR.2014.6857879.

[84] H. Elmiligi, F. Gebali, and M. W. El-Kharashi, "Multi-dimensional analysis of embedded systems security," *Microprocessors Microsyst.*, vol. 41, pp. 29–36, Mar. 2016, doi: 10.1016/j.micpro.2015.12.005.

[85] H. Read, I. Sutherland, K. Xynos, and F. Roarson, "Locking out the investigator: The need to circumvent security in embedded systems," *Inf. Secur. J., Global Perspective*, vol. 24, nos. 1–3, pp. 39–47, Jul. 2015, doi: 10.1080/19393555.2014.998847.

[86] C. Moreno and S. Fischmeister, "Non-intrusive runtime monitoring through power consumption to enforce safety and security properties in embedded systems," *Formal Methods Syst. Design*, vol. 53, no. 1, pp. 113–137, Aug. 2018, doi: 10.1007/s10703-017-0298-3.

[87] I. Garitano, S. Fayyad, and J. Noll, "Multi-metrics approach for security, privacy and dependability in embedded systems," *Wireless Pers. Commun.*, vol. 81, no. 4, pp. 1359–1376, Apr. 2015, doi: 10.1007/s11277-015-2478-z.

[88] D. Mu, B. Ma, B. Mao, and W. Hu, "A bottom-up approach to verifiable embedded system information flow security," *IET Inf. Secur.*, vol. 8, no. 1, pp. 12–17, Jan. 2014, doi: 10.1049/iet-ifs.2012.0342.

[89] L. Shian, "Design and development of a security kernel in an embedded system," *Int. J. Control Autom.*, vol. 7, no. 11, pp. 49–58, Nov. 2014, doi: 10.14257/ijca.2014.7.11.06.

[90] M. Vasilevskaya, L. A. Gunawan, S. Nadjm-Tehrani, and P. Herrmann, "Integrating security mechanisms into embedded systems by domain-specific modelling," *Secur. Commun. Netw.*, vol. 7, no. 12, pp. 2815–2832, Dec. 2014, doi: 10.1002/sec.819.

[91] E. Thayer, "Adversarial Testing to Increase the Overall Security of Embedded Systems: A Review of the Process," *IEEE Control Syst.*, vol. 37, no. 2, pp. 104–108, Apr. 2017, doi: 10.1109/MCS.2016.2643258.

[92] J. Hasler and S. Shah, "Security implications for ultra-low power configurable SoC FPAA embedded systems," *J. Low Power Electron. Appl.*, vol. 8, no. 2, p. 17, Jun. 2018, doi: 10.3390/jlpea8020017.

[93] Y. Mahmoodi, S. Reiter, A. Viehl, O. Bringmann, and W. Rosenstiel, "Attack surface modeling and assessment for penetration testing of IoT system designs," in *Proc. 21st Euromicro Conf. Digit. Syst. Desigh DSD*, vol. 2018, 2018, pp. 177–181, doi: 10.1109/DSD.2018.00043.

[94] S. Checkoway, "Automotive attack surfaces," in *Proc. USENIX Secur.*, 2011, pp. 1–5, doi: 10.1109/TITS.2014.2342271.

[95] A. Bartel, J. Klein, Y. Le Traon, and M. Monperrus, "Automatically securing permission-based software by reducing the attack surface: An application to android," in *Proc. 27th IEEE/ACM Int. Conf. Autom. Softw. Eng.*, 2012, pp. 274–277, doi: 10.1145/2351676.2351722.

[96] L. Wilbanks, "Whats your IT risk approach?" *IT Prof.*, vol. 20, no. 4, pp. 13–17, Jul. 2018, doi: 10.1109/MITP.2018.043141663.

[97] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer security incident handling guide?: Recommendations of the National Institute of Standards and Technology," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST Special Publication 800-61 Revision 2, 2012, doi: 10.6028/NIST.SP.800-61r2.

[98] J. Creasy. (2013). *Cyber Security Incident Response Guide*. [Online]. Available: http://www.crest-approved.org/wp-content/uploads/CSIR-Procurement-Guide.pdf

[99] A. Dorofee. (2018). *Incident Management Capability Assessment*. [Online]. Available: https://resources.sei.cmu.edu/asset_files/TechnicalReport/2018_005_001_538866.pdf

[100] S. Sultana, D. Midi, and E. Bertino, "Kinesis: A security incident response and prevention system for wireless sensor networks," in *Proc. 12th ACM Conf. Embedded Netw. Sensor Syst.*, 2014, pp. 148–162, doi: 10.1145/2668332.2668351.

[101] *Crest*. Accessed: Oct. 16, 2019. [Online]. Available: https://www.crest-approved.org/index.html

[102] *CERT Division*. Accessed: Oct. 16, 2019).[Online]. Available: https://www.sei.cmu.edu/about/divisions/cert/

[103] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper resistance mechanisms for secure embedded systems," in *Proc. 17th Int. Conf. VLSI Desigh Proc.*, 2004, pp. 605–611, doi: 10.1109/icvd.2004.1260985.

[104] P. Gupta, S. Ravi, A. Raghunathan, and N. K. Jha, "Efficient fingerprint-based user authentication for embedded systems," in *Proc. 42nd Desigh Autom. Conf.*, 2005, pp. 244–247.

[105] M. Sveda and V. Oplustil, "Experience with integration and certification of COTS based embedded system into advanced avionics system," in *Proc. Int. Symp. Ind. Embedded Syst.*, Jul. 2007, pp. 282–287, doi: 10.1109/SIES.2007.4297346.

[106] A. Ukil, J. Sen, and S. Koilakonda, "Embedded security for Internet of Things," in *Proc. 2nd Nat. Conf. Emerg. Trends Appl. Comput. Sci.*, Mar. 2011, pp. 50–55, doi: 10.1109/NCETACS.2011.5751382.

[107] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of autonomous systems employing embedded computing and sensors," *IEEE Micro*, vol. 33, no. 1, pp. 80–86, Jan. 2013, doi: 10.1109/MM.2013.18.

[108] M. Al-Haidary and Q. Nasir, "Physically unclonable functions (PUFs): A systematic literature review," in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, Mar. 2019, pp. 1–6, doi: 10.1109/ICASET.2019.8714431.

[109] N. Alimohammadi and S. B. Shokouhi, "Secure hardware key based on physically unclonable functions and artificial neural network," in *Proc. 8th Int. Symp. Telecommun. (IST)*, Sep. 2016, pp. 756–760, doi: 10.1109/ISTEL.2016.7881924.

[110] S. Nimgaonkar, M. Gomathisankaran, and S. P. Mohanty, "TSV: A novel energy efficient memory integrity verification scheme for embedded systems," *J. Syst. Archit.*, vol. 59, no. 7, pp. 400–411, Aug. 2013, doi: 10.1016/j.sysarc.2013.04.008.

[111] A. Venäkauskas, N. Jusas, E. Kazanaviàius, and V. Štuikys, "An energy efficient protocol for the Internet of Things," *J. Electr. Eng.*, vol. 66, no. 1, pp. 47–52, Jan. 2015, doi: 10.1515/jee-2015-0007.

[112] D. Xu, "Virtualization and security," in *Proc. 4th ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, Mar. 2014, pp. 73–74. [Online]. Available: https://dl.acm.org/doi/10.1145/2557547.2557590, doi: 10.1145/2557547.2557590.

[113] D. Jha and B. Shahi, "A proposed methodology for end to end encryption for communicating embedded systems," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Mar. 2017, pp. 1–3, doi: 10.1109/ICIIECS.2017.8275910.

[114] R. Chatterjee, R. Chakraborty, and J. K. Mandal, "Design of cryptographic model for End-to-End encryption in FPGA based systems," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2019, pp. 459–465, doi: 10.1109/ICCMC.2019.8819761.

[115] Y. Cheng, J. Deng, J. Li, S. A. Deloach, A. Singhal, and X. Ou, "Metrics of security," *Adv. Inf. Secur.*, vol. 62, pp. 263–295, Dec. 2014, doi: 10.1007/978-3-319-11391-3_13.

[116] (2019). *Educause*. Accessed: Nov. 9, 2019. [Online]. Available: https://library.educause.edu/topics/cybersecurity/security-metrics

[117] S. Biswas and R. Chellappa, *Encyclopedia of Cryptography and Security*, vol. 20, no. 6. Boston, MA, USA: Springer, 2011.

[118] R. Ma, P. Cheng, Z. Zhang, W. Liu, Q. Wang, and Q. Wei, "Stealthy attack against redundant controller architecture of industrial cyber-physical system," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9783–9793, Dec. 2019, doi: 10.1109/jiot.2019.2931349.

[119] A. Elahi, *Computer Systems*. Cham, Switzerland: Springer, 2018. [Online]. Available: https://link.springer.com/book/10.1007%2F978-3-319-66775-1, doi: 10.1007/978-3-319-66775-1.

[120] S. Sudhakar, E. P. Kumar, and S. Thiyagarajan, "Border security and multi access robot using embedded system," *Indian J. Sci. Technol.*, vol. 9, no. 16, May 2016, doi: 10.17485/ijst/2016/v9i16/92205.

[121] C. Britto. (2014). *International Journal of Innovative Technology and Research*. Accessed: Sep. 8, 2019. [Online]. Available: https://plu.mx/plum/a/?elsevier_id=2-s2.0-85061817104&theme=plum-scopus-theme

[122] K. Kimani, V. Oduol, and K. Langat, "Cyber security challenges for IoT-based smart grid networks," *Int. J. Crit. Infrastruct. Protection*, vol. 25, pp. 36–49, Jun. 2019, doi: 10.1016/j.ijcip.2019.01.001.

[123] M. Waidner and M. Kasper, "Security in industrie 4.0—Challenges and solutions for the fourth industrial revolution," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2016, pp. 1303–1308, doi: 10.3850/9783981537079_1005.

[124] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming channels," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 163–177, doi: 10.1145/3243734.3243802.

[125] A. Stáhring, G. Ehmen, and S. Fröschle, "Analyzing the impact of injected sensor data on an advanced driver assistance system using the OP2TIMUS prototyping platform," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2016, pp. 523–526, doi: 10.3850/9783981537079_0361.

[126] A. P. Fournaris and N. Sklavos, "Secure embedded system hardware design—A flexible security and trust enhanced approach," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 121–133, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.011.

[127] W. A. Arbaugh and L. van Doorn, "Embedded security: Challenges and concerns," *Computer*, vol. 34, no. 10, pp. 40–41, Oct. 2001, doi: 10.1109/MC.2001.955096.

[128] S. Han, M. Xie, H.-H. Chen, and Y. Ling, "Intrusion detection in cyber-physical systems: Techniques and challenges," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1052–1062, Dec. 2014, doi: 10.1109/jsyst.2013.2257594.

[129] T. Sakuneka, A. Marnewick, and J.-H. Pretorius, "Industry 4.0 competencies for a control systems engineer," in *Proc. IEEE Technol. Eng. Manag. Conf.*, Dec. 2019, pp. 1–6, doi: 10.1109/temscon.2019.8813717.

[130] C. Baron and B. Daniel-Allegro, "About adopting a systemic approach to design connected embedded systems: A MOOC promoting systems thinking and systems engineering," *Syst. Eng.*, vol. 15, pp. 1–20, Sep. 2019, doi: 10.1002/sys.21513.

[131] G. Mohay, "Technical challenges and directions for digital forensics," in *Proc. 1st Int. Work. Syst. Approaches to Digit. Forensic Eng.*, 2005, pp. 155–161, 2005, doi: 10.1109/SADFE.2005.24.

[132] W. Hasselbring and R. Reussner, "Toward trustworthy software systems," *Computer*, vol. 39, no. 4, pp. 91–92, Apr. 2006, doi: 10.1109/MC.2006.142.

[133] T. M. Van Engers and E. Glassée, "Facilitating the legislation process using a shared conceptual model," *IEEE Intell. Syst. Their Appl.*, vol. 16, no. 1, pp. 50–55, 2001, doi: 10.1109/5254.912385.

[134] G. Hatzivasilis, I. Papaefstathiou, and C. Manifavas, "Software security, privacy, and dependability: Metrics and measurement," *IEEE Softw.*, vol. 33, no. 4, pp. 46–54, Jul. 2016, doi: 10.1109/MS.2016.61.

ABDULMOHSAN ALOSEEL received the B.S. degree in management information systems (MIS) from King Faisal University, in 2006, and the M.S. degree in computer and network security from Middlesex University London, London, in 2015. He is currently pursuing the Ph.D. degree in cybersecurity of embedded systems with the School of Aerospace, Transport and Manufacturing (SATM), Cranfield University. He has been working as an IT Officer with the Royal Saudi Air Force (RSAF) since 2007.

HONGMEI HE (Senior Member, IEEE) received the B.Eng. degree in electronics and computer engineering from Anhui Polytechnic University, Wuhu, China, in 1997, and the M.Sc. degree in multimedia and internet computing and the Ph.D. degree in computer science from Loughborough University, Loughborough, U.K., in 2003 and 2006, respectively. She is currently a Lecturer of AI and cybersecurity with Cranfield University, Cranfield, U.K. Previously, she was a Research Fellow with the University of Kent from January 2012 to October 2013, the PDRA, University of Ulster, from April 2011 to December 2011, and the University of Bristol, from January 2007 to March 2011. Before coming to the U.K., she was a Senior Embedded System Engineer with Motorola Design House, Shenzhen, China. Her expertise in AI has been explored in a wide arrange of applications, such as cognitive robotics, cognitive cybersecurity, data/sensor fusion, cloud resource allocation, flood prediction, computational finance, and graph drawings. Her current research interests include AI and cybersecurity, covering AI for cognitive cybersecurity, safety and security of autonomous systems, and cognitive cybersecurity. She is an active member of IEEE's computational intelligence, RAS, cybersecurity, and women-in-engineering societies. She is the secretary of IEEE UK & Ireland's RAS Chapter and a member of the Adaptive and Dynamic Programming and Reinforcement Technic Committee (ADPRLTC) of IEEE's computational intelligence society. She is also a working group member of IEEE Technical Ethics P7000 standard. She has been an Editorial Board member of *Advances in Computing* since 2011 and an Associate Editor of *Frontiers in Blockchain* since 2018.

CARL SHAW received the Ph.D. degree in physics from Queen's University Belfast, in 1997. He then worked in U.K. defense before leaving for the private sector, where he worked in the semiconductor industry for STMicroelectronics. Throughout this period, he worked on the electronic design, system architecture, and software of embedded systems. For the last 16 years, he has been active in software and hardware security and is currently the Co-Founder of Cerberus Security Laboratories Ltd., a U.K. security consultancy, where he advises global multinationals on electronic product cybersecurity and works closely with academic institutions researching secure hardware and embedded systems.

MUHAMMAD ALI KHAN received the Ph.D. degree in condition monitoring from The University of Manchester, in 2008. He is currently a Senior Lecturer in Fatigue and Damage Tolerance. He has over 18 years' experience in failure analysis, diagnostics tools, and condition monitoring. He has worked on key defense projects sponsored by the U.S. Marines, General Dynamics, The British Army, and QinetiQ, U.K. He is also the Director of maintenance engineering and the Asset Management course with Cranfield University. He has authored a book on machine health diagnostics and published more than 60 research articles in reputed international journals and conferences. He is also a Chartered Engineer and a member of the IMechE and BINDT technical committees on condition monitoring.