machines

MDPI

Article

Security by Design for Industrial Control Systems from a Cyber–Physical System Perspective: A Systematic Mapping Study †

Ahmed Elmarkez [1,2,3], Soraya Mesli-Kesraoui [1], Pascal Berruet [2] and Flavio Oquendo [3,*]

[1] SEGULA Engineering, 165 rue de la Montagne du Salut, 56600 Lanester, France; ahmed.elmarkez@segula.fr (A.E.); soraya.kesraoui@segula.fr (S.M.-K.)
[2] Laboratoire des Sciences et Techniques de l'information de la Communication et de la Connaissance (LAB-STICC), Université Bretagne Sud, Rue de Saint-Maudé, 56100 Lorient, France; pascal.berruet@univ-ubs.fr
[3] Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Université Bretagne Sud, Rue Yves Mainguy, BP 573, 56000 Vannes, France
* Correspondence: flavio.oquendo@irisa.fr
† This paper is extended of the paper Elmarkez, A.; Kesraoui-Mesli, S.; Oquendo, F.; Berruet, P. Insights on Security-by-Design of Cyber-Physical Production Systems: A Systematic Mapping. In Proceedings of the CIGI QUALITA MOSIM 2025—Conference on Modeling, Optimisation and Simulation, Troyes, France, 8–10 July 2025.

Abstract: Industrial Control Systems (ICSs), a specialized type of Cyber–Physical System, have shifted from isolated and obscured environments to ones exposed to diverse Information Technology (IT) security threats, which are now highly interconnected. Their adoption of IT introduces vulnerabilities which they were not originally designed to handle, posing critical risks. Thus, it's imperative to integrate security measures early in CPS development, particularly during the design and implementation phases, to mitigate these vulnerabilities effectively. This study aims to identify, classify, and analyze existing research on the security-by-design paradigm for CPSs, exploring trends and defining the characteristics, advantages, limitations, and open issues of current methodologies. A systematic mapping study was conducted, selecting 55 primary studies through a rigorous protocol. The findings indicate that the majority of methodologies concentrate on the design phase, frequently overlooking other stages of development. Moreover, while there is a notable emphasis on security analysis across most primary studies, there is a notable gap in considering the integration of mitigation measures. This oversight raises concerns about the efficacy of security measures in real-world deployment scenarios. Additionally, there is a significant reliance on human intervention, highlighting the need for further development in automated security solutions. Conflicts between security requirements and other system needs are also inadequately addressed, potentially compromising overall system effectiveness. This work provides a comprehensive overview of CPS security-by-design methodologies and identifies several open issues that require further investigation, emphasizing the need for a holistic approach that includes vulnerability handling, clear security objectives, and effective conflict management, along with improved standard integration, advanced validation methods, and automated tools.

Keywords: security by design; Industrial Control Systems; Cyber–Physical System; cybersecurity; security engineering; security integration; system engineering; Industry 4.0

check for updates

Academic Editors: Andreas L. Symeonidis and Manos Tsardoulias

Received: 12 May 2025
Revised: 12 June 2025
Accepted: 17 June 2025
Published: 20 June 2025

Citation: Elmarkez, A.; Mesli-Kesraoui, S.; Berruet, P.; Oquendo, F. Security by Design for Industrial Control Systems from a Cyber–Physical System Perspective: A Systematic Mapping Study. *Machines* **2025**, *13*, 538. https://doi.org/10.3390/machines13070538

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Industry 4.0, which materialized in the 21st century, integrates physical and digital technologies, embodying the principle of Cyber–Physical Systems (CPSs). This includes technologies such as the Internet of Things (IoT), artificial intelligence (AI), and cloud computing. Through the interconnection of machinery, sensors, and data systems, Industry 4.0 facilitates enhanced efficiency, flexibility, and data-informed decision-making in manufacturing [1]. Industrial control systems (ICSs) within these recent phases have evolved from isolated systems employing proprietary components and communication protocols to systems using standardized components derived from Information Technology (IT) and exhibiting extensive inter-connectivity [2–4].

The transformation of ICSs toward more interconnected and digitized architectures—driven by the paradigms of Industry 4.0 and the growing integration of Operational Technology (OT) with Information Technology (IT)—has introduced profound cybersecurity challenges. In the effort to optimize cost efficiency and enhance system reliability, stakeholders have often emphasized the adoption of IT components, frequently underestimating the cybersecurity implications associated with such integration [5].

Acknowledging these implications, the European Union has enacted the Cyber Resilience Act [6] to strengthen the cybersecurity of all systems incorporating digital products throughout their lifecycle. This legislation requires manufacturers and system integrators to implement cybersecurity measures from the design phase, adhering to the principles of security by design [3,7,8].

Effective cybersecurity, however, extends beyond merely deploying dedicated protective components such as firewalls. Instead, it results from a combination of factors, including system architecture, component selection criteria, configuration strategies, and operational requirements [3]. Additionally, security requirements are often intertwined with and dispersed across system functional requirements [7]. Consequently, neglecting security during the early development lifecycle complicates the integration of its requirements in later phases.

Based on the above-cited issues and the great need for security-by-design methodologies, we conclude that an overview of security-by-design methodologies is required. Despite their importance, there is a notable scarcity of studies explicitly addressing security-by-design approaches for ICSs. This work seeks to fill this gap by exploring the security-by-design paradigm from the perspective of CPSs, offering a deeper understanding of its current state and potential for development.

This paper provides a systematic mapping that highlights security-by-design trends and characteristics. Moreover, it studies the relationship between security and other system requirements and how they were integrated together. We followed guidelines from [9,10] to conduct this systematic mapping. At first, we applied an automatic search on three databases. After selection, we applied a second search technique called snowballing to finally obtain 55 final primary studies.

The goal of this systematic mapping is mainly to answer these two questions:

- **RQ1**: How is the security-by-design paradigm defined?
  - *RQ1.1* What exactly are the system development lifecycle phases concerned by security-by-design methodologies?
  - *RQ1.2* Which security aspects (e.g., attacks, threats, vulnerabilities, and security controls) were focused on?
  - *RQ1.3* Which security objectives (CIA, etc.) were addressed?
  - *RQ1.4* Which other system requirements were considered while integrating security?
- **RQ2**: What are the characteristics of a security-by-design methodology?

- *RQ2.1* What are the steps that constitute each security-by-design methodology?
- *RQ2.2* Which security integration mechanism is used by the methodology?
- *RQ2.3* What element did the methodology focus on to consider security?
- *RQ2.4* What are the methodology models and how were they modeled?

This article is a revised and expanded version of a paper entitled "Insights on Security-by-Design of Cyber-Physical Production Systems: A Systematic Mapping" [11], which was accepted at CIGI QUALITA MOSIM 2025, Troyes, France, 8–10 July 2025. This preliminary version concentrated exclusively on ICSs. However, the initial study yielded limited conclusions. To address this limitation, the current article broadens the scope to encompass CPSs more generally. Moreover, while the initial study considered publications up to 2022, this article incorporates studies published in 2023 and 2024 to ensure a more comprehensive and up-to-date analysis. In addition to refining the research focus, a new research question was introduced to explore the characteristics of security-by-design studies. This represents a significant extension beyond the original work, which primarily sought to identify and define the security-by-design paradigm within existing methodological frameworks.

The remainder of this paper is organized as follows: Section 2 presents the background and related works. Then Section 3 describes the research method followed. In Section 4, the results are presented. Section 5 presents their discussion and open issues. Then, in Section 6, threats to validity are presented. Finally, we conclude in Section 7.

2. Background and Related Works

This section presents the background needed to understand this study's concepts, followed by a review of the related works.

2.1. Background

2.1.1. ICS and CPS Relationship

CPSs are sophisticated frameworks that integrate computational processes and communication networks deeply within physical environments. These systems interact dynamically with physical processes to enhance the capabilities of traditional physical systems by enabling real-time monitoring, adaptive control, and intelligent decision-making mechanisms [12].

The architecture of CPSs is generally structured into three principal layers [13]. The physical layer comprises tangible components such as sensors and actuators, which interact directly with the environment and are susceptible to both security threats and operational safety challenges. The cyber layer, in contrast, encompasses computational elements, communication protocols, and data processing units that do not directly interface with the physical world but play a crucial role in system intelligence and automation. The cyber–physical layer serves as an intermediary between these two domains, facilitating seamless interaction between digital computation and physical processes. Unlike purely cyber components, which operate in isolation from the physical domain, cyber–physical components engage directly with physical entities, ensuring synchronized operations across digital and mechanical infrastructures.

In the industrial domain, CPS concepts are extensively applied to manufacturing and production systems, and they are often termed CPPSs (Cyber–Physical Production Systems) or ICSs [14]. The evolution of traditional ICSs into CPPSs represents a paradigm shift driven by advancements in ubiquitous computing, industrial automation, and networked communication technologies. This transformation has led to highly interconnected and intelligent industrial ecosystems, where data-driven decision-making and automation optimize operational efficiency, enhance adaptability, and improve system resilience [15]. CPPS leverage edge sensor networks to continuously monitor and regulate industrial processes,

facilitating real-time responsiveness and predictive maintenance strategies. By integrating cyber capabilities into physical production environments, CPPSs enable higher levels of precision, automation, and resource efficiency. However, this increased interconnectivity also introduces new cybersecurity risks, necessitating robust security architectures to mitigate potential threats. Figure 1, containing a programmable logic controller (PLC), an actuator, a sensor, and other maintenance and supervision components, provides an illustrative example of how CPS principles are embedded within ICSs, demonstrating the interplay between computational intelligence and industrial physical processes.

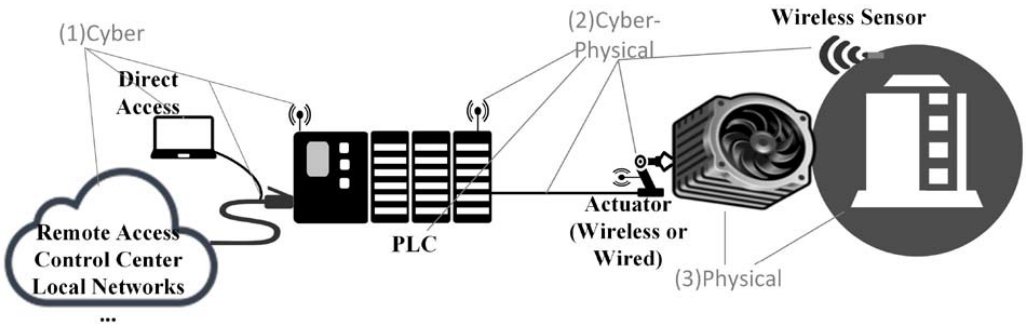


Figure 1. CPS aspects in ICSs [13].

2.1.2. System Requirements

Throughout their development lifecycle, CPSs must meet various requirements to ensure the effective operation of complex systems while also fulfilling customer needs [16]. According to the existing literature, these requirements can be categorized into four distinct groups [17]:

- **Functional requirement:** It encompasses the functions/components of the system and their interactions, which are necessary to provide the intended functionality [18].
- **Performance requirement:** It relates to performance considerations and encompasses constraints on factors such as timing, processing or reaction speed, data volume, and throughput [17].
- **Specific quality requirement:** It consists of specific qualities that the system or a component shall have like security, safety, and reliability [17].
- **Constraint:** It is a requirement that constrains the solution space beyond what is necessary for meeting the given functional, performance, and specific quality requirements [17].

In the remainder of this article, these will be referred to as "other system requirements", denoting system requirements as distinct from security-related ones.

2.1.3. Terminology

Different definitions of essential technical concepts are presented below:

- Attack: "An attempt to gain unauthorized access to system services, resources, or information, or an attempt to compromise system integrity, availability, or confidentiality" [2].
- Threat: "Any circumstance or event with the potential to adversely impact agency operations (including mission, functions, image, or reputation), agency assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service" [2].
- Mitigation: "A decision, action, or practice intended to reduce the level of risk associated with one or more threat events, threat scenarios, or vulnerabilities" [19].
- Weakness: "Defect or characteristic that may lead to undesirable behavior" [20].

- Vulnerability: "Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source" [2].
- Security objectives: These mainly consists of confidentiality, integrity, and availability (CIA) [21]. Other cybersecurity concepts are considered as security objectives such as authentication and non-repudiation. They are mentioned in this systematic mapping as "other security objectives".
- Security aspects: "security threats, vulnerabilities, attacks, and security solutions as different aspects to be considered while engineering security" [7].
- Model: "a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process" [22].
- Modeling language: "A language in which a model is expressed. It includes syntax, rules, and its semantics" [23].
- Security by design is a security approach defined in reference [3] as the sufficiently early integration of security decisions into an established engineering workflow (or base workflow) which ensures that crucial design decisions remain open to influences.

2.2. Related Works

This section reviews existing work on security-by-design methodologies. Table 1 provides a comparative summary highlighting the distinctions between the present study and prior approaches. A detailed analysis of these differences is presented in the following discussion.

Table 1. Comparison of related work.

Related Work	Focus on Security by Design	Consideration of Other Requirements	Focus on CPS
[3]	✓		
[24]	✓		
[8]	✓		✓
[7]	✓		✓
[25]		✓	✓
[13]			✓
[26]			✓
[27]		✓	
[28]	✓		✓
[29]	✓	✓	
[30]			✓
[31]	✓		
This work	✓	✓	✓

The authors of reference [3] present some existing security-by-design approaches for ICSs and group them according to how they integrate security into existing engineering workflows. The main drawback of [3] is that it is not a systematic study. In addition, the number of compared security by design approaches is limited. Also, it compares the chosen study based on just one metric (the security integration mechanism). Moreover, the authors in [3] only consider ICS studies.

The authors in reference [24] conducted a systematic literature review where they provided a detailed analysis of the state of the art in model-driven security. Yet, their study does not focus just on CPSs (only 6% of the studies they covered are from the CPS domain) and is specific to model-driven approaches.

In reference [8], Geismann and Bodden present a systematic literature review of model-driven security engineering for CPSs. They found seven approaches specifically developed for CPSs. They provide a detailed description of these approaches by expressing each one's limitations. However, the study is a systematic literature review and is specific to model-driven methodologies. Moreover, they only consider papers that integrate security into all phases of the system development lifecycle in contrast to our study, which focuses on the design and implementation phases.

The authors in reference [7] present a systematic mapping of model-based security engineering approaches for CPSs. Their study focuses on finding out the statistics of model-based security engineering approaches for CPSs, their characteristics, and the open issues in this research area. However, this study focuses on model-based engineering approaches and does not consider whether the primary studies had taken into account the other system requirements. The paper by [25] is a systematic mapping study that aims at identifying, classifying, and analyzing existing research on CPS security in order to better understand how security is addressed when dealing with CPS. However, it considers security in CPSs in general and does not focus in particular on security-by-design approaches and their specific concepts. It is important to note that the representation of our paper is influenced by this article, but the results differ in terms of the study objectives.

In their surveys [13], Humayed et al. studied existing research on CPS from three perspectives: the security perspective, the CPS component perspective, and the CPS perspective. Yet, it is not a systematic study. Moreover, it focuses neither on security by design nor on other system requirements considered by the primary studies.

Uzunov et al. surveyed in reference [26] a large set of methodologies for security in distributed systems. They consider different types of security methodologies such as model-driven methodologies, architecture-driven methodologies, pattern-driven methodologies, and agent-driven methodologies. Nevertheless, their review does not specifically address studies that examine other system requirements considered by the security-by-design methodologies.

In their survey [27], the authors provide a comparative analysis of the existing approaches to industrial facility design and risk assessments that consider safety and security. Yet, their study focuses on risk assessments. In addition, they only consider approaches that combine safety and security.

In their work [28], Huang et al. conduct a systematic literature review that explores existing security modeling methods, specifically examining how these methods address real-world cybersecurity threats and attacker capabilities unique to CPSs throughout their lifecycle. However, their study is narrowly focused on CPS security modeling, including both threat and attack modeling. In contrast, our research adopts a broader perspective, concentrating on the full spectrum of CPS security-by-design methodologies, rather than limiting the scope to modeling approaches alone. This allows for a more comprehensive investigation of how security is integrated into the early stages of CPS development phases.

Mashkoor et al. in reference [29] conducted a systematic mapping study focusing on the model-driven engineering of safety and security concerns in software systems. However, their research is limited in scope, as it is exclusively centered on software engineering and considers only studies that consider safety and security simultaneously.

In their study [30], Harkat et al. conducted a systematic review on the security of CPSs. Their work focuses on identifying CPS security threats, vulnerabilities, attacks, threat

detection approaches, and security assessment mechanisms. Additionally, they highlight the ethical issues and societal impacts associated with CPS security. However, unlike our study, their review addresses CPS security in a general context, without specifically concentrating on security-by-design methodologies or analyzing these methodologies in conjunction with other system requirements.

Del-Real et al. provide a comprehensive review in reference [31], addressing the key definitions and conceptual distinctions between security by design and privacy by design within the context of software systems. Unlike our study, which delves into the detailed steps and specific characteristics of these methodologies, their systematic literature review concentrates on comparing the foundational definitions of security by design and privacy by design, without extending the analysis to the procedural and operational aspects of these frameworks. This narrower focus highlights conceptual contrasts rather than practical implementations or methodological attributes.

Our study stands out from previous research by comprehensively focusing on the security-by-design paradigm and examining various other system requirements and their integration with security measures. Also, we consider all types of methodologies regarding techniques used to integrate security. Moreover, our research does not focus only on a specific system layer; it considers all three layers of CPSs.

3. Research Method

In recent years, the number of mapping studies has been continuously increasing, and there is a great interest in the methodology [10]. A systematic mapping study serves as a research methodology designed to impartially, objectively, and systematically address a series of research questions. Its primary aim is to comprehensively uncover all pertinent research findings within a specific research domain, offering a structured and thorough approach to analyzing the available information [25].

We followed guidelines from references [9,10] to conduct this systematic mapping. There are three main phases in a systematic mapping review:

1. **Planning**: During this phase, our main objective is to define the protocol to follow. Firstly, the research questions were identified (Section 3.1) after analyzing the needs for this systematic mapping and extracting other systematic studies' limitations according to our objectives. Secondly, a protocol describing in detail the steps we must follow while conducting the study is defined. After, a data extraction scheme is created. This defines the various data we should extract in order to answer our research questions. Finally, the protocol and the data extraction scheme are reviewed several times until they are approved by all the participants before moving to the second phase.
2. **Conducting**: During this phase, we put into practice the defined protocol. First, we began by searching for a set of candidate papers on security by design. During this step, we used an automatic search in three databases (IEEE explore, ACM Digital Library, and ScienceDirect). Another technique used for searching for candidate papers is snowballing. It is used after the selection of the first set of primary studies. Second, the candidate papers are then filtered to obtain the final list of primary studies that correspond to and answer our research questions. Third, we extract data according to the extraction scheme defined in the first phase. Finally, we synthesize the data collected and analyze them using different types of graphics. This activity elaborates on the information extracted in order to answer the research questions one by one.
3. **Reporting**: The final phase corresponds to writing up the results and disseminating the study to potentially interested parties. This report elaborates on the main findings

but also discusses the threats to validity. In the end, this report is written and reviewed for publication.

3.1. Research Questions

This section details the research questions raised in Section 1. The first concern of this systematic mapping is to extract from the primary studies the elements that define a security-by-design methodology. For that, **RQ1** is divided into four sub-questions. First, as previously stated, security by design means the integration of security into the early stages of the systems development lifecycle. Hence, as asked in ***RQ1.1***, *what exactly are the system development lifecycle phases concerned by security-by-design methodologies?* Answering this question allows us to refine the definition of a security-by-design methodology to which system development lifecycle phases are based on. Second, each security-by-design approach focuses on a combination of security aspects to analyze the security of a given system design. These security aspects could be attacks/threats, vulnerabilities, or security controls. So, as mentioned in ***RQ1.2***, *which security aspects (e.g., attacks, threats, vulnerabilities, and security controls) were focused on?* The answers to these questions will help us discover the common ways and the basis of system security analysis used for security-by-design methodologies. Third, given their nature, the security objectives of CPSs differ from those of IT systems. To find out which security objectives (confidentiality, integrity, and availability) are prioritized during the integration of security into the design of CPSs, we aim to answer ***RQ1.3***; that is, which security objectives (CIA, etc.) were addressed? Fourth, a security-by-design methodology must not ignore the effect of security on other functional and non-functional requirements. For this, the question (***RQ1.4:*** *Which other system requirements were considered while integrating security?*)will help us discover which system requirement type has been preserved during the integration of security.

Our second aim from this systematic mapping (RQ2) is to find out what characterizes security-by-design methodologies. To be more specific, this research question is divided into four sub-questions.The first sub-question is ***RQ2.1:*** *what are the steps that constitute each security-by-design methodology?* The answer will provide us with the main common steps that constitute a security-by-design methodology. The second question (***RQ2.2:*** *Which security integration mechanism is used by the methodology?*) aims at extracting the security integration mechanism used by the methodology. The answer will allow us to determine how security was integrated into various base development workflows and to what degree the security integration has influenced design decisions. The third question (***RQ2.3:*** *What element did the methodology focus on to consider security?*) considers if the methodologies have focused on the system, the assets, or the attacker when considering security. The fourth question (***RQ2.4:*** *What are the methodology models and how were they modeled?*) considers mostly model-based methodologies. This question provides us with different models and languages used by different papers.

3.2. Exclusion Criteria

The following exclusion criteria (EC) are used to exclude papers during the paper selection process.

- (EC1): Unrelated domain studies are excluded. (They must concern the CPS domain and its sub-domains.)
- (EC2): Studies not focusing on a "security-by-design" methodology are excluded. This means only studies that integrate security in the system development lifecycle are considered.
- (EC3): Studies not focusing on the design or implementation phases are excluded. This means that papers not considering at least one of these phases are not included.

- (EC4): The old versions of studies are excluded. For instance, we excluded certain workshop and conference papers when their extended versions were published in academic journals.
- (EC5): Studies before 2012 or after 2024 are excluded.
- (EC6): Papers less than 4 pages long are excluded.
- (EC7): Secondary and tertiary studies are excluded.
- (EC8): Non-peer-viewed papers are excluded.
- (EC9): Non-English papers are excluded.

A quality assessment was not employed in the selection process. As noted in reference [10], "quality assessment is more essential in systematic reviews to determine the rigor and relevance of the primary studies. In contrast, no quality assessment is required for systematic maps".

3.3. Search Strategy

The systematic mapping study aims to find as many primary studies relating to the research questions as possible using an unbiased search strategy [9]. An automatic search is the most common way of finding primary studies according to [10]. We use three online databases (IEEE explore, ACM Digital Library, and ScienceDirect). Moreover, to achieve maximal coverage, we used another search technique which is snowballing (Section 3.3.5) [32]. We employed Parsifal (https://parsif.al/), an online tool designed to facilitate the management of candidate papers throughout the selection process. Therefore, our search strategy is composed of 5 main steps, as shown in Figure 2:

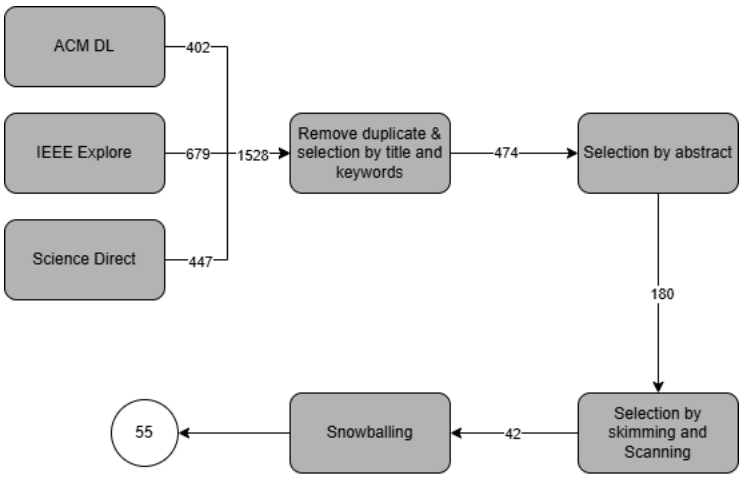


Figure 2. Selection strategy.

3.3.1. Step 1: Automatic Search

During this step, we use an automated search in online databases using a search string. (A total of 1528 papers were identified.) This search string was constructed using the PICOC (population, intervention, comparison, outcomes, and context) concept [9]. This involves identifying search terms and organizing them into groups (Table 2) based on the following factors:

- The population in which the evidence is collected i.e., which groups of system types are of interest for the review. Our review concentrates on CPSs. We derived other sub-types of systems from CPSs such as the Internet of Things and ICSs.
- The intervention applied in the empirical study, i.e., which technology, tool, or procedure is under study. In our case, it corresponds to security by design.
- The comparison to which the intervention is compared to. Due to the exploratory study, the comparison is not relevant, so it was excluded.

- The outcomes represent the goal of the paper. In our case, we search for methodologies. This is not considered in the search string but rather in the exclusion criteria.

Table 2. Search string keywords (Population AND Intervention).

PICO	Keywords
Population	"Abstract":cps OR "Abstract":"cyber-physical system" OR "Abstract": "CPS" OR "Abstract": ics OR "Abstract": "industrial control systems" OR "Abstract": "industrial control system" OR "Abstract": IoT OR "Abstract": "internet of things" OR "Abstract": "digital twin" OR "Abstract":"device shadow" OR "Abstract":"autonomous vehicle" OR "Abstract": "smart grid" OR "Abstract": scada OR "Abstract": "embedded systems" OR "Abstract": "power grid" OR "Abstract": "smart car" OR "Abstract":"aircraft system" OR "Abstract": "healthcare system" OR "Abstract":"embedded system" OR "Abstract":"industry 4.0"
Intervention	"Abstract":"security by design" OR "Abstract": "secure by design" OR "Abstract": "secure by construction" OR "Abstract":"security by construction" OR "Abstract":"security engineering" OR "Abstract": "secure engineering" OR "Abstract": "secure development life-cycle" OR "Abstract":"security development lifecycle" OR "Abstract": "engineering security" OR "Abstract":"secure design" OR "Abstract":"security-aware" OR ((("Abstract":"model-based") OR ("Abstract":"model-driven")) AND "Abstract":security)

3.3.2. Step 2: Removing Duplicate Papers and Selection by Title and Keyword

As shown in Figure 2, after removing duplicate papers, we first filter out the candidate papers using their title and the specified keywords according to the exclusion criteria (Section 3.2). Four hundred seventy-four papers were selected after this step.

3.3.3. Step 3: Selection by Abstract

A second selection is performed by reading just the abstract and deciding whether the paper in question should be excluded or not according to the exclusion criteria. One hundred eighty papers were selected after this step.

3.3.4. Step 4: Selection by Skimming and Scanning

This third selection corresponds to skimming and scanning the candidate papers and then applying the exclusion criteria to include or exclude them. At the end of this step, we had a set of primary studies. Forty-two papers were selected after this step.

3.3.5. Step 5: Snowballing

This search strategy corresponds to the examination of references (backward snowballing) and citations (forward snowballing) of each primary paper of the set of papers obtained in Step 4. The examination consists of repeating Steps 2, 3, 4, and 5 for each referenced or cited paper. We found 13 more primary studies from this process (55 final papers).

3.4. Data Extraction

The extraction of the important data from the primary studies is performed using a data extraction scheme. (The concepts extracted are presented in Figure 3.) Note that this tree of concepts was defined before we conducted this systematic study. This foundational structure provided a guiding context for the research, ensuring that the subsequent analysis was informed by clearly defined categories and relationships among the concepts. The data extracted aims to answer different sub-questions defined in Section 3.1. The scheme contains

three main parts. Firstly, the documentation part consists of regrouping documentation-related information such as the domain, the venue, the type of venue, and the context (academic or industrial). Secondly, the answer to RQ1 considers identifying elements that define the methodology and answer each of its sub-question. Thirdly, to answer RQ2, we examine each of its sub-questions to extract the elements that characterize one type of methodology from the others.

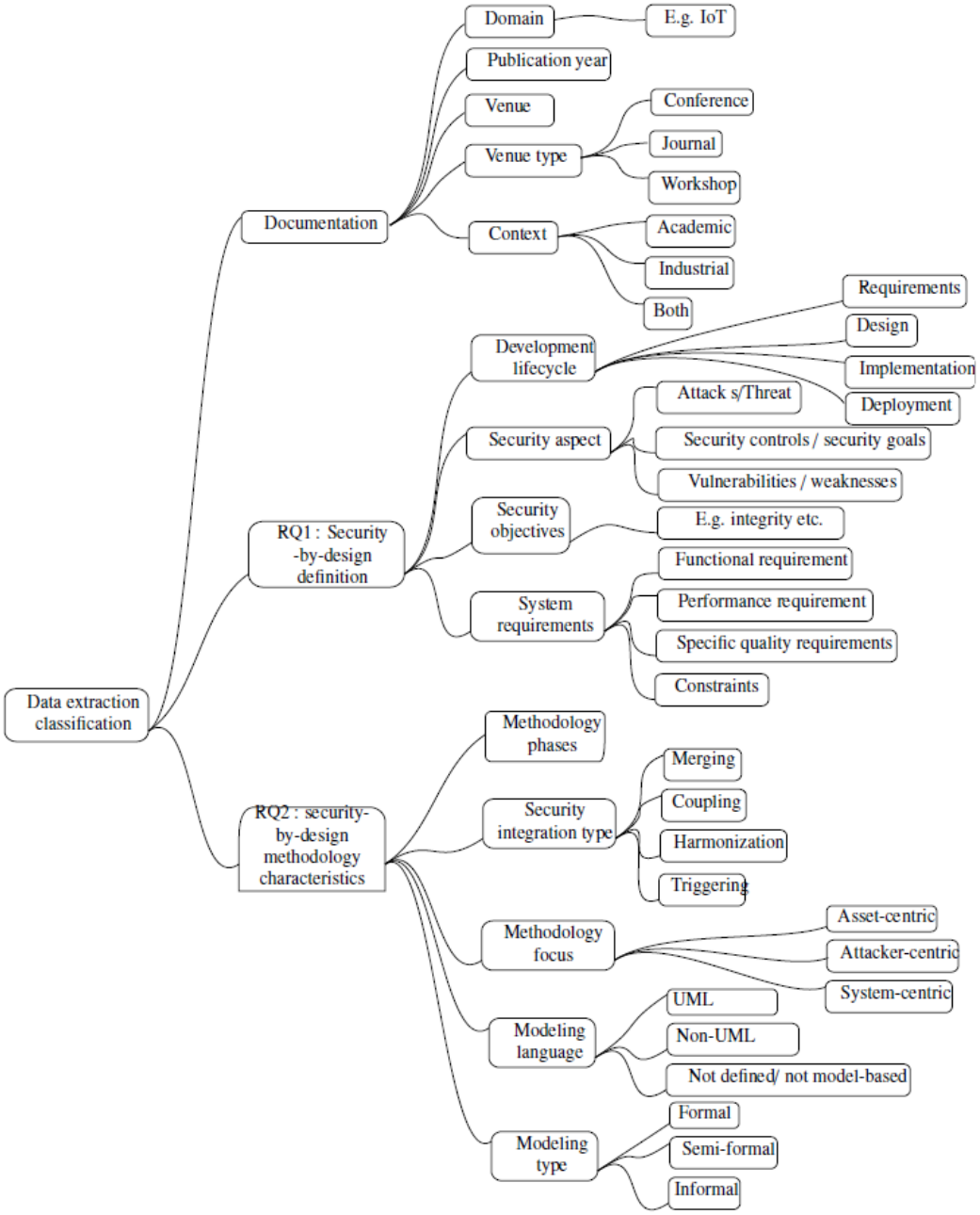


Figure 3. Data extraction scheme.

4. Results

The first author extracted data according to the extraction scheme using Microsoft Excel. Several revisions of the extracted data were made by the other authors. Afterward, the extracted data were synthesized. The results are presented in this section. The extracted data can be found in reference [33]. Primary studies are presented in Appendix A.

4.1. Documentation

This section provides results related to documentation and publication trends.

4.1.1. Domain

Figure 4 presents the distribution of papers according to the domain. The first interesting result of our study is that approximately 36% of primary studies consider CPSs in general. The figure also gives us a closer look at the specific domains where the integration of security is taken into account during the design phase. We observe that embedded systems, IoT, automotive, and ICSs are the domains that consider security during the early phases of the development lifecycle. On the other hand, there are only a few studies in rail systems and smart grid domains.

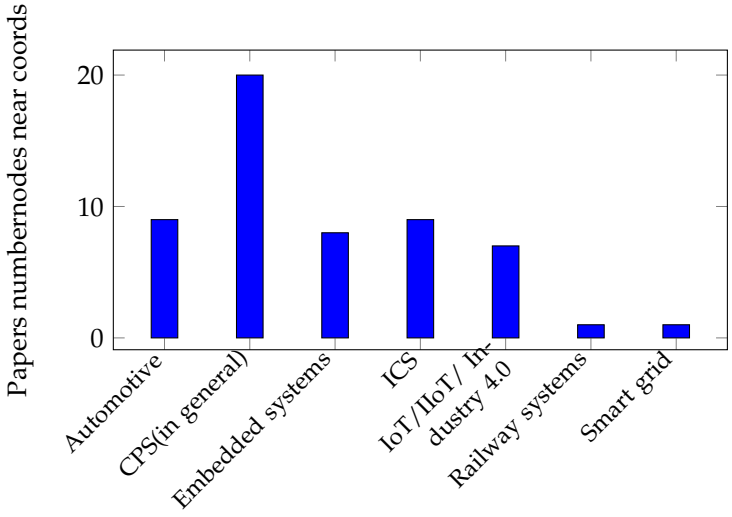


Figure 4. Distribution of papers per domain.

4.1.2. Publication Year

Figure 5 shows that there has been an increasing interest in integrating security in the early phases of the lifecycle over the last six years. Indeed, 56% of studies were published in the period between 2019 and 2024 (the last four years). Also, we notice that 19 out of the 55 primary studies (34.5%) were published between 2019 and 2021. This sharp increase in interest might be explained by the increase in cybercrime against different kinds of systems during this period of time [34].

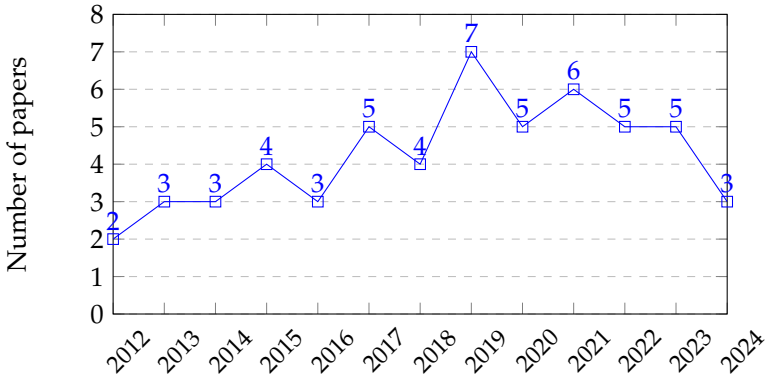


Figure 5. Distribution of papers per Year.

4.1.3. Venue and Type

Figure 6 illustrates the distribution of papers based on the venue type. The data indicates that more primary studies have been published in conferences proceedings (32 papers) compared to journals (17 papers) and workshops (6 papers). This trend may reflect the emerging nature of this research area, where conferences facilitate more effective exchange of ideas, while the field may not yet be sufficiently developed for extensive journal publication.

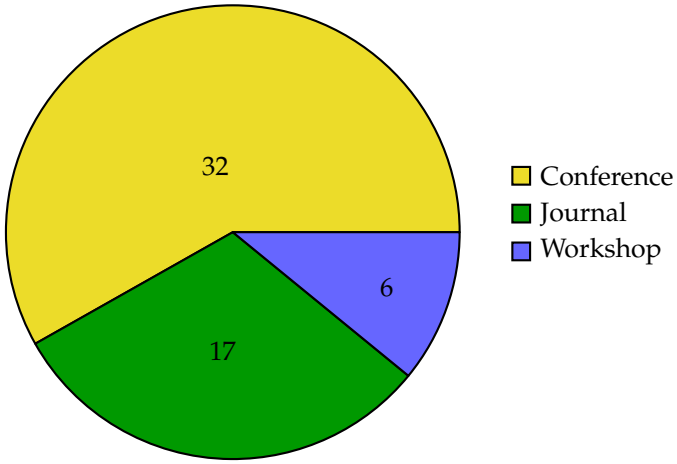


Figure 6. Distribution of papers according to their venue type.

Table 3 presents the publication venue with more than one primary study. We notice that there are only five venues with more than one primary study, and the maximum number of publications in one venue is three papers. (This is the case for IEEE Access.) On the other hand, 36 venues had one published paper. In addition, the nature of the venues is heterogeneous and encompasses different areas of research. This can be explained by the multidisciplinary nature of CPSs. Also, we observe that publication venues are not specifically related to cybersecurity. This may be an explanation of the lack of venues fully dedicated to CPS security and also explains the low number of papers published in workshops as shown in Figure 6.

Table 3. Publication venue with more than one selected study.

Venue	Number of Studies	Type
IEEE Access	3	Journal
ACM/EDAC/IEEE Design Automation Conference (DAC)	2	Conference
IEEE International Symposium on Software Reliability Engineering Workshops	2	Workshop
IEEE Transactions on Industrial Informatics	2	Journal
International Conference on Cyber-Technologies and Cyber-Systems	2	Conference

Concerning the distribution of papers exploring security by design per domain according to the venue type (Figure 7), we notice that there are more workshops (three) for the IoT domain than the others.

4.1.4. Context

The context of each study was established based on the affiliations of its authors. We see in Figure 8 that 40 papers (approximately 72.7%) are academic, 13 papers are from a partnership between the academic and industrial domains, and only 2 papers are from the industrial domain. Only a quarter of the primary studies approximately have the involvement of the industrial domain. In Figure 9, we present the distribution of papers per domain according to their context. We can notice the involvement of the industry in the majority of domains (except for the smart grid domain). While this presence justifies the necessity, it seems to be constrained or limited in some way. The core reasons for neglecting security by design in the industrial context—such as the intense pressure for rapid product development and market deployment, immediate cost-saving priorities, constrained resources for specialized security expertise, and organizational cultures that treat security as an afterthought rather than an intrinsic design element—also often explain their limited capacity or willingness to invest in longer-term research collaborations.

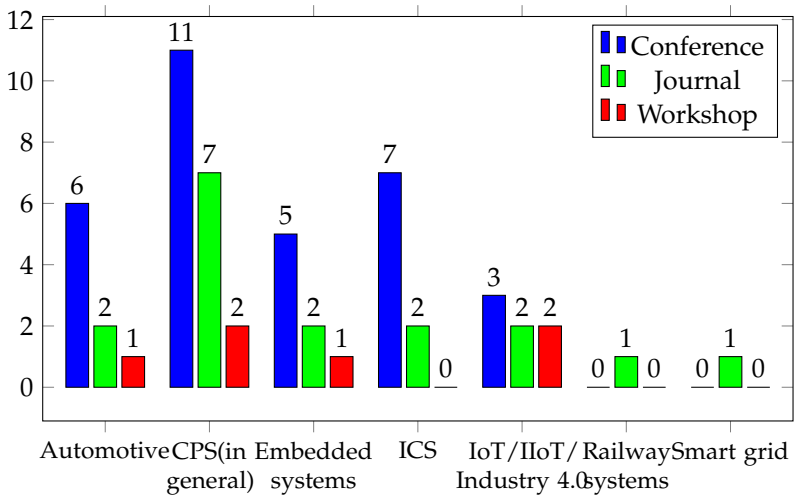


Figure 7. Distribution of papers per domain according to the venue type.

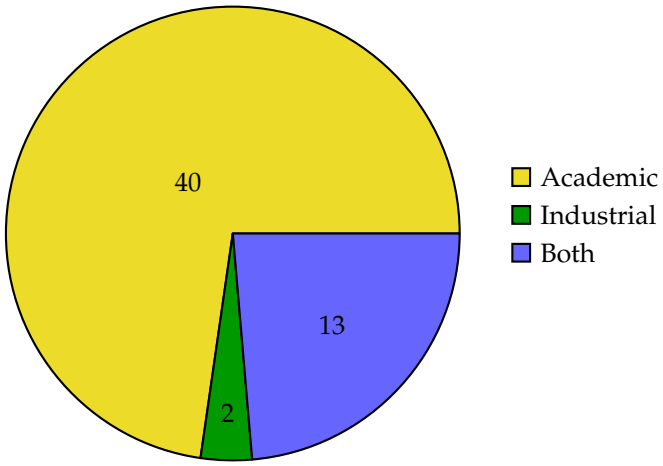


Figure 8. Distribution of papers according to their context.

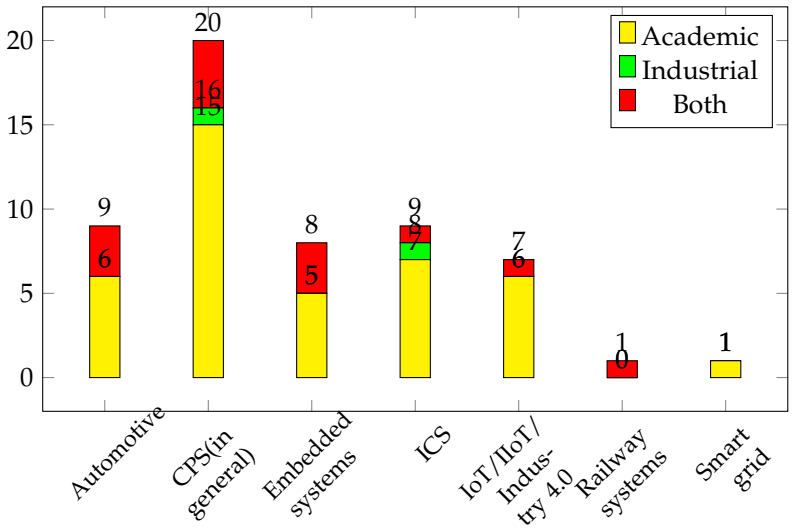


Figure 9. Distribution of papers per domain according to their context.

As we can see in Figure 10, the involvement of the industry has increased in the last 4 years. In fact, it indicates that at least one publication involves industrial authors. This trend appears to correlate with the rise in cybercrimes during this period, highlighting the current limitations of existing security methodologies. It underscores the necessity of integrating security considerations early in the design phase as a cohesive and cost-effective approach for developing security-critical CPSs [35].

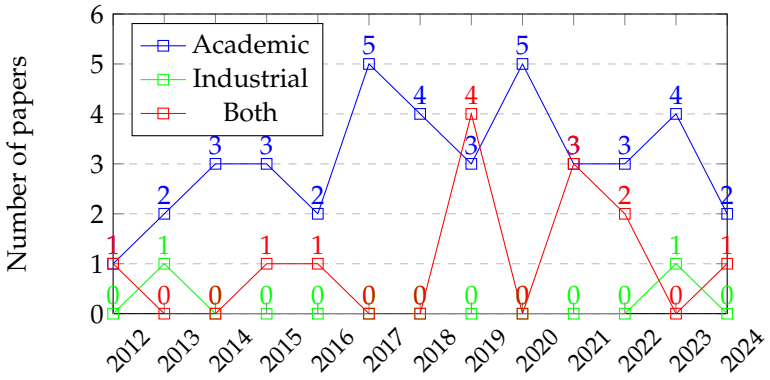


Figure 10. Distribution of papers per year according to their context.

4.2. RQ1: Towards a Security-by-Design Definition

This section provides the results that answer RQ1 and its sub-questions.

4.2.1. RQ1.1—What Exactly Are the System Development Lifecycle Phases Concerned by Security-by-Design Methodologies?

As already explained in the exclusion criteria, we focus our study on methodologies that principally consider the design or implementation phases. Consequently, as shown in Figure 11, all studies consider at least one of these. We also observe in Figure 11 that all studies consider the design phase. Eight studies (approximately 14.5%) consider the whole development lifecycle from the requirement phase to the deployment phase, such as S9, S16, and S27.

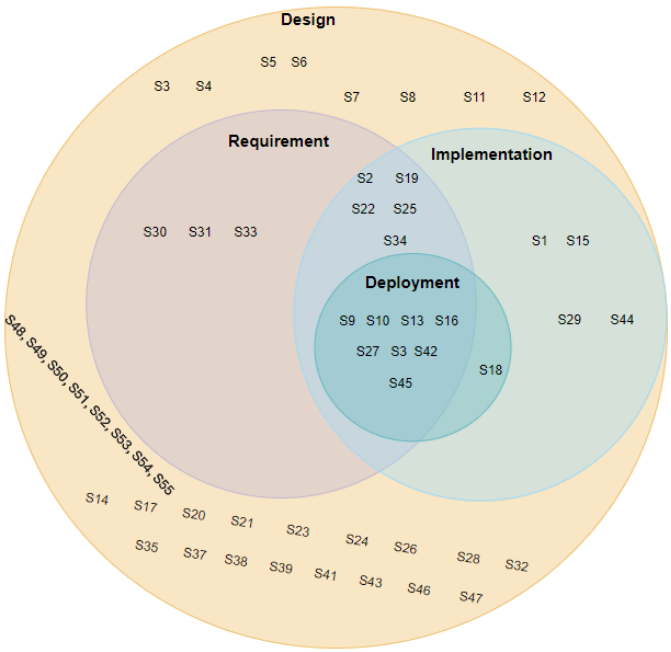


Figure 11. Distribution of papers according to development lifecycle phases.

4.2.2. RQ1.2—Which Security Aspects (e.g., Attacks, Threats, Vulnerabilities, and Security Controls) Were Focused on?

Figure 12 shows the distribution of papers according to security aspects. We observe that 13 papers use "vulnerabilities/weaknesses" or "security controls/security goals" in comparison to "attacks/threats" (43, approximately 78%). A total of 11 studies jointly use "vulnerabilities/weaknesses" and "attacks/threats" in security analysis. For instance, the authors of S19, S20, and S21 first identified vulnerabilities of the system and then extracted threats/attacks related to these vulnerabilities in order to specify security requirements. Specifically, S23 uses security threats to identify corresponding security mitigation strategies during the design stage and then uses vulnerabilities from the CVE database during the simulation phase to identify more attacks/threats.

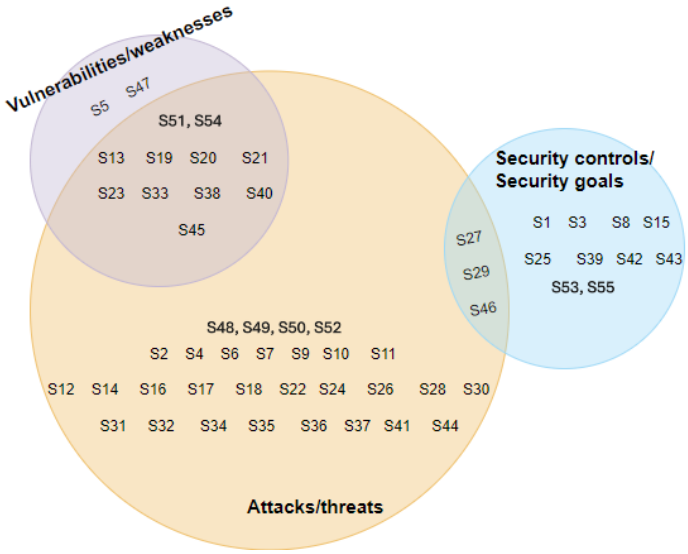


Figure 12. Distribution of papers according to security aspects.

4.2.3. RQ1.3—Which Security Objectives (CIA, etc.) Were Addressed?

As shown in Figure 13, nearly 56% of primary studies consider security implicitly. That means they do not specify explicitly what security objectives they try to accomplish

when protecting systems. For example, the authors of S14 and S16 use attack vectors to determine security controls without specifying which security objectives they intend to reinforce when addressing system security.

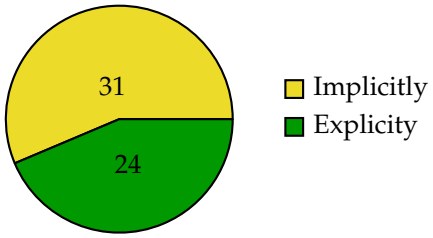


Figure 13. How were security properties addressed in the primary studies?

We count the number of articles that had specifically addressed each security objective (Table 4) in Figure 14. We notice that integrity is the most considered factor in the primary studies followed by confidentiality. Other security objectives are addressed such as authentication (for example, S31) and non-repudiation (for example, S19).

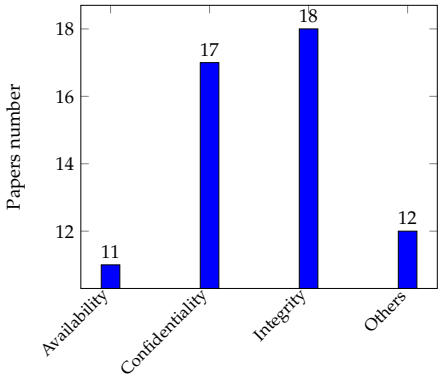


Figure 14. The security objectives addressed in the primary studies.

Table 4. Studies explicitly considering each security objective.

Security Objective	Study ID
Availability	S3, S4, S6, S8, S11, S26, S36, S38, S42, S43, and S45
Confidentiality	S1, S3, S7, S8, S11, S15, S19, S26, S29, S30, S31, S34, S36, S41, S42, S44, and S45
Integrity	S3, S6, S7, S8, S15, S17, S19, S26, S29, S30, S31, S34, S36, S38, S41, S42, S43, and S45
Other	S1, S3, S4, S11, S17, S19, S25, S29, S30, S31, S42, and S43

4.2.4. RQ1.4—Which Other System Requirements Were Considered While Integrating Security?

The integration of security produces some conflict with other system requirements [36]. Figure 15 shows the number of studies that took into account each system's requirement type (Table 5). We can see that 26 papers do not tackle conflicts that may arise when integrating security with other system requirements. Also, we can notice that a significant number of studies have considered specific quality requirements mainly safety (for example, S9, S27, and S30). As illustrated in Figure 15, other system requirements are also considered such as performance (S12, S17, S29, and S30), resource limitation/usage (S17 and S29), functional requirements (S23), and cost (S7 and S39).

Figure 16 illustrates the distribution of papers across various domains concerning system requirements. One significant observation is that, in the context of embedded systems, the emphasis on performance—particularly real-time performance—tends to be predominant when integrating security. This focus is followed by considerations of resource constraints, which are inherent to these systems. In contrast, the automotive sector, ICSs, and CPSs generally prioritize safety, reflecting their classification as safety-critical systems. Additionally, functional requirements are recognized in the domains of IoT, ICSs, automotive, and CPSs. Conversely, the remaining domains, such as railways and smart grids, do not appear to address other system requirements when dealing with security.

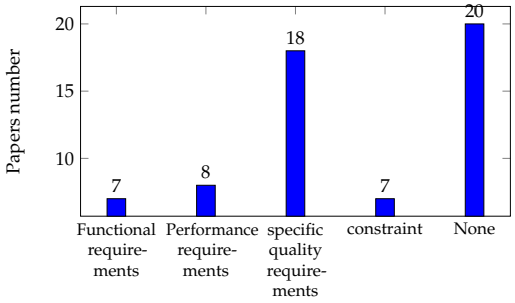


Figure 15. The system requirements that were considered by the primary studies.

Table 5. Studies explicitly considering other system requirements.

System Requirement	Study ID
Functional requirements	S5, S13, S23, S33, S36, S38, and S43
Performance requirements	S1, S7, S12, S17, S25, S29, S30, and S33
Specific quality requirements	S4, S5, S9, S10, S14, S16, S18, S24, S25, S26, S30, S31, S36, S37, S39, S43, S52, and S53
Constraints	S7, S13, S17, S29, S39, S42, and S44

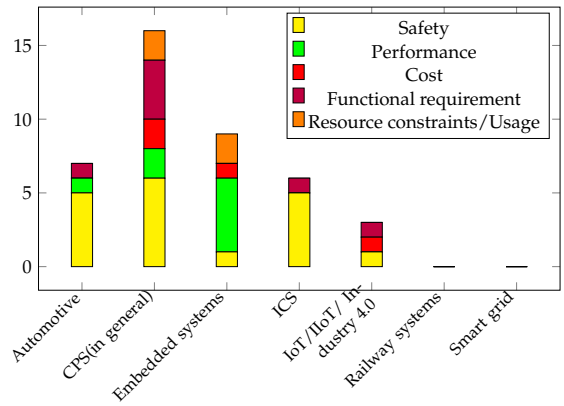


Figure 16. Distribution of papers per domain according to system requirements.

4.3. RQ2: Security-by-Design Methodology Characteristics

4.3.1. RQ2.1—What Are the Steps That Constitute Each Security-by-Design Methodology?

Table 6 presents various methodology steps extracted from the primary studies. In the requirement phase, most methodologies focus on establishing security requirements. Security requirements are either the results of the risk assessment or serve as entry points to identify security properties. Some studies also consider establishing specific quality requirements, especially safety requirements by identifying hazards and defining safety requirements.

Table 6. Security-by-design methodology steps.

Development Phases	Methodology Steps	Sub-Steps	Frequency	Article IDs
Requirements	System requirement establishment	Formalization of the uses needed	1	S33
		System domain analysis	1	S2
	Specific quality requirement establishment	Risk assessment	5	S9, S10, S19, S22, and S30
		Security requirement establishment/ modeling	14	S2, S9, S10, S13, S16, S19, S25, S27, S31, S34, S36, S38, S39, S40, and S45
		Security requirement prioritization	2	S19 and S36
		Anti-requirements/misuse case definition/ modeling	1	S22
		Hazard identification	3	S9, S10, and S30
		Definition of safety requirements	3	S10, S16, and S39
Design	System modeling	Function modeling	5	S3, S12, S30, S35, and S38
		Data/control flow	4	S5, S46, S47, and S50
		System components	21	S6, S8, S11, S21, S23, S25,S26, S28, S34, S35, S39, S41, S43, S45, S48, S49, S51, S52, S52, S54, and S55
		Use cases	1	S14
		Task/Resource matrix	1	S17
		System state	1	S27
		System of system	2	S1 and S32
		Building blocks	1	S33
		Black box	1	S37
		Missions	1	S43
		Task	2	S7 and S29
	Design security Analysis	System model annotation (with missing security properties/requirements)	4	S1, S3, S15, and S47
		Attack/threat analysis	19	S2, S7 , S10, S12, S13, S14, S16, S17, S23, S29, S32, S38, S41, S44, S48, S49, S50, S52, and S54
		Security property/goal/objective analysis	12	S2, S3, S6, S7, S8, S19, S23, S25, S42, S43, S53, and S55
		Security flaw/weakness/ vulnerability analysis	5	S8, S21, S38, S47, and S51

Table 6. *Cont.*

Development Phases	Methodology Steps	Sub-Steps	Frequency	Article IDs
Design		Zone and conduit analysis	3	S10, S27, and S35
		Relevant system task/asset analysis	10	S3, S18, S20, S23, S24, S29, S40, S41, S42, and S44
		Security concept modeling	11	S2, S6, S11, S13, S22, S23, S26, S28, S33, S40, and S41
		Risk estimation	2	S20 and S31
		Attack path/threat prioritization	1	S23
		Security requirement identification/ analysis	6	S5, S7, S12, S17, S20, and S53
		Attack impact analysis	1	S17
	Specific quality requirements concept analysis	System hazard analysis	3	S5, S26, and S37
		Identification of security risks/threats causing system hazards	3	S4, S24, and S26
		Safety property modeling	1	S14
		Safety mitigation definition/integration	2	S10 and S30
		Harmonization/ consolidation/alignment of safety and security and/or performance	5	S4, S9, S10, S14, and S30
	Constraint analysis		3	S13, S19, S42, and S52
	Performance analysis	Design space exploration according to performance and security	4	S7, S12, S29, and S30
	Secure system synthesis	Security mitigation selection and/or integration	27	S1, S3, S6, S10, S13, S16, S17, S18, S22, S23, S25, S26, S27, S30, S31, S32, S34, S36, S38, S40, S41, S42, S44, S45, S46, S53, and S55
		Constraint satisfaction problem	1	S39
		Security policy definition	2	S22 and S31
		Security mitigation selection and its impact analysis	1	S19
		Model checking	1	S28
		Virtual prototyping	1	S45

Table 6. *Cont.*

Development Phases	Methodology Steps	Sub-Steps	Frequency	Article IDs
Design	Design verification	Verification of security policies	1	S22
		Verification that the system is protected against a type and level of attack	1	S33
		Analysis of security and safety properties using theorem proving	1	S43
		Qualitavive analysis	1	S38
		Quantitative analysis	2	S11 and S38
	Planning for implementation and deployment	Planning of operation and maintenance	1	S36
		Planning of safety and security validation	1	S36
		Planning of installation and commissioning	1	S36
		Cybersecurity assurance planning	1	S36
Implementation	Secure development	Security mechanisms implementation	3	S19, S22, and S34
		Secure/safe hardware/software development	5	S9, S16, S27, S36, and S45
	Security testing	Unit/static/dynamic testing	4	S2, S13, S27, and S36
		Verification of security requirements	1	S19
		Simulation/ prototyping	2	S23 and S25
		Vulnerability scanning/fuzzing/ penetration testing	1	S36
	Code generation	Security algo-rithms/components	2	S13 and S15
		Secure code automatic generation	2	S15 and S30
	Hardware/platform implementation	Hardware threat specification	1	S24
		Security artifact generation	1	S24
		Platform artifact specification	1	S24
Deployment	Safety and security monitoring and periodic assessment		4	S9, S10, S13, and S18
	Incident response		3	S18, S27, and S36
	Security management		1	S27

In the design phase, most methodologies start by modeling the system. Various ways are used to do this, with modeling the system in terms of components being the most popular one. The system is modeled according to its different components and their interactions. The function, tasks, missions, and subsystems are some other ways used to model the system when the methodologies are model-based. Security and other system requirements are then analyzed. Attacks/threats, security goals/properties/objectives, security flaws/weaknesses/vulnerabilities, or a combination of these are used to analyze security. In many studies, these elements are modeled (see Section 4.3.4). They serve to identify missing security requirements or properties (for instance, S1, S3, and S7), identify zones and conduits (as in S10 and S27), estimate risks and prioritize them (for example, S20 and S23), or analyze the impact of the risks (such as S17). Safety, performance, and constraints such as resource limitations or cost are analyzed together with security in many studies. The identification of safety hazards is often the first step in the process of safety analysis. Studies such as S4, S24, and S26 use security risks to identify safety hazards. When dealing with safety, studies often tend to either harmonize it with security requirements (for example, S9) or consolidate it with security decisions (for example, S10). Performance is usually taken into account in most cases using design space exploration on system components according to the execution time of security solutions. Here, the authors seek the optimal solution that satisfies both security requirements and performance requirements. (This is the case in S7 and S12.) It is important to note that other system requirements are either considered directly or during the verification step. In the latter, various techniques are used (model checking, virtual prototyping, qualitative and quantitative analyses, etc.)

In the implementation phase, methodologies focus on implementing security mechanisms, developing secure hardware and software, and code generation. In addition, security is tested by various techniques such as penetration testing, fuzzing, and prototyping. Few studies provide the detail of these steps, and the majority just describe the steps without giving the tools or practicing them.

In the deployment phase, the steps consider run-time monitoring, the incident response, and security management, which includes patch management, periodic risk assessment, etc.

4.3.2. RQ2.2—Which Security Integration Mechanism Is Used by the Methodology?

Security integration can be achieved through four different mechanisms [3].

- **Merging:** The base workflow is modified to incorporate (merge) a set of security steps from the security workflow.
- **Coupling:** The security and base workflows are independent but interconnected through connection points, with each utilizing and feeding back into the other.
- **Triggering:** In this mechanism, a single connection point to the base workflow triggers the security workflow. After completion, the results can, but do not have to, be returned to the base workflow.
- **Harmonization:** In this mechanism, the security workflow is adapted to be as similar as possible to the base workflow; i.e., the two workflows are harmonized.

Table 7 shows that the most commonly used mechanisms are merging (25 papers) and triggering (22 papers). There are five studies that use harmonization as a security integration mechanism. For instance, in S30, the authors try to harmonize safety, performance, and security requirements during the design of CPSs. In S7, S12, and S29, the authors use design space exploration to choose the system components that satisfy the security and performance requirements by merging the development workflow and the security workflow to choose the optimal solution design.

Table 7. Methodologies for security integration mechanisms.

Security Integration Mechanisms	Study ID	Number of Studies
Merging	S2, S6, S7, S10, S12, S16, S17, S18, S22, S24, S25, S27, S29, S33, S35, S38, S40, S42, S43, S45, S46, S49, S52, S53, and S55	25
Triggering	S3, S4, S5, S8, S11, S15, S28, S20, S21, S23, S26, S37, S47, S31, S32, S34, S41, S44, S48, S50, S51, and S54	22
Harmonization	S1, S9, S14, S30, and S39	5
Coupling	S13, S19, and S36	3

4.3.3. RQ2.3—What Element Did the Methodology Focus on to Consider Security?

Figure 17 shows the distribution of papers according to the focus of the methodology. We can notice that more methodologies are system-centric or asset-centric than those that are attacker-centric. Also, almost 16% of studies are system-centric and asset-centric at the same time.

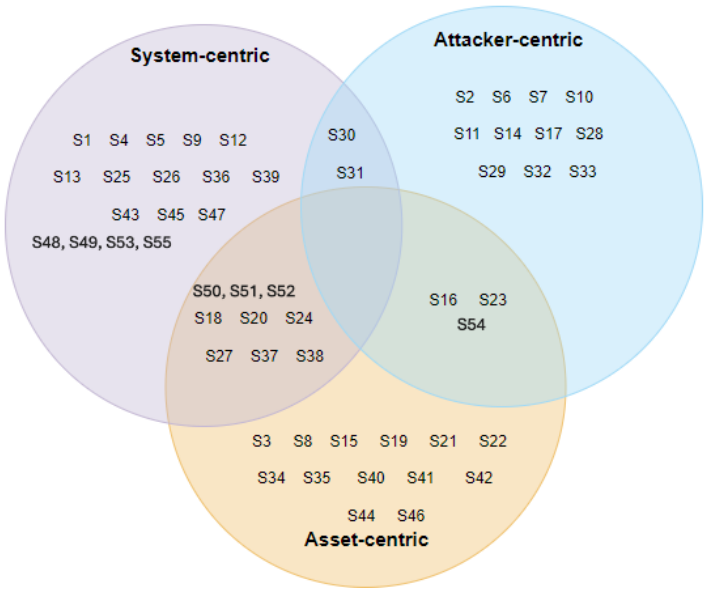


Figure 17. Distribution of papers according to the focus of the methodology.

As shown in Figure 18, we observe that the attacker-centric methodologies essentially use attack/threat aspects (9 out of 11 papers). For the asset-centric methodologies, the majority use attacks/threats or security controls. In these methodologies, security is analyzed either by how an asset can be attacked, how an asset can be protected for maximum security, or a combination of two. On the other hand, in system-centric methodologies, the three types of security aspect are used. Approximately 47% of methodologies (those that are system-centric or a combination of system-centric and another type of methodology) essentially use the attack/threat aspects.

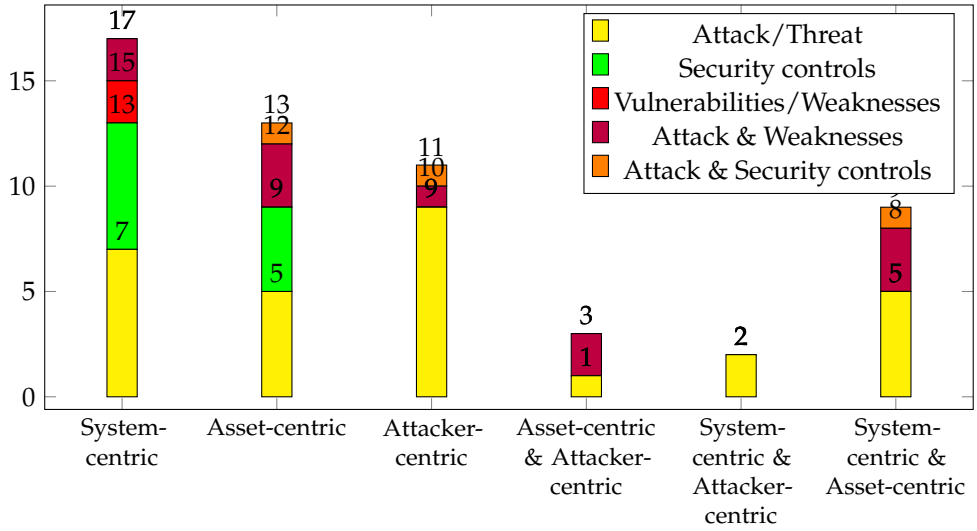


Figure 18. Distribution of papers per methodology focus according to security aspects.

4.3.4. RQ2.4—What Are the Methodology Models and How Were They Modeled?

Figure 19 shows the distribution of papers according to the modeling languages type. We notice that eight primary studies are not model-based or do not specify the language they use. (These primary studies present the general secure workflow without entering into details about the language, for example, S9, S13, and S24). The model-based studies are classified into UML and derivative or non-UML. Some studies use models from the two classes, for example, in S41. The authors use attack trees (non-UML) to model attack scenarios and UML to model the system. In S25, the authors use the BCL language to model system requirements in the BSL language, and they use SysML to model the system. Also, we can notice that the number of studies using non-UML languages is twice the number of studies using UML languages. Indeed, many studies not using UML use a domain-specific language (DSL) as the modeling language. This reflects the heterogeneous nature of CPSs [7]. The DSL approaches for modeling CPSs stem from various design fields (software engineering, mechanical engineering, electrical engineering, electronics engineering, and security engineering) [37].

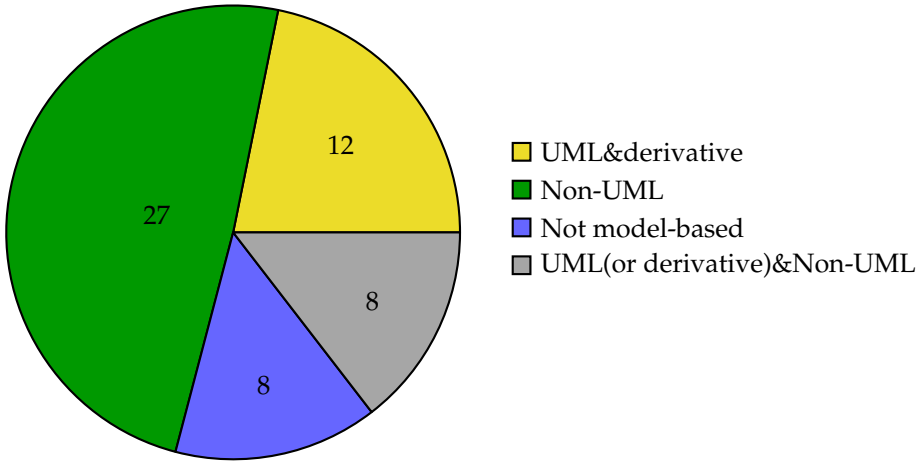


Figure 19. Distribution of papers according to the modeling language.

As we can see in Figure 20, the majority of studies use semi-formal languages. Ten studies use formal languages, and three use informal languages. Table 8 shows the elements being modeled in different primary studies and the languages used to model them. Trees are used to model attacks, hazards, and security mechanisms. The system is modeled using

semi-formal languages such as Secure Tropos, SysML/UML, IEC 61499, P&ID, and Procom. It is also modeled using a formal language. This is the case for S8, where the authors use a continuous Markov chain to model the system architecture. Security mechanisms are modeled using UML, trees, and the Ecore language. Different types of constraints and requirements are modeled using a first-order logic formal language and other semi-formal languages such as the BCI language and SysML/UML. Many domain-specific languages are specifically used to model the system, the data, and functions.

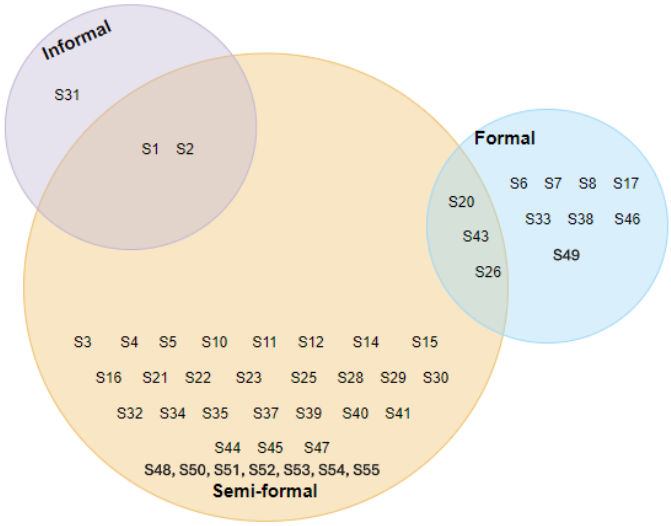


Figure 20. Number of papers that use models according to the modeling type.

Table 8. Languages used in the primary studies.

Type of Element Being Modeled	The Element Being Modeled	Model Languages Used
System	System Components	Secure Tropos, continuous-time Markov chain, P&ID, IEC 61499, DSL, Procom, SysML/UML, Ptolemy II, AADL, TOGAF Archimate, and EAST-ADL
	Function/task	DSL, stochastic Petri net, UML, and task/resource graph
	System behavior	State machine diagram
	Data	DSL and text
	Information flow	Data flow diagram, SysML/UML, and graph
	Operation/mission	UML
Security aspects	Threat/attack	Attack tree, Bayesian network-based attack graph (BNAG), text, Ptolemy 2, stochastic petri net, and UML
	Security mechanisms/ patterns	UML and tree
Safety aspects Requirements	Hazards/failures	Fault tree
	Safety use cases Requirements	SysML (sequence diagram) First-order logic, SysML/UML, and BCI

5. Open Issues and Research Directions

In this section, we discuss open issues and future research directions for security-by-design methodologies, based on earlier results.

5.1. Development Lifecycle Phases Beyond the Design Phase Are Not Sufficiently Considered

All the studied papers consider the design phase. Moreover, 60% of studies consider only the design phase. In essence, the design phase is the common phase where security is integrated when dealing with the security-by-design paradigm. However, this narrow focus poses a significant risk. Subsequent phases in the development lifecycle present opportunities for introducing modifications and vulnerabilities into the system. For example, "A secure design can still have implementation defects leading to vulnerabilities that may be exploited" [38] such as buffer overflows, race conditions, etc. For this, security integration should concern all development lifecycle phases, especially design and implementation where most vulnerabilities are introduced into the system.

5.2. Vulnerabilities/Flaws Are Not Sufficiently Considered in Security Analysis

The predominant aspect analyzed in security assessments is attacks, most likely due to the widespread use of attack trees as the leading model for representing cyber risk [39]. However, a limited number of primary studies, as illustrated in Figure 12, delve into security weaknesses and vulnerabilities in their analyses. Yet, it is worth noting that attacks or threats exploit weaknesses and vulnerabilities to induce system damage. Furthermore, mitigating system weaknesses and vulnerabilities not only safeguards against known threats but also shields against unknown ones [40]. Thus, prioritizing approaches that address system weaknesses and vulnerabilities may offer a more rational strategy.

5.3. Security Properties/Objectives Are Not Often Very Explicit

Approximately 56% of primary studies consider security objectives implicitly. In this type of primary study, the authors either focus on threats/attacks as an input to their methodology from which they should protect their systems without specifying which security objectives they intend to enforce, or they deal with security objectives in a general way; for example, they extract them from the requirements (that change from one system to another) [41]. In addition, in the studies where security objectives are explicitly mentioned, integrity and confidentiality are the most considered, although availability is the top priority in this type of system [13]. The rationale behind this is that availability is commonly dealt with as a safety objective in this type of system. For this, more studies should consider availability from the cybersecurity perspective in the future. Moreover, from the perspective of security engineering, security methodologies should be guided by security properties. By clearly delineating the security objectives being addressed, security by design could systematically and persuasively address these concerns [7].

5.4. Conflicts Related to Security and Other System Requirements Are Not Well Addressed

Numerous primary studies have incorporated various system requirements alongside security considerations. For instance, safety concerns are addressed to mitigate hazards arising from security issues. Additionally, many studies involve a trade-off between performance and security. Particularly in embedded systems, trade-offs are made between resource limitations or usage and implementing security countermeasures due to inherent system constraints. Moreover, ensuring functional objectives necessitates aligning security measures with functional requirements. However, the majority of these studies ignore conflicts between security and other system requirements. Thus, there is a need for more

comprehensive methodologies to address conflicts between security and other system requirements early in the design phase.

5.5. Challenges in Security-by-Design Methodologies: Incomplete Standard Integration, Insufficient Technical Detail, Limited Formal Verification, Low Automation, and Neglect of Security Engineering Principles

While some studies incorporate domain-specific standards, their integration remains limited. For example, study S18 proposes a framework that aligns ISA 84 (safety) and ISA 99/ISA 62443 (security) to develop a unified evaluation approach for Industrial Control Systems and Subsystems (ICSSs), taking both cybersecurity and safety into account during the design phase. Similarly, the methodology in S34 facilitates the secure-by-design development of IEC 61499 distributed applications by leveraging security requirements drawn from the IEC 62443 standard. In S35, the authors introduce a security zoning strategy that segments the system architecture into zones sharing uniform security requirements, as defined by ISA 62443. Despite these contributions, the broader body of work rarely incorporates standards. This gap is significant, as the adoption of established standards enhances security posture, fosters trust and regulatory compliance, mitigates risks, reduces costs, and promotes opportunities for continuous improvements and competitive advantages.

Moreover, only a limited number of studies employ simulations or formal verification techniques to validate selected security countermeasures against threats or system-level requirements. For instance, S45 uses virtual prototyping to enable early-stage security evaluations in embedded system development. Similarly, S28 and S8 apply model checking to verify security properties within system models. Despite their proven benefits, such techniques remain underutilized. Formal verification offers mathematical rigor and correctness, which are particularly valuable for safety-critical systems, while simulations allow for the exploration of system behavior under varying scenarios, enabling the early detection of design flaws in a cost-effective manner.

In addition, many primary studies provide only high-level methodological guidance, lacking detailed technical specifications. These approaches often fail to specify the tools employed, conflict resolution strategies, or how security is reconciled with other system requirements. Approximately 36% of the reviewed studies address cyber–physical systems (CPSs) in a generalized manner without tailoring their methodologies to specific CPS domains. Given the domain-dependent nature of CPSs—whose behavior and requirements are closely tied to their operational context—security integration cannot be uniformly applied across all CPS types. Customization based on system functionality and environmental interaction is essential for effective security by design.

Furthermore, a majority of methodologies are not fully automated, frequently requiring expert intervention to identify security risks or select appropriate mitigation strategies. Even when validation and verification mechanisms are included, a manual redesign is typically necessary in cases of non-conformance. Increased automation in security-by-design processes represents a promising avenue for future research, with the potential to reduce both time and cost while improving consistency and scalability.

Finally, most studies focus predominantly on security analysis and give insufficient attention to the integration of mitigation measures—an essential phase in security engineering and central to preventative strategies. The selection and implementation of appropriate mitigation strategies have significant implications for other system requirements, such as performance, safety, and functionality. Therefore, comprehensive engineering of security solutions and elucidating their interaction with broader system parameters warrant deeper investigation and methodological integration.

5.6. Lack of Independent and Influential Security-by-Design Methodologies

Most studies integrate security using "Merging" or "Triggering" mechanisms. This means that the majority of methodologies are either dependent on the domain and the base workflow they were merged with (for the merging mechanisms) or security decisions that do not sufficiently impact design decisions (for the triggering mechanism). A combination of the two could increase the influence and the unreliability criteria when integrating security.

5.7. Lack of Joint Consideration of Different Perspectives

Only 16% of the studies are system-centric and asset-centric. These studies concern complex systems, where security is first analyzed at a system level and then at a component level. This granularity in the integration of security guarantees the efficiency of security decisions at both the system level and the component level (or system-of-system level and system level).

Notably, none of the surveyed studies simultaneously integrates all three perspectives (attacker-centric, asset-centric, and system-centric) within a unified security-by-design methodology. This gap reveals a significant opportunity for future research. A comprehensive framework that jointly considers the system architecture, the value and vulnerabilities of critical assets, and the capabilities, goals, and potential pathways of adversaries would enable a more robust context-aware security engineering process. Such an integrative approach could improve the identification of attack surfaces, optimize the prioritization of defensive measures, and enhance the overall resilience of cyber–physical systems by aligning threat modeling with both system behavior and asset value. Developing methodologies that incorporate these three dimensions would represent a substantial advancement in the field and could serve as a foundation for more adaptive and risk-informed security design practices.

6. Threats to Validity

This section presents the threats that could weaken our approach and how we overcome them.

6.1. Construct Validity

"It concerns the validity of the extracted data with respect to our research questions" [25]. Two threats can be defined in construct validity. Firstly, the research string can be one of these. We construct the systematic mapping research string using a wide variety of words and synonyms in order to expand the research scope. The choice of keywords is made after several discussions between the authors of this systematic mapping study. Secondly, to mitigate the threat of the choice of digital libraries, we conduct our automatic research on the three popular digital bases in our domain. Moreover, we extend our research process using the backward and forward snowballing procedure.

6.2. Internal Validity

This corresponds to "influences that can affect the independent variable with respect to causality, without the researcher's knowledge" [24].

In our case, it concerns the effect caused by the defined protocol and data collection. The first threat is mitigated by multiple verification steps and discussions between all the study members. Also, a well-defined tree of concepts and a rigorously chosen classification helped minimize the second threat effects. Concerning the conclusion that is drawn, we support our analysis using well-assessed statistical graphs, thereby allowing for the replication of our results.

6.3. Conclusion Validity

Conclusion validity concerns issues that affect the ability to draw the correct conclusion about relations between the extracted data and the resulting findings. This means the possibility that data observation could lead to incorrect findings. For this, the analysis and the synthesis of the collected data are discussed and validated periodically by all the authors.

6.4. External Validity

External validity includes "conditions that limit our ability to generalize the results of the study" [24].

One threat of this category corresponds to having a set of primary studies that are not representative of all studies on security by design for CPSs. The use of an automatic search and snowballing help mitigate this threat by selecting a representative sample of studies in the research area. Moreover, the clear and well-defined exclusion criteria contribute to the exclusive selection of primary studies relevant to our main research questions.

7. Conclusions and Future Work

This paper reports the results of a systematic mapping study on the existing security-by-design methodologies for ICSs from a CPS perspective. This study highlights the publication trends, open issues, and future directions of security-by-design methodologies for the period between 2012 and 2024. Also, it was designed and conducted based on a well-defined systematic mapping study protocol and data extraction scheme in order to guarantee the quality of the results and their analysis. Our analysis of the 55 selected primary studies reveals the answers to the two research questions:

- *RQ1: How is the security-by-design paradigm defined?* The "security-by-design" paradigm, in its most effective form, advocates for the integration of security measures from the very inception of the development lifecycle. However, the common focus is too often placed exclusively on the design phase, a practice that, while crucial, is insufficient on its own. New and significant vulnerabilities can, and often do, emerge during the subsequent implementation and deployment phases. Therefore, a truly robust security-by-design methodology must extend its reach beyond mere architectural planning to permeate every stage of development. It also tends to focus more on addressing potential attacks and threats rather than underlying system vulnerabilities, which can lead to gaps in system resilience and disruption prevention from both known and unknown threats. Security objectives are frequently considered implicitly, with a greater emphasis on integrity and confidentiality over availability, despite the latter being critical in many systems. Additionally, while some studies recognize the need to balance security with other system requirements such as safety and performance, these conflicts are not adequately resolved. To improve the security-by-design approach, it should encompass the entire development lifecycle, prioritize addressing vulnerabilities, clearly define security objectives, and effectively manage conflicts that may arise with other system requirements.
- *RQ2: What are the characteristics of a security-by-design methodology?* A security-by-design methodology emphasizes building cybersecurity and safety into systems from the outset by following several core principles. Many approaches align with established domain standards, providing structured evaluation frameworks that integrate safety and security requirements. However, simulation and formal verification, while beneficial for understanding system behavior and ensuring rigor, are rarely implemented. Additionally, these methodologies often lack technical depth, which limits their applicability to complex CPSs with diverse needs. Manual intervention is still

common, as automation tools are underutilized, suggesting an opportunity to streamline processes and lessen dependency on human expertise. Security solutions also tend to prioritize analysis over practical integration, leaving mitigation strategies underdeveloped. Lastly, many methodologies rely on "Merging" or "Triggering" mechanisms independently, though combining these could better address both system-wide and component-level security, ultimately enhancing decision-making in complex systems.

Strategically, this systematic mapping study was undertaken to inform future works and define precise research directions that are crucial, especially for engineering a systematic and automated security-by-design methodology for ICSs. The core impetus for this methodology is to effectively address and resolve the pressing open issues that our prior analysis brought to light (as discussed in Section 5). Through the rigorous exploration of these outlined research avenues, our ultimate objective is to build a more resilient security framework for ICSs, one where security considerations are intrinsically integrated throughout the entire design lifecycle.

Author Contributions: Conceptualization, A.E., S.M.-K., F.O. and P.B.; methodology, A.E., S.M.-K., F.O. and P.B.; validation, A.E., S.M.-K., F.O. and P.B.; formal analysis, A.E.; investigation, A.E.; data curation, A.E.; writing—original draft preparation, A.E. and S.M.-K.; writing—review and editing, S.M.-K., F.O. and P.B.; visualization, A.E.; supervision, S.M.-K., F.O. and P.B.; project administration, A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data can be found in reference [33].

Conflicts of Interest: The Authors A.E. and S. M.-K. are employed by the company Segula Engineering. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Selected Studies

S1	Saadatmand, Mehrdad, and Thomas Leveque. "Modeling security aspects in distributed real-time component-based embedded systems." 2012 Ninth International Conference on Information Technology-New Generations. IEEE, 2012.
S2	J. F. Ruiz, R. Harjani, A. Mana, V. Desnitsky, I. Kotenko and A. Chechulin, "A Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components," 2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing, Munich, Germany, 2012, pp. 261–268,
S3	Vasilevskaya, Maria, et al. "Integrating security mechanisms into embedded systems by domain-specific modelling." Security and Communication Networks 7.12 (2014): 2815–2832.
S4	R. Oates, D. Foulkes, G. Herries and D. Banham, "Practical extensions of safety critical engineering processes for securing industrial control systems," 8th IET International System Safety Conference incorporating the Cyber Security Conference 2013, Cardiff, 2013, pp. 1–6,
S5	William Young and Nancy Leveson, "Systems thinking for safety and security", 29th Annual Computer Security Applications Conference (ACSAC '13). Association for Computing Machinery, New York, NY, USA, 1–8,
S6	A. Wasicek, P. Derler and E. A. Lee, "Aspect-oriented modeling of attacks in automotive Cyber-Physical Systems," 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2014, pp. 1–6,

S7	I. Stierand, S. Malipatlolla, S. Fröschle, A. Stühring and S. Henkler, "Integrating the Security Aspect into Design Space Exploration of Embedded Systems," 2014 IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, 2014, pp. 371–376,
S8	P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy and S. Chakraborty, "Security analysis of automotive architectures using probabilistic model checking," 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2015, pp. 1–6,
S9	C. Schmittner, Z. Ma and E. Schoitsch, "Combined safety and security development lifecylce," 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, UK, 2015, pp. 1408–1415.
S10	Sabaliauskaite, Giedre and Aditya P. Mathur. "Aligning Cyber-Physical System Safety and Security." CSDM Asia (2014),
S11	S. Marrone, R. J. Rodríguez, R. Nardone, F. Flammini, V. Vittorini, "On synergies of cyber and physical security modelling in vulnerability assessment of railway systems, Computers &Electrical Engineering, Volume 47, 2015
S12	A. Motii, B. Hamid, A. Lanusse and J. -M. Bruel, "Guiding the Selection of Security Patterns for Real-Time Systems," 2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS), Dubai, United Arab Emirates, 2016, pp. 155–164,
S13	S. Cong, M. Jianfeng and Y. Qingsong, "On the Architecture and Development Life Cycle of Secure Cyber-Physical Systems," in Journal of Communications and Information Networks, vol. 1, no. 4, pp. 1–21, Dec. 2016,
S14	J. -P. Nicklas, M. Mamrot, P. Winzer, D. Lichte, S. Marchlewitz and K. -D. Wolf, "Use case based approach for an integrated consideration of safety and security aspects for smart home applications," 2016 11th System of Systems Engineering Conference (SoSE), Kongsberg, Norway, 2016, pp. 1–6,
S15	C. Bernardeschi ,M. Di Natale, G. Dini, D. Varano, "Modeling and generation of secure component communications in AUTOSAR",
S16	Cui, Jin, and Giedre Sabaliauskaite. "On the alignment of safety and security for autonomous vehicles." IARIA CYBER, Barcelona, Spain (2017).
S17	Tan, Benjamin, Morteza Biglari-Abhari, and Zoran Salcic. "An automated security-aware approach for design of embedded systems on MPSoC." ACM Transactions on Embedded Computing Systems (TECS) 16.5s (2017): 1–20.
S18	Gabriel, Angelito, Juan Shi, and Cagil Ozansoy. "A proposed alignment of the National Institute of Standards and Technology Framework with the funnel risk graph method." IEEE Access 5 (2017): 12103–12113.
S19	Josyula, Surya Kant, and Daya Gupta. "A new security methodology for internet of things." 2017 international conference on computing, communication and automation (ICCCA). IEEE, 2017.
S20	Neureiter, Christian, et al. "A concept for engineering smart grid security requirements based on SGAM models." Computer Science-Research and Development 31 (2016): 65–71.
S21	Bakirtzis, Georgios, et al. "A model-based approach to security analysis for cyber-physical systems." 2018 Annual IEEE International Systems Conference (SysCon). IEEE, 2018.
S22	Geismann, Johannes, Christopher Gerking, and Eric Bodden. "Towards ensuring security by design in cyber-physical systems engineering processes." Proceedings of the 2018 international conference on software and system process. 2018.
S23	Mouratidis, Haralambos, and Vasiliki Diamantopoulou. "A security analysis method for industrial Internet of Things." IEEE Transactions on Industrial Informatics 14.9 (2018): 4093–4100.
S24	Lisova, Elena, et al. "A systematic way to incorporate security in safety analysis." 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2018.
S25	Passerone, Roberto, et al. "A methodology for the design of safety-compliant and secure communication of autonomous vehicles." IEEE Access 7 (2019): 125022–125037.

S26	Tantawy, Ashraf, Sherif Abdelwahed, and Qian Chen. "Continuous stirred tank reactors: Modeling and simulation for CPS security assessment." 2019 11th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2019.
S27	Eckhart, Matthias, et al. "Security development lifecycle for cyber-physical production systems." IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society. Vol. 1. IEEE, 2019.
S28	Mili, Saoussen, Nga Nguyen, and Rachid Chelouah. "Transformation-based approach to security verification for cyber-physical systems." IEEE Systems Journal 13.4 (2019): 3989–4000.
S29	Gressl, Lukas, Christian Steger, and Ulrich Neffe. "Consideration of security attacks in the design space exploration of embedded systems." 2019 22nd Euromicro Conference on Digital System Design (DSD). IEEE, 2019.
S30	Apvrille, Ludovic, and Letitia W. Li. "Harmonizing safety, security and performance requirements in embedded systems." 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019.
S31	Chattopadhyay, Anupam, Kwok-Yan Lam, and Yaswanth Tavva. "Autonomous vehicle: Security by design." IEEE Transactions on Intelligent Transportation Systems 22.11 (2020): 7015–7029.
S32	Aigner, Andreas, and Abdelmajid Khelil. "A Semantic Model-based Security Engineering Framework for Cyber-Physical Systems." 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2020.
S33	Levshun, Dmitry, et al. "Design and verification of a mobile robot based on the integrated model of cyber-Physical systems." Simulation modelling practice and theory 105 (2020): 102151.
S34	Tanveer, Awais, Roopak Sinha, and Matthew MY Kuo. "Secure links: secure-by-design communications in IEC 61499 industrial control applications." IEEE Transactions on Industrial Informatics 17.6 (2020): 3992–4002.
S35	Kern, Matthias, et al. "An architecture-based modeling approach using data flows for zone concepts in industry 4.0." 2020 IEEE International Symposium on Systems Engineering (ISSE). IEEE, 2020.
S36	Moukahal, Lama J., Mohammad Zulkernine, and Martin Soukup. "Towards a secure software lifecycle for autonomous vehicles." 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2021.
S37	Japs, Sergej, Harald Anacker, and Roman Dumitrescu. "SAVE: Security & safety by model-based systems engineering on the example of automotive industry." procedia CIRP 100 (2021): 187–192.
S38	Tripathi, Dipty, et al. "Model based security verification of Cyber-Physical System based on Petrinet: A case study of Nuclear power plant." Annals of Nuclear Energy 159 (2021): 108306.
S39	Suri, Kunal, Gabriel Pedroza, and Patrick Leserf. "Model-Based Approach for Co-optimization of Safety and Security Objectives in Design of Critical Architectures." Model and Data Engineering: 10th International Conference, MEDI 2021, Tallinn, Estonia, June 21–23, 2021, Proceedings 10. Springer International Publishing, 2021.
S40	Larsen, Martin H., Gerrit Muller, and Satyanarayana Kokkula. "A Conceptual Model-Based Systems Engineering Method for Creating Secure Cyber-Physical Systems." INCOSE International Symposium. Vol. 32. 2022.
S41	Escamilla-Ambrosio, Ponciano Jorge, et al. "IoTsecM: a UML/SysML extension for internet of things security modeling." Ieee Access 9 (2021): 154112–154135.
S42	Aranha, Helder, et al. "Securing the metrological chain in IoT environments: an architectural framework." 2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT). IEEE, 2021.
S43	Quamara, Megha, Gabriel Pedroza, and Brahim Hamid. "Formal analysis approach for multi-layered system safety and security co-engineering." European Dependable Computing Conference. Cham: Springer International Publishing, 2022.

S44	Casola, Valentina, et al. "Designing Secure and Resilient Cyber-Physical Systems: A Model-based Moving Target Defense Approach." IEEE Transactions on Emerging Topics in Computing (2022).
S45	Mahmoodi, Yasamin, et al. "Security Analysis of Embedded Systems Using Virtual Prototyping."
S46	Promyslov, Vitaly G., Kirill V. Semenkov, and Georgy V. Promyslov. "Practical Method of the I&C System Security Architecture Design Using Graph Models." IFAC-PapersOnLine 55.9 (2022): 227–232.
S47	Sion, Laurens, et al. "Towards automated security design flaw detection." 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW). IEEE, 2019.
S48	SHAKED, Avi. A model-based methodology to support systems security design and assessment. Journal of Industrial Information Integration, 2023, vol. 33, p. 100465.
S49	LISBOA MALAQUIAS, Felipe, GIANTAMIDIS, Georgios, BASAGIANNIS, Stylianos, et al. Towards a Methodology to Design Provably Secure Cyber-physical Systems. ACM SIGAda Ada Letters, 2023, vol. 43, no 1, p. 94–99.
S50	KHALIL, Shaymaa Mamdouh, BAHSI, Hayretdin, OCHIENG'DOLA, Henry, et al. Threat modeling of cyber-physical systemsa case study of a microgrid system. Computers & Security, 2023, vol. 124, p. 102950.
S51	GOWDANAKATTE, Shwetha, RAY, Indrakshi, et ABDELGAWAD, Mahmoud. Model based risk assessment and risk mitigation framework for cyber-physical systems. In : 2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPSISA). IEEE, 2023. p. 203–212.
S52	FÖLDVÁRI, András, BRANCATI, Francesco, et PATARICZA, András. Preliminary risk and mitigation assessment in cyber-physical systems. In : 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2023. p. 267–274.
S53	AMIRI, Amirali, STEINDL, Gernot, GORTON, Ian, et al. Integrated safety and security by design in the IT/OT convergence of industrial systems: A graph-based approach. In : 2024 IEEE International Conference on Software Services Engineering (SSE). IEEE, 2024. p. 123–129.
S54	FISCHER, Alexander, TOLVANEN, Juha-Pekka, et KOLAGARI, Ramin Tavakoli. Automotive Cybersecurity Engineering with Modeling Support. In : 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS). IEEE, 2024. p. 319–329.
S55	HOSSEINI, Ali M., SAUTER, Thilo, et KASTNER, Wolfgang. Integrating Security into Industrial Control System Architecture Based on IEC 42010. In : 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2024. p. 1–8.

References

1. Short History of Manufacturing: From Industry 1.0 to Industry 4.0, 2021. Available online: https://kfactory.eu/the-industrial~-revolution-short-history-of-manufacturing (accessed on 16 June 2025).
2. Stouffer, K.; Pease, M.; Tang, C.; Zimmerman, T.; Pillitteri, V.; Lightman, S.; Hahn, A.; Saravia, S.; Sherule, A.; Thompson, M. *Guide to Operational Technology (OT) Security: Technical Report NIST Special Publication (SP) 800-82 Rev. 3*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]
3. Fluchs, S.; Tasten, E.; Mertens, M.; Horch, A.; Drath, R.; Fay, A. Security by Design Integration Mechanisms for Industrial Control Systems. In Proceedings of the IECON 2022—48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 17–20 October 2022; pp. 1–6, ISSN 2577-1647. [CrossRef]
4. Ani, U.P.D.; He, H.M.; Tiwari, A. Review of cybersecurity issues in industrial critical infrastructure: Manufacturing in perspective. *J. Cyber Secur. Technol.* **2017**, *1*, 32–74. [CrossRef]
5. Cheminod, M.; Durante, L.; Valenzano, A. Review of Security Issues in Industrial Networks. *IEEE Trans. Ind. Inform.* **2013**, *9*, 277–293. [CrossRef]
6. European Commission. Cyber Resilience Act | Shaping Europe's Digital Future. 2024. Available online: https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act (accessed on 16 June 2025).

7. Nguyen, P.H.; Ali, S.; Yue, T. Model-based security engineering for cyber-physical systems: A systematic mapping study. *Inf. Softw. Technol.* **2017**, *83*, 116–135. [CrossRef]
8. Geismann, J.; Bodden, E. A systematic literature review of model-driven security engineering for cyber–physical systems. *J. Syst. Softw.* **2020**, *169*, 110697. [CrossRef]
9. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report EBSE 2007-01; Keele University: Newcastle, UK; Durham University: Durham, UK, 2007.
10. Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* **2015**, *64*, 1–18. [CrossRef]
11. Elmarkez, A.; Kesraoui-Mesli, S.; Oquendo, F.; Berruet, P. Insights on Security-by-Design of Cyber-Physical Production Systems: A Systematic Mapping. In Proceedings of the CIGI QUALITA MOSIM 2025—Conference on Modeling, Optimisation and Simulation, Troyes, France, 8–10 July 2025.
12. Lee, E.A. Computing needs time. *Commun. ACM* **2009**, *52*, 70–79. [CrossRef]
13. Humayed, A.; Lin, J.; Li, F.; Luo, B. Cyber-Physical Systems Security—A Survey. *IEEE Internet Things J.* **2017**, *4*, 1802–1831. [CrossRef]
14. Kumar, C. Cyber-physical Systems (CPS) Security: State of the Art and Research Opportunities for Information Systems Academics. *Commun. Assoc. Inf. Syst.* **2020**, *47*. [CrossRef]
15. Kayan, H.; Nunes, M.; Rana, O.; Burnap, P.; Perera, C. Cybersecurity of Industrial Cyber-Physical Systems: A Review. *ACM Comput. Surv.* **2022**, *54*, 1–35. [CrossRef]
16. Bouskela, D.; Falcone, A.; Garro, A.; Jardin, A.; Otter, M.; Thuy, N.; Tundis, A. Formal requirements modeling for cyber-physical systems engineering: An integrated solution based on FORM-L and Modelica. *Requir. Eng.* **2022**, *27*, 1–30. [CrossRef]
17. Glinz, M. On Non-Functional Requirements. In Proceedings of the 15th IEEE International Requirements Engineering Conference (RE 2007), Delhi, India, 15–19 October 2007; pp. 21–26, ISSN 2332-6441. [CrossRef]
18. Tripathi, D.; Singh, L.K.; Tripathi, A.K.; Chaturvedi, A. Model based security verification of Cyber-Physical System based on Petrinet: A case study of Nuclear power plant. *Ann. Nucl. Energy* **2021**, *159*, 108306. [CrossRef]
19. Ross, R.; Pillitteri, V.; Graubart, R.; Bodeau, D.; McQuaid, R. *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach*; Technical Report NIST Special Publication (SP) 800-160 Vol. 2 Rev. 1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021. [CrossRef]
20. Ross, R.; Winstead, M.; McEvilley, M. *Engineering Trustworthy Secure Systems*; Technical Report NIST Special Publication (SP) 800-160 Vol. 1 Rev. 1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022. [CrossRef]
21. National Institute of Standards and Technology. *Standards for Security Categorization of Federal Information and Information Systems*; Technical Report Federal Information Processing Standard (FIPS) 199; U.S. Department of Commerce: Gaithersburg, MD, USA, 2004. [CrossRef]
22. DoD Modeling and Simulation (M&S) Glossary. Available online: https://apps.dtic.mil/sti/citations/ADA349800 (accessed on 16 June 2025).
23. Modeling Language (Glossary)—SEBoK. Available online: https://sebokwiki.org/wiki/Modeling_Language_(glossary) (accessed on 16 June 2025).
24. Nguyen, P.H.; Kramer, M.; Klein, J.; Traon, Y.L. An extensive systematic review on the Model-Driven Development of secure systems. *Inf. Softw. Technol.* **2015**, *68*, 62–81. [CrossRef]
25. Lun, Y.Z.; D'Innocenzo, A.; Malavolta, I.; Benedetto, M.D.D. Cyber-Physical Systems Security: A Systematic Mapping Study. *arXiv* **2016**, arXiv:1605.09641.
26. Uzunov, A.V.; Fernandez, E.B.; Falkner, K. *Engineering Security into Distributed Systems: A Survey of Methodologies*; Verlag der Technischen Universität Graz: Graz, Austria, 2012. [CrossRef]
27. Kriaa, S.; Pietre-Cambacedes, L.; Bouissou, M.; Halgand, Y. A survey of approaches combining safety and security for industrial control systems. *Reliab. Eng. Syst. Saf.* **2015**, *139*, 156–178. [CrossRef]
28. Huang, S.; Poskitt, C.M.; Shar, L.K. Security Modelling for Cyber-Physical Systems: A Systematic Literature Review. *arXiv* **2024**, arXiv:2404.07527. [CrossRef]
29. Mashkoor, A.; Egyed, A.; Wille, R.; Stock, S. Model-driven engineering of safety and security software systems: A systematic mapping study and future research directions. *J. Softw. Evol. Process.* **2023**, *35*, e2457. [CrossRef]
30. Harkat, H.; Camarinha-Matos, L.M.; Goes, J.; Ahmed, H.F.T. Cyber-physical systems security: A systematic review. *Comput. Ind. Eng.* **2024**, *188*, 109891. [CrossRef]
31. Del-Real, C.; De Busser, E.; van den Berg, B. Shielding software systems: A comparison of security by design and privacy by design based on a systematic literature review. *Comput. Law Secur. Rev.* **2024**, *52*, 105933. [CrossRef]
32. Wohlin, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; pp. 1–10. [CrossRef]

33. Extracted Data Repository. Available online: https://shorturl.at/Lp0wi (accessed on 16 June 2025).
34. Advisory: COVID-19 Exploited by Malicious Cyber Actors. Available online: https://www.ncsc.gov.uk/news/covid-19-exploited-by-cyber-actors-advisory (accessed on 16 June 2025).
35. On the Architecture and Development Life Cycle of Secure Cyber-Physical Systems. *J. Commun. Inf. Netw.* **2016**, *1*, 1–21. [CrossRef]
36. Apvrille, L.; Li, L.W. Harmonizing Safety, Security and Performance Requirements in Embedded Systems. In Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition, Florence, Italy, 25–29 March 2019; pp. 1631–1636, ISSN 1558-1101. [CrossRef]
37. Mosterman, P.; Zander, J. Cyber-physical systems challenges: A needs analysis for collaborating embedded software systems. *Softw. Syst. Model.* **2015**, *15*, 5–16. [CrossRef]
38. A04 Insecure Design OWASP Top 10:2021. Available online: https://owasp.org/Top10/A04_2021-Insecure_Design/ (accessed on 16 June 2025).
39. Konsta, A.M.; Lluch Lafuente, A.; Spiga, B.; Dragoni, N. Survey: Automatic generation of attack trees and attack graphs. *Comput. Secur.* **2024**, *137*, 103602. [CrossRef]
40. Young, W.; Leveson, N. Systems thinking for safety and security. In Proceedings of the 29th Annual Computer Security Applications Conference, New Orleans, LA, USA, 9–13 December 2013; pp. 1–8. [CrossRef]
41. Ruiz, J.F.; Harjani, R.; Mana, A.; Desnitsky, V.; Kotenko, I.; Chechulin, A. A Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components. In Proceedings of the 2012 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, Munich, Germany, 15–17 February 2012; pp. 261–268, ISSN 2377-5750. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.