Enhancing Cyber-Physical Systems Security: A Comprehensive SRE Approach for Robust CPS Methodology

Shafiq ur Rehman[††]
srehman@imamu.edu.sa
College of Computer and Information Sciences, Imam Muhammad bin Saud Islamic University (IMSIU)
Riyadh, Saudi Arabia

Abstract
Cyber-Physical Systems (CPS) are introduced as complex, interconnected systems that combine physical components with computational elements and networking capabilities. They bridge the gap between the physical world and the digital world, enabling the monitoring and control of physical processes through embedded computing systems and networked communication. These systems introduce several security challenges. These challenges, if not addressed, can lead to vulnerabilities that may result in substantial losses. Therefore, it is crucial to thoroughly examine and address the security concerns associated with CPS to guarantee the safe and reliable operation of these systems. To handle these security concerns, different existing security requirements methods are considered but they were unable to produce required results because they were originally developed for software systems not for CPS and they are obsolete methods for CPS. In this paper, a Security Requirements Engineering Methodology for CPS (CPS-SREM) is proposed. A comparison of state-of-the-art methods (UMLSec, CLASP, SQUARE, SREP) and the proposed method is done and it has demonstrated that the proposed method performs better than existing SRE methods and enabling experts to uncover a broader spectrum of security requirements specific to CPS. Conclusion: The proposed method is also validated using a case study of the healthcare system and the results are promising. The proposed model will provide substantial advantages to both practitioners and researcher, assisting them in identifying the security requirements for CPS in Industry 4.0.

Keywords:
Cyber-Physical Systems (CPS), Security, Security Requirements Engineering, Threats, Healthcare.

1. Introduction

Cyber Physical Systems (CPS) represent a significant and rapidly advancing domainwithin the information industry. These systems empower the precise and real-time operation of smart applications and services through the seamless integration of cyber and physical systems [1]. They primarily consist of three interconnected layers, which are interconnected and they communicate through a complex “Network Layer”. The “Physical Layer” perceives the information from the “Real World” using sensors or other network devices and response to changes in the physical environment, that depends on the area of application. Meanwhile, the "Software Layer" takes charge of system control and processing the data gathered from the "Physical Layer". CPS systems have diverse applications in different domains such as energy production, aerospace, civil infrastructure, chemical industry, agricultural systems, autonomous systems like drones and self-driving vehicles. They also play a pivotal role in improving medical related services. Moreover, the application of CPS technology is growing in supply chain management, fostering environmentally friendly, cost-effective, and secure manufacturing processes [1,14]. Enhancements in CPS technology will enhance scalability, adaptability, usability, security, and safety, resulting in systems that are different from basic embedded systems.

Inherently, CPS are complex, geographically dispersed systems with diverse embedded devices like sensors and actuators, networked for real-time physical world monitoring and control. Resource scheduling, including activating actuators and sensors, is crucial but can be challenging due to data transmission over networks without proper security measures. The interconnected nature of these systems presents physical vulnerabilities while emphasizing the need for enhanced security and resilience against cyber threats, posing new challenges to traditional control, communication, and software theories [2]. Cyber-physical systems are widely applied in healthcare to enhance the efficiency of patient treatment by connecting various medical devices [4], these systems help in improving overall performance of healthcare systems as shown in figure 1. Medical Cyber-Physical Systems (MCPS) create vital, context-aware networks of medical devices by integrating physical and computational elements [28]. Leveraging recent IoT advancements like wireless sensors and connected medical devices, MCPS emerges as a promising platform to enhance patient treatment

Manuscript received May 5, 2024
Manuscript revised May 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.5.5

effectiveness and healthcare quality. They continuously monitor and analyze data from medical devices to assess a patient’s health condition, enabling timely treatment interventions through healthcare provider feedback or automated medical actuation. Besides having several uses, MCPS presents significant security and privacy concerns. The heterogeneity of MCPS, combined with the growing utilization of wireless and mobile technologies, introduces vulnerabilities and presents new attack surface area. These security breaches in MCPS may lead to unauthorized access to sensitive medical and personal health data [4,28].

A significant challenge in CPS development is the Security Requirements Engineering (SRE) phase, and this phase has to be implemented in every step of the development life-cycle because medical devices no longer only work independently but now have a wireless or wired connection to sensors and a network. The complexity of these devices has increased enormously and must therefore be created in a structured development process with a multilateral approach that takes all the stakeholders into account [1,14]. Hence, the deployment of updates to operational systems has become a critical aspect of CPS in healthcare settings. The continuous flow of information through CPS components poses inherent risks. Healthcare data, in particular, demands strong protection due to the sensitive personal health information involved. While rigorous security requirements can secure the data, software failures remain a concern. Automation is essential to update systems seamlessly without interrupting operations, promoting user acceptance and safeguarding against system failures. All security solutions in the CPS development must adhere to the three core "CIA" objectives: confidentiality, integrity, and availability [8]. Secure software developers now have to handle with the novel challenge of incorporating the physical layer into development. Consequently, an enhanced security requirements engineering methodology is essential for CPS to address these emerging risks comprehensively. This system should encompass risk analysis, stakeholder perspectives, and a focus on ensuring confidentiality, integrity, and availability within the developed system from its inception [1].

1.1. Security and Network Technology for CPS

In the development of a CPS, easy-to-use cryptographic libraries, end-to-end encryption and a well-designed patch management system are required. Following are some libraries that can be used for the development:

1.1.1. Networking and Cryptography Library (NaCl)

This is a very fast and easy-to-use software library that creates secure encryption and decryption, provides network communication and supports signatures. It is a very high-level implementation of the security features. The main advantages of this library are the speed and the easy implementation of security features. NaCl does not allow specific scenarios that would lower the overall security. It does not allow:

- Encryption without authentication
- Any data flow from secrets to load addresses or branch conditions
- Cryptographic primitives breakable in fewer than 2128 operations

A modification of NaCl is called as “libdsodium”. It improves the portability of NaClso it can be used on more platforms and not only on UNIX-based platforms. Libsodium also creates an extended API which should help improve the usability of NaCl.

1.1.2. Transport Layer Security (TLS)

TLS creates a good security for the transportation of data over the internet. It sends the data after a handshake and a key exchange encrypted to the receiving node where it gets decrypted back into plain text. TLS can be seen as standard for the transport of information these days. However, there are also many attacks to different TLS versions that can destroy the security TLS ensures. BEAST Attack, Breach Attack and FREAK attack are some of the attacks that affect to TLS [6]. Most of the attacks can be prevented by some workarounds or a focused implementation of TLS with a new standard such as TLS 1.2.

1.1.3. IoTree

IoTree establishes an end-to-end connection with a secure TLS 1.2 solution. It includes an easy but robust cryptographical key management. The keys are not hardcoded into the devices or generated in the production. IoTree provides a secure system by design and it is built this way and can help create a safer medical CPS environment.

1.1.4. Ethical Patch Management (EPM)

While developing CPS for healthcare, it crucially involves recognizing the significant role these systems play in human life, whether for health monitoring or devices like pacemakers. Ethical dilemmas arise when patching such systems due to the potential risk of harming human lives through inadequate quality assurance. Therefore, an explicit patch management step is required.

The Security Requirements Engineering Process (SREP) consists of nine activities to be consistently executed within each stage of the Unified Process (UP) [5]. SREP employs an asset-centric and risk-focused methodology, drawing inspiration from SQUARE, while integrating the Common Criteria (CC) into the Unified Process lifecycle model. This integration facilitates the classification of software developed using SREP into security levels defined by the CC [8]. However, it is worth noting that SREP inherits the criticisms associated with the CC [5].
This paper makes the following research contributions:

- A comprehensive security requirements engineering methodology is proposed to develop a robust and secure CPS.
- Introduce activities to increase the overall security of CPS.
- A case study of healthcare is used to assess the proposed security requirements engineering methodology for CPS, and the findings are presented in this paper.
- The findings would strongly support of researchers and practitioners in this field of research for CPS.

The remaining paper is organized as follows: Section II describes the background and existing work. Section III of the paper, explains the methodology of the proposed method in detail and also explains different activities involved in the entire process. In the next section IV, using case study of healthcare system, the proposed methodology is compared with state-of-the-art SRE methods. Section V and VI describe the conclusion.

2. Related Work

In recent years, the field of security requirements engineering has seen significant development, resulting in the proposal of various security frameworks through academic papers and scientific research [13,15]. Today, there exists a diverse range of approaches for security requirements engineering, which includes multilateral approaches like SQUARE, UML-based methods such as UMLSec, goal-oriented frameworks like KAOS/Tropos, and Common Criteria-based approaches like SREP [2]. Multilateral approaches, in particular, are considered more contemporary compared to unilateral ones, as they encompass stakeholder perspectives and prioritize the negotiation of compromises among differing stakeholder viewpoints [13,16]. The previous work can be categorized into four classes which work in different phases of cyber-physical system and its development lifecycle. These categories include UMLSec, CLASP, SREP, and SQUARE. In the following sections, the background of the four different methodologies is presented. These systems build the foundation for the proposed security requirements engineering methodology for smart healthcare systems.

2.1. UMLSec

UMLSec was invented by Jan Jürjens in 2002 [17] and is a lightweight extension for the unified modelling language (UML). It uses the extension mechanisms UML, provides and extends the modelling language with “Stereotypes, tagged values and constraints". It contains 21 different predefined stereotypes with predefined tags and constraints [17,18]. Stereotypes such as “Internet, encrypted and LAN” are the representations of the possible communication channels. These are the vulnerable points where a possible adversary can attack the system. UMLSec allows the user the possibility of creating a custom adversary model [19]. For CPS, there has to be a custom model which takes the wide attack angle of CPS into account. Many CPS can be accessed through Internet and LAN and also have other wireless communication channels and sensors that can be attacked [13]. Jürjens developed a security engineering methodology that is easy to include in the development of a software system. This ease of integration is due to UML being an industry standard for software system construction, widely recognized and familiar to developers [20]. Moreover, Jürjens supports his methodology over the years and due to that there are some examples where UMLSec was successfully applied [19–21]. Although considerable efforts have been made, industry-wide acceptance of UMLSec remains incomplete [21]. This might be attributed to its limited scope, primarily aiding in the construction of a secure software architecture [20]. Another problem of UMLSec formal correct UML-models can expand to a very complex model but its variations like UMLSec does not help to keep this problem on a low level [20].

2.2. Comprehensive, Lightweight Application Security Process (CLASP)

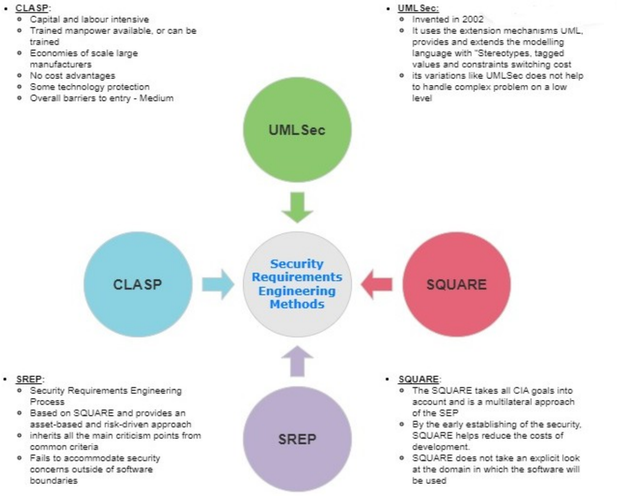
CLASP, the Comprehensive, Lightweight Application Security Process, was invented by Secure Software but is now available by the Open Web Application Security Project (OWASP). The complete process includes 24 independent activities and supplemental resources. These activities can be tailored around the software lifecycle the company or team uses, so it can be integrated in the process [22,27]. John Viega [27] clearly summarizes the CLASP process at a high level. However, main limitation of CLASP is its complex structure which cannot be easily understood and handled by any organization. The organization that wants to implement CLASP needs a security expert who has the inside knowledge to adjust and elicit the activities. This SEP may be more expensive for companies than other SEPs because of the longer training period [22,27].

2.3. Security Quality Requirements Engineering (SQUARE)

The Security Quality Requirements Engineering (SQUARE) was developed at the Carnegie Mellon Software Institute. The SQUARE takes all CIA goals into account and is a multilateral approach of the SEPs [24,26]. It establishes the security requirements engineering into the early stages of the software development and also through the whole development lifecycle [23,25]. By the early establishing of the security, SQUARE helps reduce the costs of development. However, SQUARE does not take an explicit look at the domain in which the softwarewill be used [8]. SQUARE finishes with security requirements that are classified and prioritized can help gain a better understanding of the gathered requirements. This is a big improvement in contrast to SEP without this elicitation categorization and prioritization. Another good point is that SQUARE considers all the CIA goals and dedicates a whole step for risk analysis [8,26]. This step is very important for use of SQUARE in CPS environments because of the large attack area a CPS offers. In an advanced methodology, this step should be extended to consider the variety of attack spots.

2.4. Security Requirements Engineering Process (SREP)

SREP is based on SQUARE and provides an asset-based and risk-driven approach. It integrates the Common Criteria (CC) in a lifecycle model, called the Unified Process. Through the help of the CC, software that was built with SREP can be easily categorized into the security levels which were provided by CC but it inherits all the main criticism points from CC [3]. UMLSec fragments can be easily used in the Security Requirement Engineering, Risk Management and Analysis [3]. This can help improve the SREP at the architecture layer. The developers of the SREP have noticed that, due to its iterative nature, this process has many positive features. SREP can handle changing requirements, it is easier to reuse and it corrects some errors done in previous iterations. Also, risks can be discovered earlier in the development and the users can achieve a better understanding with the help of the iterations and can improve it over the whole process with the next iteration. These aspects of SREP make it a good foundation for an extended SRE for CPS. In figure 1, summary of different groups of Security Requirements Engineering Methodologies is provided.



- CLASP:
  - Capital and labour intensive
  - Trained manpower available, or can be trained
  - Economies of scale large manufacturers
  - No cost advantages
  - Some technology protection
  - Overall barriers to entry - Medium

- UMLSec:
  - Invented in 2002
  - It uses the extension mechanisms UML, provides and extends the modelling language with "Stereotypes, tagged values and constraints switching cost
  - its variations like UMLSec does not help to handle complex problem on a low level

- SREP:
  - Security Requirements Engineering Process
  - Based on SQUARE and provides an asset-based and risk-driven approach
  - inherits all the main criticism points from common criteria
  - Fails to accommodate security concerns outside of software boundaries

- SQUARE:
  - The SQUARE takes all CIA goals into account and is a multilateral approach of the SEP
  - By the early establishing of the security, SQUARE helps reduce the costs of development.
  - SQUARE does not take an explicit look at the domain in which the software will be used

Figure 1. Four categories Security Requirements Engineering Methodologies and their limitations.

3. Proposed Security Requirements Engineering Methodology for CPS

- Due to the dynamic nature of security requirements engineering, an enhanced Security Requirements Engineering Methodology (SREM) is proposed for CPS, building upon the existing method. In this adapted version, the specific shortcomings of SRE are rectified to align with the requisites of cyber-physical systems. Furthermore, a modified CPS Security Resource Repository (CPS-SRR) has been designed. The extended repository now holds additional information about CPS and the associated domain, making it valuable for use in various CPS development. This allows organizations to gather more valuable knowledge in comparable situations. Figure 3 shows the flow of activities in the CPS-SREM and how much time each activity will likely take within the Unified Process (UP) and the process iterations. The explanation of these activities is deferred to the subsequent section labeled 4.1, wherein a medical application-based case study is employed to conduct a comparative analysis of SREP and CPS-SREM models, encompassing all the mentioned activities. Lifecycle models play a crucial role in software development, and the unified process is a widely used due to its incremental nature, aligning well with the incremental functions of SREM. Hence, the utilization of an incremental lifecycle model is essential when employing SREM.

3.1. CPS Security Resource Repository (CPS-SRR)

The CPS-SRR incorporates a domain extension, supplementary attack trees, and a UMLSec model intended to address the domain's architectural aspects. To manage the system's complexity, these extensive models can be subdivided into smaller, more manageable components. An additional component introduced in CPS-SRR is the inclusion of the Physical Environment node, where Misuse Cases, Attack Trees, and UMLSec models can be incorporated. This approach contributes to the development of a secure architecture for the physical environment, characterized by its asset-driven, threat-driven, and environment driven structure. Users can search for specific threats, assets, or environments to access security requirements collected from prior projects, streamlining the development process for CPS by providing developers with immediate access to relevant security requirements linked to threats, assets, or environments. A model of the extended (CPS-SREM) is given in Figure 3. The main change is incorporated in the physical layer. Addition of Misuse cases, UMLSec Model and Attack Trees in the extended version assists in structuring CPS-specific artifacts, identifying threats, and managing assets. Additionally, it enhances the overall project management process for securing CPS.

3.2. Stages of the CPS-SREM

Unilateral security requirements engineering methodologies are outdated now and they are not considered state-of-the-art in software development, especially for CPS due to their addition of physical layer. This is the main reason to update SREM to a multilateral approach which could be easily done with an update of the nine activities. These activities are not only updated with this new feature, but also get a new description that fits the development of CPS. The main information of SREP model is used in the proposed methodology, this information is gathered into Figure 4. All these steps are significant in developing a CPS for any application. Similarly, A CPS for healthcare system should have all these components. For example, a pacemaker could be attacked through the patch system. The pacemaker regulates the heartbeats of the patient so in a way it monitors the life of this person. A good patch management could close vulnerabilities at the same time that they are discovered. Security breaches in healthcare systems are often also a high risk for the life of patients. On the contrary, a badly designed patch management could open doors for attackers through the patch distribution to run non-authorized software on the pacemaker. The quality of a CPS can be responsible for the life of human beings. These CPS systems can take a life in one moment of failure.

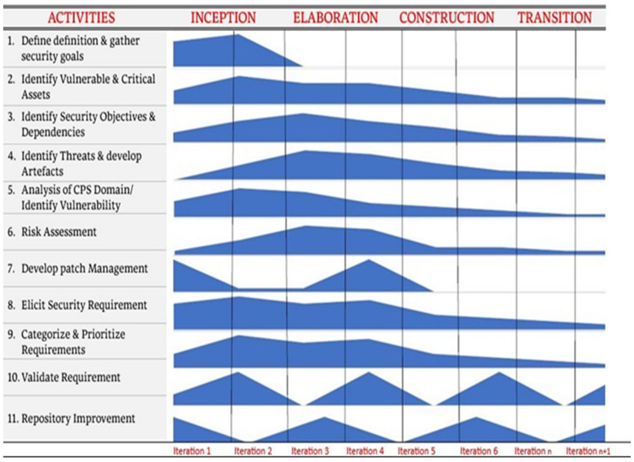


Figure 2. CPS-SREM Methodology

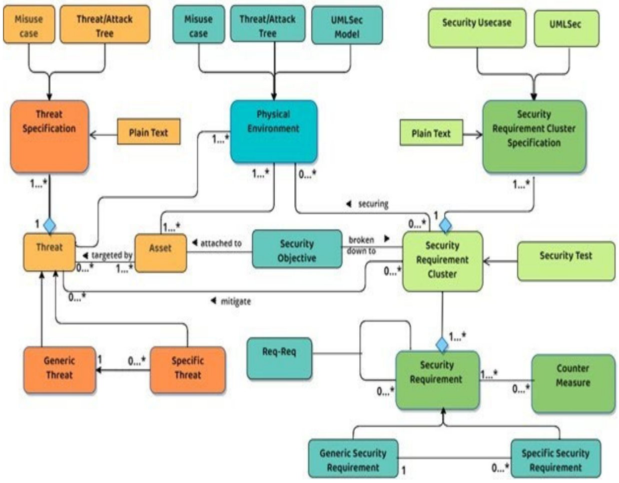


Figure 3. CPS-SREM Security Resource Repository

Cyber-physical systems are structured with a three-layer architecture. The initial layer is the physical layer, encompassing sensor nodes, control systems, and all components responsible for measuring or interacting with the physical environment. This layer introduces a novel challenge for CPS, rendering traditional techniques of safeguarding the system against attacks obsolete. The second layer is the network layer, which is responsible for managing real-time communication within the CPS using various network technologies. The final layer is the application layer, serving as the core component of the system. This layer serves as the focal point for all collected data within the CPS. It not only governs the CPS but also acts as the component that integrates sensors and other nodes into an intelligent system. Some points are added in the revised SREM which is used in the proposed method for CPS. Table 1 depicts different stages and additions of information in the revised version of the traditional security requirements engineering methods.

3.3. Threats to CPS

Identifying threats in any model is a significant challenge for security experts, affecting not only traditional software development but also CPS. This task is critical for achieving optimal security.

Table 1. Additional information added in the revised SREM in all stages

Stage	Traditional SREP	Changes in Revised-SREM for CPS
Agree on Definition and Gather Security Goals	Takes Unilateral security view approach	Takes multilateral security viewapproach
Identify Vulnerable and/or Critical Assets	In the SRR, previously known assets are added for later reuse.	Detailed analysis of all the as- sets including pyhsical aspects are also added
Identify Security Objectives and Dependencies	Produces the “Security Objectives Document”	No change
Identify Threats and Develop Artefacts	Assess threats but risk levels arenot defined	Risk levels are defined using a method from US Sandia Lab
Analysis of the Domain/Vulnerability Identification	The awareness of the domain and surrounding areas of the system	UMLSec or Attack Trees are used to find vulnerabilities and threats
Risk Assessment	Performs risk assessment without any simulation	Includes the NIST Risk Management Framework fragments and also the results of the simulation
Develop Patch Management	The specifications for the laterpatch deployment are planned	A well-designed quality assurance and patch distribution is used
Elicit Security Requirements	All the gathered information about the threats are analysed to elicit suitable security requirements	No change
Categorize Prioritize Requirements	Developed requirements arenow categorized and prioritized	Different well-defined focuspoints are defined and used
Requirement Inspection	Validate all the models, documents, artefacts and requirements that were developed	No change
Repository Improvement	After verification of gathered information, it is added in the repository	No change

The increased complexity of systems in CPS can make them more vulnerable to attacks. CPS comprises three primary layers, and the associated threats for each layer are outlined as follows:

1. The first layer is the physical layer. It includes the sensor nodes, control systems and everything else to measure or interact with the physical world.

Threats: Physical layer introduces many new and unexplored threats to CPS. Sensors need a mechanism that prevents them from leaking critical data to an attacker. CPS models can be responsible for the life of human beings as well so at this level threats should be handled carefully.

2. The second layer is the network layer where the real-time communication of the CPS is handled through different kind of networks.

Threats: In this layer the confidentiality, integrity and authenticity must be secured.

This can be achieved through point-to-point or end-to-end encryption [1]. The communication is also vulnerable to sink nodes and many other routing attacks that must be prevented through a complete security requirements engineering.

3. The third layer is the application layer. It is the main part of the system. Every bit of collected information comes together in this layer of the CPS. It controls the CPS and is the part of the system which combines the sensors and other nodes to an intelligent system.

Threats: It controls the data flow and also contains a great amount of data, for example patient health information. These data must be secured through a strong encryption. Experts have to make sure that the measured data are valid. Otherwise, an attacker can exploit this and endanger human life or trigger weak points in the system. This can be done by redundant sensors or software that evaluates anomalies in the system

4. Comparison between SREP and CPS-SREM

This section conducts a comparison between SREP and CPS-SREM by implementing these two security requirements engineering methods in an extended version of the "Medical Video Chat" application. The case study centers on the initial stages of the process. A process flow diagram is illustrated below in figure 5. In this section a case study for the standard SREP is presented. Then in contrast to this study, a case study with the CPS SREM is discussed and compared.

4.1. Case Study

To compare SREP and CPS-SREM models, a medical application is used. A precise comparison is shown using all activities (A1-A11).

Activity 1: SREP (Agree on definitions) - In this step the involved parties must agree on a common set of definitions. Both parties must ensure that everyone knows what these definitions mean.

CPS − SREM: (Agree on definitions and gather security goals) - In the initial phase, stakeholders and security experts collaborate to establish definitions. This step closely resembles the one found in SREP but with a significant modification. It involves a meeting where both experts and stakeholders contribute their perspectives to address issues effectively. This approach transforms CPS-SREM into a multilateral framework that acknowledges the diverse opinions of stakeholders and seamlessly integrates them into the project. Subsequently, the experts and stakeholders proceed to define security objectives and goals for all three layers of CPS.

Activity 2: SREP (Identify vulnerable and/or critical assets) - Assets that are vulnerable or critical to protect within the system are identified. Therefore, extended interviews with the stakeholders are conducted and the functional requirements are investigated. In this example, the information that the patient brings into this system, whether it is personal information or data about health are the most valuable data that are transmitted.

CPS-SREM: (Identify vulnerable and/or critical assets) - CPS-SREM offers an increased probability of identifying CPS-specific vulnerable assets, whether through the specialized expertise of the physical layer expert, the establishment of security goals, or the insights gathered from previously developed CPS stored within the Security Resource Repository (SRR).

Activity 3: SREP (Identify security objectives and dependencies) - From the previously uncovered assets, the security objectives are now developed or retrieved from the SRR. The following Security Objectives (SO) can be revised over the next iterations. The security level of the objectives will also be associated to the objectives.

SO1: Unauthorized access to personal or healthcare information - High
SO2: Disrupting the communication – Medium
SO3: Availability of the healthcare data for the doctor – High
SO4: Encryption of the data stored in the application – Medium
SO5: End-to-end encryption of the communication channels – High
SO6: Authenticate the doctor and the patient - Medium
CPS-SREM: (Identify security objectives and dependencies) - This step is similar to the original SREP step. Like in any other activity the enhanced SRR and the security expert can produce a more detailed analysis for CPS. In this scenario, another Security Objective SO7 is also considered:
* SO7: Provision of a second sensor to detect disagreements within the other sensors – Medium

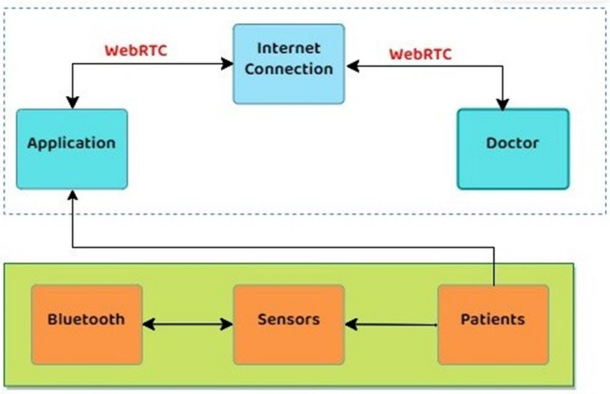


Figure 5. Process Flow Diagram of Medical Application

Activity 4: SREP (Identify threats and develop artefacts)-Artefacts are developed to identify the threats to the different assets and security objectives. It is also important to discover the threats that are not linked to any of the previous assets but endanger the security of the system. This artefact helps develop the threats that can hurt the system. From this attack tree, three threats can be received easily. The authentication/authorization, unsecure communication and possible bugs/exploits are the main threat within this system.

Threat 1: Unsecure or weak encrypted communication. Communication between the sensors or the patient and doctor.
Threat 2: Authorization issues. An unauthorized person could get access to personal information or could pretend to be the doctor.
Threat 3: False information inside the network because of an attacker.
Threat 4: Possible bug in the software that leads to an exploit in the encryption/authorization or communication.

CPS SREM (Identify threats and develop artefacts) - This activity contains some simple differences to the SREP. The CPS-SREM suggests using UMLSec and attack trees to detect threats. The attack tree from the SREP case study can be used at this point. The model would be the same. While misuse cases and use case diagrams can be incorporated for added depth, they are not mandatory. The threats that have been found are rated with the frequency the threats may appear. This research proposes a special threat rating Table 2. After that, the useful artefacts are written to the SRR. This study uses an additional UMLSec model for the identification of threats. Figure 6 shows UMLsec diagram of CPS model. When one uses the developed threat model, the secure links constraint is violated. The system does not contain secrecy for the «Internet» and «Bluetooth» communication. These violations result in Threat 1. To establish secrecy between these nodes they must be encrypted. Also, when there is no encryption in the communication and in the data storage, a lost mobile phone can result in another dangerous threat (Threat 6). All threats are defined in the list below:

Threat 1: Unsecure or weak encrypted communication - Communication between the sensors or the patient and doctor.
Threat 2: Authorization issues - An unauthorised person could get access to personal information or could pretend to be the doctor.
Threat 3: False information inside the network because of an attacker.
Threat 4: Possible bug in the software that leads to an exploit in the encryption/authorisation or communication.
Threat 5: False/no information from a sensor because of a cyber or physical attack on the sensor.
Threat 6: Loss of the mobile phone leads to discovering of the private healthcare data.

Activity 5: CPS-SREM (Domain analysis and vulnerability identification) - This activity was not defined in simple SREP model. This activity particularly emphasis on threats, assets, and vulnerabilities originating from the physical layer. The analysis predominantly centers around two significant components of CPS: the network and sensors. Sensors and other nodes play a vital role in CPS, as they are responsible for measuring and monitoring the physical world. Consequently, this phase benefits in the development of artifacts and the acquisition of security requirements for the CPS model. Everything that is discovered is compared to the previously found objects and if something does not exist in the SRR, it is added. Numerous threats and vulnerabilities have been identified concerning the network layer.

Table 2. Threat cps() adversary

Stereotype	Threats
Internet	{delete, read, insert}
Bluetooth	{delete, read, insert}
encrypted	{delete}
LAN	∅
wire	∅
sensor	{delete, read, insert}

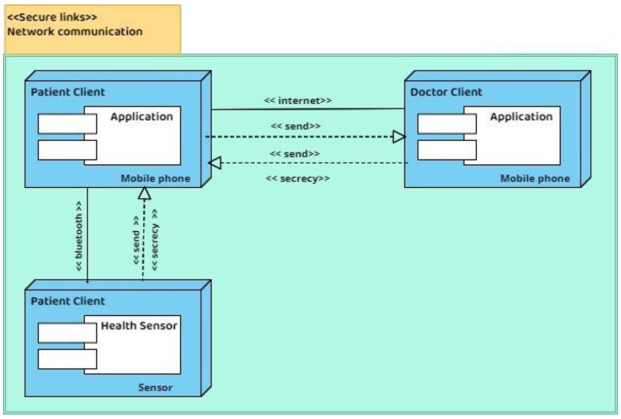


Figure 6. CPS model for Medical Application

However, there are also specific requirements that pertain to the sensors and the network layer, making them distinct and significant considerations.

1. SR: Employing redundant sensors to enhance fault tolerance.
2. SR: Ensuring that physical nodes do not leak data to non-authenticated nodes, even in the face of situations like vandalism.
3. SR: Reconnaissance data from redundant sensors can be used to identify false information inserted into the system to cause attacks.

Activity 6: SREP (Risk Assessment) - After discovering the threats, frequency and impact of them will be determines. The authors of SREP therefore used MARGERIT, a Spanish risk management approach. It is based on building artefact tables with threats, attacks and risks to determine the risk and impact of the previously discovered threats. Table 3 summarizes the risks and impact of the threats.

CPS SREM (Risk Assessment) - In this research, a robust CPS model is proposed which can be used for different cases. To achieve this, this proposed method integrates perspectives from the CPS framework by Peng, the NIST CPS framework, and NIST risk management principles [10–12].

Table 3. Table of Threats

Threat	**Impact**	**Attack**	**Probability**	**Risk**
Unsecure communication	HIGH- sniffing and change of information possible	attack on the weak encryption standard	HIGH	HIGH
Authorisation issue	HIGH - receive admin/ doctor rights	spam mail or weak attackable admin password	HIGH	HIGH
	LOW - if the attacker gains no admin/doctorrights	spam mail for password spoofing	HIGH	LOW
False information	LOW - if the system detects this information	external or internal access to the network	LOW	LOW
	HIGH - if it does not		HIGH	HIGH
Software bug encryption/ authorisation	LOW - if it causes asystem crash	battery drain in the mobile phone	LOW	LOW
	HIGH - if the attack gains access to communication or stored information	bug in encryption implementation	LOW	HIGH

For Example, the initial two phases of the NIST framework are applied to this case study as follows:
1. Categorize [9] - Information: Private patient data and healthcare information
* Integrity: HIGH
* Confidentiality: HIGH
* Availability: LOW
2. Select [9] - In this case study, the system is categorized and selected a high-security priority baseline from the extensive catalog of security controls provided by NIST.

Activity 7: CPS-SREM (Develop Patch Management) - This activity was not done in Standard SREP. In this case study, the necessity for a robust patch management system is evident to ensure high security in the coming operational years. Consequently, the application can be updated through various application store distributors. The old app version should not allow to connect to a doctor and Bluetooth sensors are updated that connect end-to-end encrypted with the secured distribution server. At this point, security requirements are defined specifically for this patch management system which are given below:
1. SR01: Deploying updates for both the application and the sensors is possible.
2. SR02: The deployment is carried out using an encrypted connection.
3. SR03: Data and video connections can only be established by the current versions.
4. SR04: Updates are required to undergo a standardized quality assurance (QA) process.
5. SR05: Every update is required to be signed for authorization.

Activity 8: SREP (Elicit Security Requirements) - After gathering the threats, assets, vulnerabilities and security objectives, elicit security requirements can be defined. If this project is not the first one, the saved data within the SRR could help discover security requirements.
1. SR01: The communication must be end-to-end encrypted.
2. SR02: The patient and the doctor must authorise themselves in a secure way.
3. SR03: The system has to filter false sensor information.
4. SR04: It is better to avoid security vulnerabilities that pose a risk to patient health.
5. SR05: The communication to the doctor must be robust without a possible loss of information.
6. SR06: No external server should be used for the data transfer and communication.
7. SR07: The healthcare date on the mobile phone must be stored encrypted.

CPS SREM (Elicit Security Requirements) - This step remains almost the same as defined in the SREP but it gets more input due to the additional steps and the physical layer expert. Eliciting security requirements of the CPS-SREM case study are the same until SR07 (as shown in SREP), and it also includes the SR01-SR05 from the "Develop Patch Management" activity of the CPS-SREM.

Activity 9: SREP (Categorize and prioritize requirements) - The developed security requirements are now rated based on the likelihood of the threats and the impact to the system. The categorization determines when the requirements are processed as follows:
1. SR01 SR02
2. SR07
3. SR06
4. SR05
5. SR03 SR04

CPS SREM (Categorize and prioritize requirements)

The security requirements of CPS-SREM are rated using the same procedure as in the standard SREP with slight additions as follows:
1. SR01 SR02 SR08
2. SR07 SR09 SR12
3. SR05 SR06 SR11 SR14
4. SR13 SR10 SR15
5. SR03 SR04

Activity 10: SREP (Requirements inspection) - The generated requirements and artefacts such as attack trees, UMLSec models, use cases, etc. are inspected and validated to ensure that they meet the organisation’s own requirements. Also, the requirements are checked based on the IEEE standard for requirements.

CPS SREM (Requirements inspection) - The CPS-SREM step is almost same as the SREP model. The artifacts and requirements are also checked and validated in accordance with the updated IEEE

29148:2011 standard. This activity produces the “Validation Report”.

Activity 11: SREP (Repository improvement) - The models created in the previous steps are added to the Security Resource Repository if they are considered helpful for further projects. Outdated models are deleted.

CPS SREM (Repository improvement) - This step is much shorter than the SREP step because the artefacts are already added in the SRR within different previous activities. The experts assess the newly added artifacts for their relevance to future projects, and they remove any outdated fragments from the documentation.

5. Comparison of SRE methods and Discussion

A brief comparison of different security requirement methods is presented in this section. Table 4 highlights the key criteria for CPS-SRE methods. It’s essential to note that, unlike other methods, UMLSec primarily focuses on the architectural layer. It can be integrated into other methods to generate artifacts and establish an architectural understanding of the domain. Nowadays, a modern SRE method must be comprehensive, involving a step where project leaders and stakeholders’ perspectives are integrated into the security SRE process. The outdated unilateral approach has been replaced with a more collaborative one [8]. In the CPS-SREM, this collaborative analysis and consensus-building process with stakeholders takes place as an activity in step 1. Risk management has gained a growing significance in the development of secure software, particularly in the context of cyber-physical systems. The risk assessment process frequently identifies potential vulnerabilities of the process. This activity is included in CPS-SREM model.

Table 4. Comparison of SRE Methodologies

	CLASP	SQUARE	UMLSec	SREP	CPS SREM
Multilateral View	-	x	-	-	x
CIA Goals	x	x	x	x	x
Risk Management	x	x	-	x	x
Software View	x	x	x	x	x
Domain View	-	-	x	-	x
Patch Management	-	-	-	-	x
CPS SREP Steps					
Agree on Definitions and Gather Security Goals	x	x	-	x	x
Identify Vulnerable and/or Critical Assets	x	x	-	x	x
Identify Security Objectives and dependencies	x	x	-	x	x
Identify Threats and Develop Artefacts	x	x	-	x	x
Analysis of the Domain/ Vulnerability identification	-	-	-	-	x
Risk Assessment	x	x	-	x	x
Develop Patch Management	-	-	-	-	x
Elicit Security Requirements	x	x	-	x	x
Categorize Prioritize Requirements	-	x	-	x	x
Requirement Inspection	x	x	-	x	x
Repository Improvement	-	-	-	x	x

The development of cyber-physical systems is an important step in developing the entire model. The domain plays a crucial role in CPS, and the fusion of software, physical layer, and human interaction introduces a multitude of new security requirements. This is why modern SRE methodologies should include a dedicated step for analyzing the environment in which the CPS operates. CPS-SREM meets this demand. Another crucial aspect of CPS-SREM is the inclusion of a step for planning patch management post-deployment phase. This needs to be done with the regular secure CPS process. To guarantee this, CPS-SREM incorporates its own patch management activity. It can be observed from Table 4 that compares different SRE methods. The initial section involves the comparison of several key features. It is analyzed that unlike other methods, CPS-SREM fulfils every feature. The essential features required for establishing a secure CPS are not part of the older methods. In summary, older SRE methods are well-suited for software environments but are not for the development of CPS or IoT systems. When it comes to CPS, SRE methods specifically designed for cyber-physical systems are required. In the future, more comprehensive studies about CPS security and CPS-SRE are required. The healthcare sector in particular has a lot of its own regulations and requirements that need to be fulfilled. Another important game changer can be the 5G network in the future. 5G is developed with the extended number of devices and big bandwidth need in mind. But the functional feature of this network is not the only issue that has to be considered. This needs a good multi-level security model.

6. Conclusion

Cyber-physical systems are an emerging field of technology but face a lot of challenges. Security is one of the main challenges for CPS. Therefore, the main contribution of this research is to propose a security requirements engineering methodology that can handle the security requirements of cyber-physical systems throughout the entire software development lifecycle. Through a comparative analysis with existing state-of-the-art SRE methods, the study illustrates that these methods are inadequate to handle the challenges presented by CPS method. It is shown that the proposed method CPS-SREM can handle the complexity of CPS and allows experts to discover more CPS-related security requirements.These newly developed requirements are discovered mainly in two completely new steps of the CPS-SREM. Furthermore, the proposed CPS-SREM is validated through a healthcare system case study, illustrating its adaptability for various new methods with minimal or no modifications. The proposed model will significantly benefit both practitioners and researchers by aiding them in identifying the security requirements for CPS in Industry 4.0.

References

[1] Yaacoub, J., Salman, O., Noura, H., Kaaniche, N., Chehab, A. & Malli, M. Cyber-physical systems security: Limitations, issues and future trends. Microprocessors And Microsystems. 77 pp. 103201 (2020)

[2] Ding, D., Han, Q., Xiang, Y., Ge, X. & Zhang, X. A survey on security control and attack detection for industrial cyber-physical systems. Neurocomputing. 275 pp. 1674-1683 (2018)

[3] Mellado, D., Fernández-Medina, E. & Piattini, M. Applying a security requirements engineering process. Computer Security–ESORICS 2006: 11th European Symposium On Research In Computer Security, Hamburg, Germany, September 18-20, 2006. Proceedings 11. pp. 192-206 (2006)

[4] Schneble, W. & Thamilarasu, G. Attack detection using federated learning in medical cyber-physical systems. Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN). 29 pp. 1-8 (2019)

[5] Muñante, D., Chiprianov, V., Gallon, L. & Aniorté, P. A review of security requirements engineering methods with respect to risk analysis and model-driven engineering. Availability, Reliability, And Security In Information Systems: IFIP WG 8.4, 8.9, TC 5 International Cross-Domain Conference, CD-ARES 2014 And 4th International Workshop On Security And Cognitive Informatics For Homeland Defense, SeCIHD 2014, Fribourg, Switzerland, September 8-12, 2014. Proceedings 9. pp. 79-93 (2014)

[6] Sirohi, P., Agarwal, A. & Tyagi, S. A comprehensive study on security attacks on SSL/TLS protocol. 2016 2nd International Conference On Next Generation Computing Technologies (NGCT). pp. 893-898 (2016)

[7] Gao, Y., Peng, Y., Xie, F., Zhao, W., Wang, D., Han, X., Lu, T. & Li, Z. Analysis of security threats and vulnerability for cyber-physical systems. Proceedings Of 2013 3rd International Conference On Computer Science And Network Technology. pp. 50-55 (2013)

[8] Fabian, B., Gürses, S., Heisel, M., Santen, T. & Schmidt, H. A comparison of security requirements engineering methods. Requirements Engineering. 15 pp. 7-40 (2010)

[9] Ross, R. & Johnson, L. Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach. (Ronald S. Ross, L A. Johnson, 2010)

[10] Griffor, E., Greer, C., Wollman, D. & Burns, M. Framework for cyber-physical systems: Volume 2, working group reports. (Edward R. Griffor, Christopher Greer, David A. Wollman, Martin J. Burns,2017)

[11] Force, J. Risk management framework for information systems and organizations. NIST Special Publication. 800 pp. 37 (2018)

[12] Peng, Y., Lu, T., Liu, J., Gao, Y., Guo, X. & Xie, F. Cyber-physical system risk assessment. 2013 Ninth International Conference On Intelligent Information Hiding And Multimedia Signal Processing. pp. 442-447 (2013)

[13] Rehman, S., Allgaier, C. & Gruhn, V. Security requirements engineering: A framework for cyber-physical systems. 2018 International Conference On Frontiers Of Information Technology (FIT). pp. 315-320 (2018)

[14] Rehman, S. & Gruhn, V. An effective security requirements engineering framework for cyber-physical systems. Technologies. 6, 65 (2018)

[15] Japs, S. Security safety by model-based requirements engineering. 2020 IEEE 28th International Requirements Engineering Conference (RE). pp. 422-427 (2020)

[16] Asplund, F., McDermid, J., Oates, R. & Roberts, J. Rapid integration of CPS security and safety. IEEE Embedded Systems Letters. 11, 111-114 (2018)

[17] Jürjens, J. Towards development of secure systems using UMLsec. International Conference On Fundamental Approaches To Software Engineering. pp. 187-200 (2001

[18] Jürjens, J. UMLsec: Extending UML for secure systems development. International Conference On The Unified Modeling Language. pp. 412-425 (2002)

[19] 19. Best, B., Jurjens, J. & Nuseibeh, B. Model-based security engineering of distributed information systems using UMLsec. 29th International Conference On Software Engineering (ICSE'07). pp. 581-590 (2007)

[20] Jürjens, J. & Shabalin, P. Automated verification of UMLsec models for security requirements. International Conference On The Unified Modeling Language. pp. 365-379 (2004)

[21] Ruiz, J., Arjona, M., Maña, A. & Rudolph, C. Security knowledge representation artifacts for creating secure IT systems. Computers Security. 64 pp. 69-91 (2017)

[22] Gregoire, J., Buyens, K., De Win, B., Scandariato, R. & Joosen, W. On the secure software development process: CLASP and SDL compared. Third International Workshop On Software Engineering For Secure Systems (SESS'07: ICSE Workshops 2007). pp. 1-1 (2007)

[23] Ansari, M., Pandey, D. & Alenezi, M. STORE: Security threat-oriented requirements engineering methodology. Journal Of King Saud University-Computer And Information Sciences. 34, 191-203 (2022)

[24] Khan, R., Khan, S., Khan, H. & Ilyas, M. Systematic mapping study on security approaches in secure software engineering. Ieee Access. 9 pp. 19139-19160 (2021)

[25] Mead, N. & Stehney, T. Security quality requirements engineering (SQUARE) methodology. ACM SIGSOFT Software Engineering Notes. 30, 1-7 (2005)

[26] Suleiman, H. & Svetinovic, D. Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: a case study using smart grid advanced metering infrastructure. Requirements Engineering. 18 pp. 251-279 (2013)

[27] Viega, J. Building security requirements with CLASP. ACM SIGSOFT Software Engineering Notes. 30, 1-7 (2005) 631

[28] Dey, N., Ashour, A., Shi, F., Fong, S. & Tavares, J. Medical cyber-physical systems: A survey. Journal Of Medical Systems. 42 pp. 1-13 (2018).

Dr. Shafiq ur Rehman received the MS degree in Computer Science from Dresden University of Technology, Dresden, Germany and Ph.D. degree in Computer Science from the Department of Software Engineering, Duisburg-Essen University, Germany in 2020. He is an assistant professor at the College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh KSA. He also worked as a consultant (Requirements Engineer) in a well renowned international organizations in Germany. He has published several research papers in high-ranked international conferences and ISI indexed journals. He is involved in different international funded projects in the field of cyber-physical systems and cybersecurity. His research interests include AI, cyber-physical systems, cybersecurity and requirements engineering.