Computers in Industry 132 (2021) 103524

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

CARMEN: A framework for the verification and diagnosis of the specification of security requirements in cyber-physical systems

Ángel Jesús Varela-Vaca [a,*], David G. Rosado [b], Luis E. Sánchez [b], María Teresa Gómez-López [a], Rafael M. Gasca [a], Eduardo Fernández-Medina [b]

[a] Universidad de Sevilla, Escuela Técnica Superior de Ingeniería Informática, Dpto. Lenguajes y Sistemas Informáticos, Sevilla, Spain
[b] Dpto. Tecnologías y Sistemas de Información, Universidad Castilla-La Mancha, Ciudad Real, Spain

ARTICLE INFO

Article history:
Received 16 April 2021
Received in revised form 9 July 2021
Accepted 14 July 2021
Available online 27 July 2021

Keywords:
Cyber-physical system
Cybersecurity
Security
Configuration models
Security requirements
Security verification
Diagnosis

ABSTRACT

In the last years, cyber-physical systems (CPS) are receiving substantial mainstream attention especially in industrial environments, but this popularity has been accompanied by serious security challenges. A CPS is a complex system that includes hardware and software components, with different suppliers and connection protocols, forcing complex data management and storage. For this reason, the construction, verification and diagnosis of security CPS become a major challenge, which involves a correct specification of security requirements, the verification of the correct system configurations, and if necessary, the diagnosis to detect the features to be modified to obtain a security configuration. In this paper, we propose a framework for the verification and diagnosis of security requirements, according to the possible correct configurations of the CPS. The framework is based on the specification of the security requirements and their analysis supported by Model-Driven Engineering and Software Product Line Engineering (SPLE) approaches. To illustrate the usefulness, the proposal has been applied to the security requirements in an Agriculture 4.0 scenario based on automated hydroponic cultivation.

1. Introduction

Nowadays, cyber-physical systems (CPS) are drawn the attention within the industry, society and government, due to the enormous impact they have on the economy and the environment (Mörth et al., 2020), and providing citizens and businesses with a wide range of innovative applications and services (Colombo et al., 2020).

The entry of the CPS in the industry allows high connectivity between the industrial systems and brought great advantages and a wide range of new opportunities to industries but also some questions and problems, such as those related to safety and security (Mokalled et al., 2019).

Cybersecurity is a fundamental discipline that provides confidence in terms that CPS, their information, and supporting communications and information infrastructures are adequately safeguarded. CPS have many unique characteristics, including the need for real-time response and extremely high availability, predictability, and reliability, which impacts cybersecurity decisions (Ashibani and Mahmoud, 2017). Besides, an even more critical problem is that the development of these systems has been carried out without taking into account the security aspects, nor the new risks that this automation of processes implies, which put at risk the complete industrial infrastructure (Lezzi et al., 2018), and where any security breaches to these systems could have catastrophic consequences (Yaacoub et al., 2020). Taking into account the security from the earliest steps of CPS, i.e., at the design time is crucial to avoid security issues, even though it is a very challenging task (Geismann et al., 2018; Peisert et al., 2014).

Therefore, security within industrial environments becomes a critical aspect that must be taken into account at all stages of information system development, by obtaining and defining, from early analysis and security requirements (Zunino et al., 2020) before the system is in place (Souag et al., 2015). Security and safety are nonetheless two key properties of CPS (Yaacoub et al., 2020) and they share the same goal, protecting CPS from failures (Pirbhulal et al., 2021). Security and safety refer to different but very important properties (Avižienis et al., 2004), in this paper just focus on those related to security properties for the CPS.

* Corresponding author.
E-mail addresses: ajvarela@us.es (Á.J. Varela-Vaca), david.grosado@uclm.es (D.G. Rosado), luise.sanchez@uclm.es (L.E. Sánchez), maytegomez@us.es (M.T. Gómez-López), gasca@us.es (R.M. Gasca), eduardo.fdezmedina@uclm.es (E. Fernández-Medina).

https://doi.org/10.1016/j.compind.2021.103524

Although security requirements are the appropriate solution for many researchers, they are difficult to obtain, analyse and manage by their subjective nature and their description in natural language. For CPS, in addition to software security requirements, we also have physical, control and communication requirements, which make the task of identifying security requirements and translating them into the design of our CPS system even more complicated (Kim and Lee, 2020). Therefore, having a common model (or metamodel) is essential, since it facilitates the definition of security requirements for CPS where any particularity of all these elements is taken into account.

The high variability of the components involved in a CPS and their possible configurations make it extremely difficult to verify the correctness of the security requirements that reduce the threats and the possible risks (Varela-Vaca et al., 2020b). To manage this complexity, we propose the use of Feature Models (FMs) (Galindo et al., 2019a) and a set of reasoning techniques (Benavides et al., 2007) to verify the correctness of the security requirements and diagnose misconfiguration of the features (Riel et al., 2018), according to a catalogue of possible correct configurations. FMs have been previously used for checking security configurations (Varela-Vaca et al., 2020a), and the diagnosis of FM configurations is a studied problem by the community (Varela-Vaca et al., 2019a). However, how it can be adapted to the specific scenario of CPS is still an open challenge (Arrieta et al., 2015) tackled in this paper.

Based on the problems identified, we have developed a CARMEN framework that presents a systematic process to enable from the description of security requirements to the verification and diagnosis for CPS through variability models. CARMEN is focused on the design phase of CPS by presenting a support system for guiding the whole security requirement life-cycle: (1) creating a metamodel which enables the definition of security requirements for CPS based on security recommendations of ENISA (Baseline, 2018) and OWASP (OWASP, 2021) guidelines; (2) load and update a variability model that encompassed the catalogue of possible correct configurations for CPS; (3) map both security requirements and variability model resulting in a configuration; (4) verify the correctness of the configuration, and; (4) if it is not correct, the diagnosis of the configuration to modify for achieving a correct configuration according to requirements. To explain in detail each of these steps, the paper has been organised as follows: Section 2 reviews the most relevant papers in the area. Section 3 details a case study based on Agriculture 4.0 and introduces the possible cyber-risks to which this type of systems are subject. Section 4 presents the proposed framework for the diagnosis of this type of system. Section 5 applies the proposed framework to the case study to show its applicability; and, finally, conclusions are drawn, and future work is outlined in Section 6.

2. Related work

Related works have been divided into the three areas of research addressed in the article: how feature model analysis has been used in the security field; how security requirements and ontologies can be used for the modelling of risk scenarios, and; an analysis of the standards and guidelines that have been found related to CPS/IoT.

2.1. Cyber-security and feature model analysis

Feature-Oriented Domain Analysis (FODA) have become mature fields in the Software Product Line (SPL) arena in the last decades (Benavides et al., 2010). Several are the scenarios where SPLs based on feature model analysis have been applied (Galindo et al., 2019b; Varela-Vaca et al., 2019b), and different researchers highlight the advantages of these systems since the use of Model-Driven Engineering (MDE) methodology and the SPL paradigm is becoming increasingly important (Iglesias et al., 2019). The complexity and the high variability of a CPS, and how SPL can help were analysed in Arrieta et al. (2015) and Beek et al. (2018), detecting the points of variability using feature model analysis. The analysis of the variability of CPS can also support the testing (Arrieta et al., 2016).

Security is an understudied field in SPL area. Different approaches have been presented to manage the variability and specify security requirements from the early stages of the product line development (Mellado et al., 2014). Similarly, other approaches addressed the idea of including the security variability into an SPL (Sion et al., 2016). In Fægri and Hallsteinsen (2006), the authors established a software architecture as a reference to develop SPL, dealing with information security aspects. SPLs are currently being targeted for application in CPS, as for some researchers, no standard provides a structured co-engineering process to facilitate the communication between security engineers (Bramberger et al., 2020). For other researchers, information security must be a top priority when engineering C-CPS as the engineering artefacts represent assets of high value, and the research is focused on the generation of new security requirements stemming from risks introduced by CPS (Biffl et al., 2019).

On the other hand, there are approaches focused on security as a use case, such as in Arciniegas et al. (2006) and the methodology SecPL (Peldszus et al., 2018), where is highlighted the importance of specifying the security requirements and product-line variability. These are annotated in the design model of any system. Other researchers developed a security requirements engineering framework for CPS, which is an extension of SREP (ur Rehman et al., 2018). The capacity to support the high variability in the security context though Feature Models appeared in previous papers (Kenner et al., 2020), where the authors analysed which vulnerabilities could be used to simulate attack-defence scenarios, but these simulations were not oriented towards more complex scenarios, such as cyber-physical systems.

2.2. Ontologies and security requirements for cybersecurity

As seen in the introduction, nowadays CPS require an adequate security configuration (Ashibani and Mahmoud, 2017). Therefore, some researchers are focused on the development of ontologies and security requirements (Shaaban et al., 2019a; Span et al., 2018). Some researchers have developed security tools based on ontologies capable of being integrated with the initial stages of the development process of critical systems (Shaaban et al., 2019b). On the other hand, requirements have been analysed not only from the software side but also from the hardware perspective, including sensors and network security. Therefore they propose the development of a security requirements framework for CPS, analysing the existing ones, and concluding that currently there is no suitable requirement framework for this type of systems (Rehman et al., 2018; Rehman and Gruhn, 2018). Other researchers consider that CPS have unique characteristics that limit the applicability and suitability of traditional cybersecurity techniques and strategies, and therefore propose the development of a methodology of cybersecurity requirements oriented towards weapons systems (Carter et al., 2019). This methodology allows us to discover solutions that improve dimensions (such as security, efficiency, safety, performance, reliability, fault tolerance and extensibility), being possible to use automated coding tools (Zhu and Sangiovanni-Vincentelli, 2018).

Therefore, we can conclude that at present different researchers have found the need to develop requirement grammars to control the security risks associated with CPS. Moreover, derived from the

complexity of the CPS, feature models have been previously used in the context of security.

2.3. Standards and guidelines for CPS/IoT

Although CPS/IoT systems have been maturing for years, there are still few official standards specialised in this type of system. All of them are based on research proposals, guidelines, or cross-cutting standards.

Some of the most relevant standards, guidelines or other proposals are as follows: (i) the Cloud Security Alliance (CSA) that has developed some IoT oriented guides such as "Future-proofing the Connected World: 13 Steps to Developing Secure IoT Products" (Group et al., 2017), the "Identity and Access Management for the Internet of Things" (Group et al., 2016b) and the "New Security Guidance for Early Adopters of the IoT" (Group et al., 2015); (ii) the GSM Association (GSMA) also developed its own IoT-oriented guide called "IoT Security Guidelines" (Group et al., 2016a); (iii) the Industrial Internet Consortium (IIC) developed the basis of an IoT-oriented cybersecurity framework called "Industrial Internet of Things Volume G4: Security Framework" (Group et al., 2016c); (iv) the European Telecommunications Standards Institute (ETSI) has developed a guide called "ETSI EN 303 645 – Cyber Security for Consumer Internet of Things: Baseline Requirements" (ETSI, 2020) and has participated in the development of a specific standard for IoT called "oneM2M – Standards for M2M and the Internet of Things" (OneM2M, 2017); (v) Microsoft has also developed its own guide for IoT called "Cybersecurity Policy For The Internet Of Things" (Abendroth and Kleiner, 2017); (vi) the NIST has defined a guide called "Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks: NISTIR 8228" (Information Technology Laboratory, 2019), an analysis on IoT called "NIST SP 800-183 – Networks of Things" (Information Technology Laboratory, 2016), and a framework called "NIST Framework for Cyber-Physical Systems" (CPS, 2016); and (vii) finally, the ISO/IEC, counts on the standard "ISO/IEC 30141:2018 – Internet of Things (IoT) – Reference Architecture" (ISO Central Secretary, 2018), as well as the guidelines currently under development such as "ISO/IEC CD 27402.2 – Cybersecurity – IoT security and privacy – Device baseline requirements" (ISO Central Secretary, 2021a) and "ISO/IEC DIS 27400 – Cybersecurity – IoT security and privacy – Guidelines" (ISO Central Secretary, 2021b).

Other organisations are also currently working on developing new proposals on the subject, such as OWASP, which has developed the "Internet of Things Top Ten", or the EC Alliance for Internet of Things Innovation (AIOTI), the IERC European Research Cluster on the Internet of Things, and the IoT Security Foundation (IoTSF), which are developing their own guidelines on IoT.

All these documents have been the basis used by ENISA to develop its own guide on IoT, which has been used to develop part of the research presented in this paper.

3. Case study

The case study presented here (see Fig. 1) is a CPS system for hydroponic farming, in which various components are involved, both hardware and software.

Hydroponic farming is controlled by the following physical elements:

- Temperature, light and humidity sensors. They measure the existing temperature, light and humidity in the environment.
- Heater, cooler, ultraviolet, nutrient injector and water pump actuators. These actuators change the characteristics of the environment by varying the temperature, the projected light, the flow of water and the injection of nutrients needed to facilitate growth and photosynthesis. All these actuators are activated or deactivated by the controller.
- Controller. It is an Arduino device and web system that receives the data from all the sensors and sends it (via wireless connections) through the web system to the Big Data system for subsequent storage, analysis and display.

In addition to the physical part, the controller is connected to a visualisation and control system with Big Data technologies where we have deployed the following components:

- **Dashboard**. It allows the users to control the hydroponic farming in real-time and to consult statistics, as well as to interact with the farming sending HTTP requests to the actuators. It is also connected to the datastore that sends the necessary data for its correct visualisation.
- **Data handler**. It is responsible for processing the sensor data, received from the controller, and storing it in the database.
- **Datastore**. It contains a Hadoop file system (HDFS) and an HBASE database where all the values coming from the sensors are saved, as well as user data, historical data, logs and all the information needed for the Big Data system to be able to automatically control and monitor changes and act accordingly to stabilise it.

4. CARMEN framework

We propose CARMEN, a framework to assist the stakeholders in the stages, from the definition of security requirements for CPS to the verification and diagnosis of them. Fig. 2 represents the step-by-step process proposed in CARMEN. Initially, it is necessary to describe the security requirements that involve the CPS components and the security aspects, that will be later verified and diagnosed. The formalisation of those requirements in the context of CPS are defined in Section 4.1. For the verification of the correctness of the defined requirements, it is necessary to provide a repository of possible correct requirement configurations according to the international standards and security best practices, that as was commented in the introduction, will be described by means of FMs. Section 4.2 provides the necessary details to know the FM proposed to cover the possible security configurations for CPS. Moreover, it is necessary the mapping the security requirement model elements and the FMs, as explains in Section 4.3. With the FM and the defined mapping, the verification of the correctness of the security requirements can be evaluated. The diagnosis must be executed when the security requirements are not correct according to the possible correct configurations, as detailed in Section 4.4.

4.1. Step 1: Define security requirements

This step aims at the definition of a set of security requirements for a CPS. The security requirements should specify aspects related to the CPS components (i.e., assets) and the security features to reach, e.g., Authentication for users. Thus, the security requirements encompassed the security features for the assets in the CPS.

Definition 1. Security requirements for a CPS. Let be Ψ a set of m security requirements $\{sr_1, sr_2, \ldots sr_m\}$ for a CPS.

$$(1)\Psi_{CPS} = \bigcup_i sr_i$$

Definition 2. A security requirement specification for a CPS. Let sr_i be a security requirement as a tuple $\langle AT, SF\rangle$ where AT represents a set of n asset types $\{at_1, at_2, \ldots, at_n\}$ from the CPS, and SF is the set of m security features $\{sf_1, sf_2, \ldots, sf_m\}$ related to those assets.

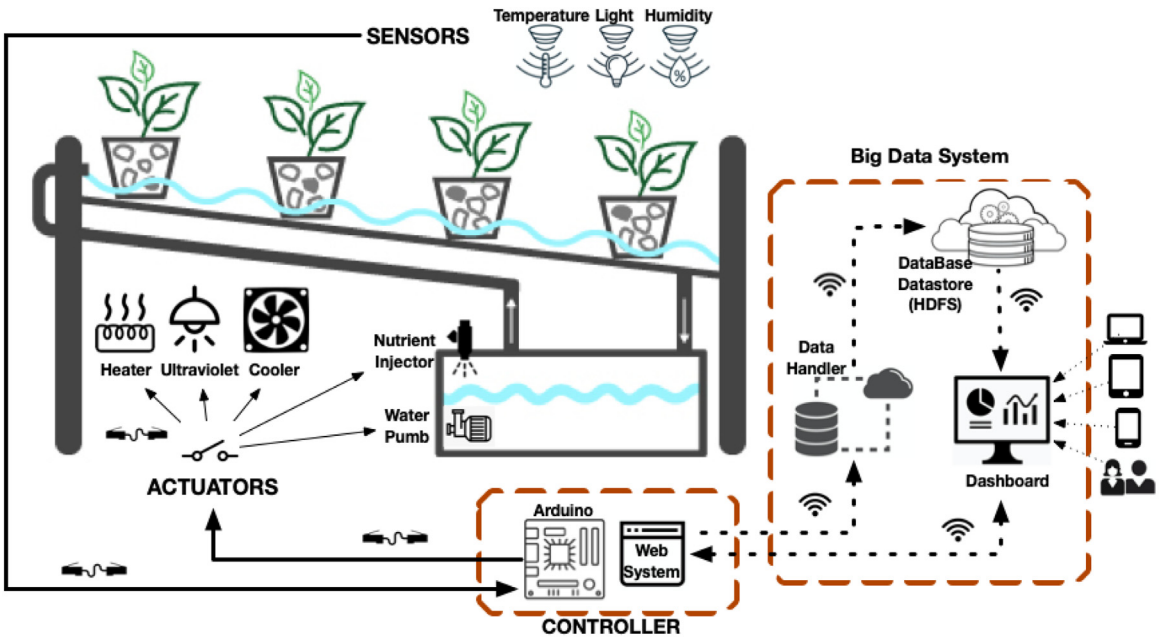

Fig. 1. Hydroponic farming.

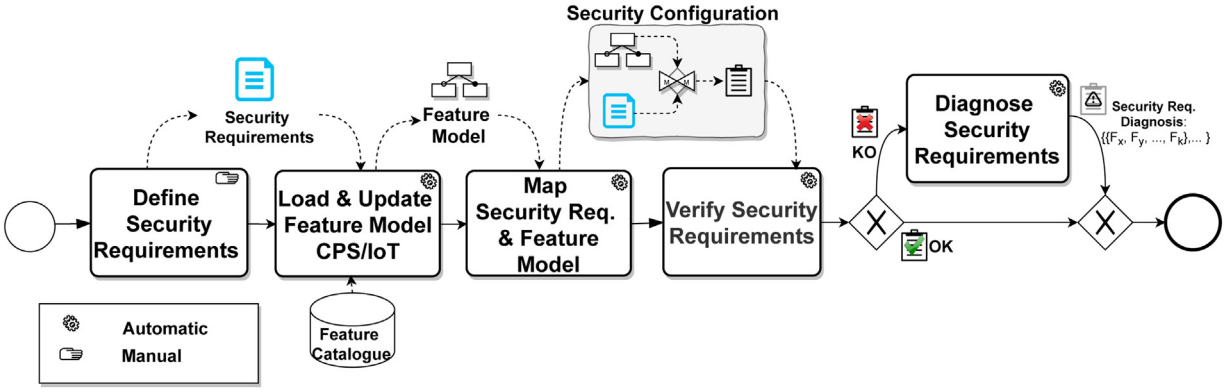

Fig. 2. CARMEN process description.

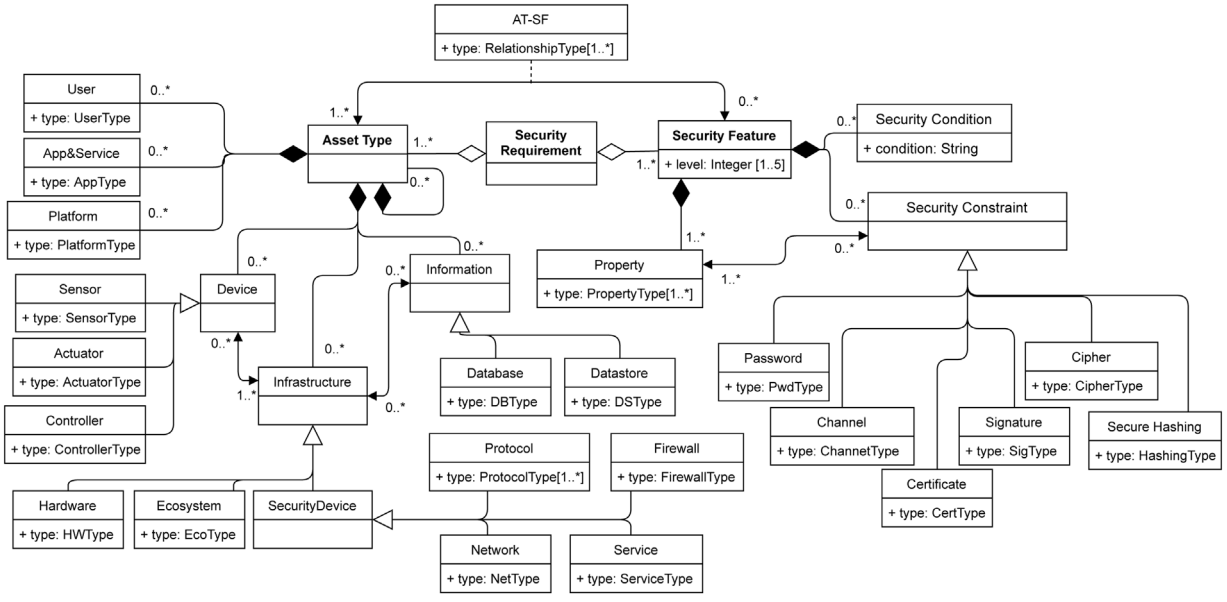

Fig. 3. View of the metamodel of security requirements for CPS.

To define *AT*, we propose to follow the security best practices for IoT in the context of critical infrastructures formulated by the ENISA agency (Baseline, 2018). Many possible *SF* for CPS are obtained from OWASP (OWASP, 2021) to extract the most important concepts (e.g., encryption, protocol, network, AES, SSL/TLS, Bluetooth, range, lifetime). The CPS design (Ashibani and Mahmoud, 2017) requires tools to model the security for the different components that are managed. However, in general, the specification of security requirements is far away from the security features supported even to be implemented. In that sense, the definition of a model (metamodel) provides many benefits, the most interesting is that it enables a reusable schema with the domain-specific semantic that can be adapted to any scenario (Brambilla et al., 2017). Therefore, we have proposed a metamodel that enables to define the concepts of *AT* and *SF* based on the ENISA and OWASP. Fig. 3 shows the proposed metamodel. As aforementioned, one of the benefits on using the metamodel is to get advantage on the adaptation capabilities to other contexts (Brambilla et al., 2017) by extending the metamodel with other dialects, for instance, to consider standards or other guidelines.

In our metamodel, a sr_i consists of two different elements: asset types (*AT*) and security features (*SF*). The *AT* is composed of any kind of asset (or set of assets) presented in a CPS system, i.e., user, application, service, platform, device, infrastructure, or information. Besides, there may be relationships between *AT*, such as between a device and the infrastructure that supports it, considering for example the type of protocol to be used by the device or the type of network established for communication (i.e., Bluetooth sensor in PAN network). There is also a relationship between the type of infrastructure and the information that is generated and transported, considering how the information flows through the communication channels and where it passes through (e.g., temperature data is transported over a WAN network to the database).

The *SF* represents the security aspects identified for the requirements related to the assets. Thus, we can define what security restrictions are involved for the assets (e.g., the type of encryption for information or communication assets, whether a two-factor (2FA) mechanism is used for passwords, or whether digital certificates are used), which property or properties are defined in the requirement, as well as defining a series of conditions that must be taken into account to satisfy the security requirement (for instance, bandwidth, memory capacity or response time). A security level is also defined as varying from very low (1) to very high (5). This security level can be used in many directions for instance to prioritise security requirements or to establish more or less strong restrictions or mechanisms. Some of these elements of the *SF* have relations between them because the values of some can restrict the values in others. For example, when the security property refers to Confidentiality or Integrity, a restriction must be defined concerning the type of encryption, or in the case of Authentication, a password policy must be defined.

Despite the relations between different assets, there is a many-to-many relation between any *AT* and the *SF*s, which has defined a type of relationship that depends on the property of the feature and the set of related assets. Thus, the metamodel allows us to have several sets of *AT* and each one of them to be linked with different, or the same, *SF* (constraint, property or condition) with different types of relationship. For example, we can establish that the user (asset) access is controlled by a 2FA authentication, linking to the authentication and the password policy features, and has another set of assets such as wireless communications that are linked to the security features of confidentiality and some type of encryption. These relationships between asset types and security features can have various types of relationships, such as secure communication and user authentication.

Table 1 shows the possible values of the attributes of our metamodel. These values have been defined of an initial effort to cover the maximum number of CPS components based on asset taxonomy of ENISA and OWASP.

The description of any sr_i is based on an instantiation of the metamodel. An example for the case study of hydroponic farming is given in Fig. 4. The sr_1 establishes that the wireless communication between the temperature sensor and Arduino as a micro-controller must be encrypted, ensuring high confidentiality. Moreover, the requirement establishes a high-level security feature and is that communication is encrypted in compliance with confidentiality. To this end, we define the property of Confidentiality, which is implicitly associated with a type of encryption (Camellia) and a secure communication channel (HTTPS). This feature is defined for a set of assets that is the temperature sensor that communicates with the Arduino through a WLAN network whose wireless communication can be BLE or RFID. This relationship between this security feature and the set of assets is defined as “secure communication”.

4.2. Step 2: Load and update of feature model for CPS/IoT

The metamodel enables to instantiate different security requirements but they cannot be validated with regard the great variability of security features. To manage the variability on the security features for the assets of a CPS, we propose to use Software the Product Line (SPL) (Benavides et al., 2010) approach including variability models, such as FMs (Galindo et al., 2019a). Therefore, we propose to use a variability model in the form of FMs which gathers all the necessary information to enable the reasoning on the correctness of the security configuration derived from the security requirements.

In this step, to reduce the effort analysing the security requirements (Ψ_{CPS}) and their possible correct configuration, we propose loading and updating the FMs for a predesigned catalogue described in (Varela-Vaca et al., 2020b). Thus, these FMs are used as the core model for the verification but they can be updated considering other relations or restrictions that are not in the model, e.g., a new security constraint related to a new asset. The use of FMs (Benavides et al., 2010) is a wide broad technique for analysing and reasoning on them. FM is a model which represents a set of products but defined by their features and their relationships. A feature is a characteristic of the systems that can be configured, for instance, to choose the protocol of communication from a bunch of alternatives.

There exist several approaches to define and formalise FMs (Batory, 2005), although the most used notation is the proposed by Czarnecki et al. (2004). In this paper, a FM is defined as following.

Definition 3. Feature model. Let *FM* be a variability model which consists of a tuple (*F,R*), where *F* is a set of *n* features $\{f_1, f_2, \cdots, f_n\}$, and *R* is a set of relations among features $\{r_1, r_2, \cdots, r_m\}$.

Fig. 5 illustrates through an example of CPS some *AT*s and *SF*s related among them. In general, FM diagrams are composed of six types of relations (Czarnecki et al., 2004) (i.e., mandatory, optional, alternative, Or-relations, alternatives, require, exclude) between a parent feature and its child features, although there exist extensions that enable attributes and extra-functionalities for features.

The automated analysis of FMs, the so-called Feature-Oriented Domain Analysis (FODA), can be achieved by formal methods (Benavides et al., 2010). Most of the approaches in the literature make a transformation from the FMs to a formal model (Dechter, 2003). This analysis of FMs enables the performance of different reasoning operations on them, e.g., to determine whether the model is valid or not, to obtain all possible configurations, or to ascertain whether it is correct or not concerning the model and based on a configuration. Thus, we can verify a concrete configuration according to an FM, if the FM is formalised and adapted to the

Table 1
Values for data types of our metamodel.

Type	Values
UserType	Consumer, provider, process, third-party
AppType	Analytic and visualisation, device and network management, device usage
PlatformType	Web-based services, Cloud infrastructure and services
SensorType	Humidity, temperature, accoustic, pressure, motion, chemical, luminosity, flowmeter
ActuatorType	Hydraulic, mechanical, electric, pneumatic, magnetic, thermal, TCP/SCP
ControllerType	MicroController, microProcessor, FPGA
DSType	NFS, GPFS, HDFS
DBType	SQL, NoSQL, GraphDB
HWType	Router, Gateway, PowerSupply
EcoType	Interface, DeviceManage, EmbeddedSystems
FirewallType	Software, Hardware
ServiceType	CloudAuthentication, AuthenticationSystem, ID-S/IPS
NetType	PAN, WPAN, WAN, VPN, LAN, WLAN
ProtocolType	BLE, RFID, Wife, ZigBee, ZWave, CoAPP, MQTT, LoRaWAN
PropertyType	Identification, authentication, authorisation, confidentiality, integrity, non-repudiation, availability, privacy, trust, audit, detection
PwdType	Strong, weak, multi-factor
CipherType	AES128GCM, Camellia, ChaCha20
ChannelType	SSL/TLS, HTTPS, tunneling
CertType	$\times$509, openPGP, openSSL, SAML
SigType	SRP, PSK
HashingType	SHA-2, SHA-3
RelationshipType	Secure communication, encrypted information, identified user, authenticated user, authorised user, trust guaranteed, privacy guaranteed, audit guaranteed, non-repudiation guaranteed, detection system

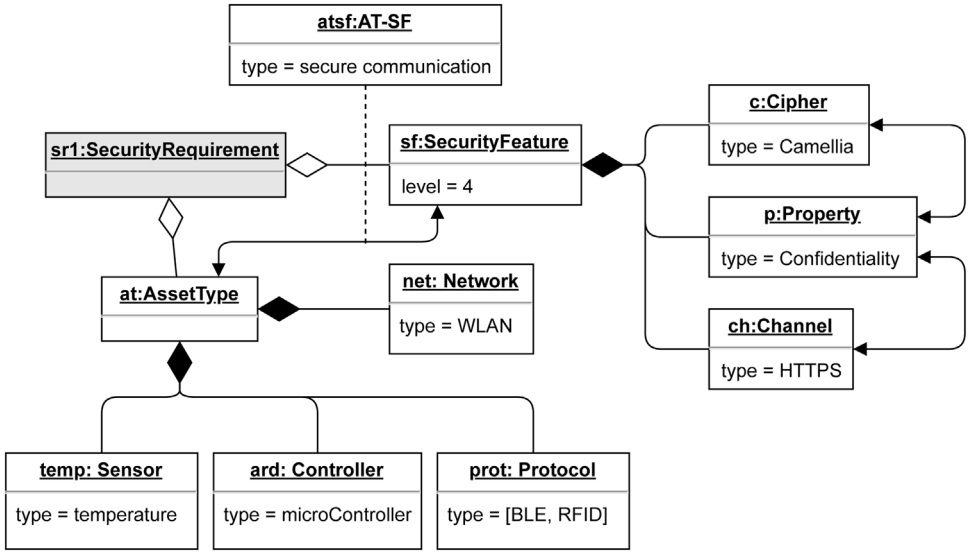

Fig. 4. Instance of security requirement (sr_1).

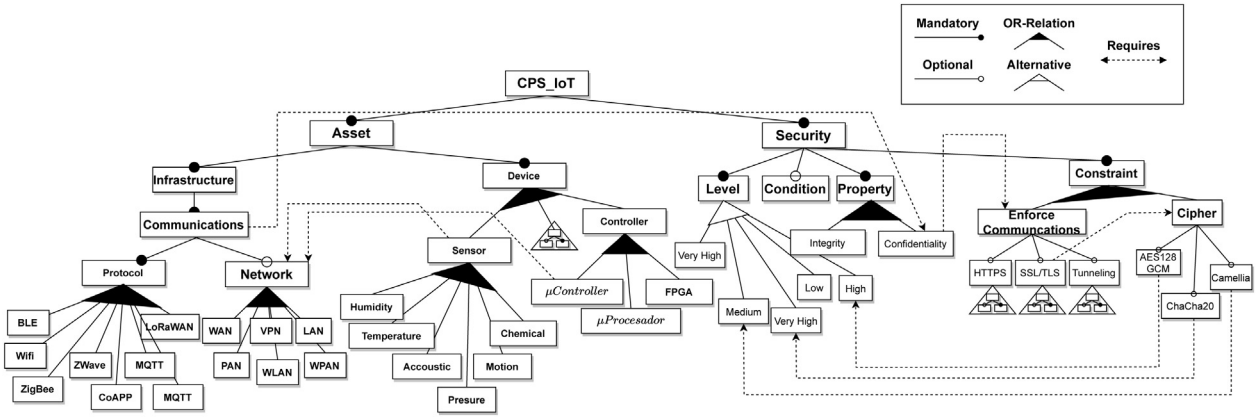

Fig. 5. (Partial) Feature model with security features.

specific context of CPS according to the *SR*s formalised in Section 4.1.

In Fig. 6, a FM was defined (Varela-Vaca et al., 2020b) for the definition of a security CPS and IoT according to the ENISA and OWASP. Once, this is just a FM proposal which encompasses the approaches of ENISA and OWASP but it can be extended or updated to support other standards and approaches. Bear in mind that several parts have been hidden for clarity as some require relations and sub-models. As can be seen, the FM is encompassed of two main parts: (1) the assets (cf., Asset) involved in the security requirement, and; (2) the security requirements (cf., Security) specification where properties, conditions, and constraints can be defined. Some of the included components are obviously related to the mentioned in the sr_i, but not with a one-to-one relationship, the reason why a later mapping is necessary. Derived from these two parts (Asses and Security) the Feature Model Formalisation definition (Varela-Vaca et al., 2019a) is adapted by dividing the types of features into these two subgroups (F_A and F_S) those can be related. The adaptation of the general definition of FM to the security requirement context will facilitate the later automation of the verification and diagnosis analysis.

Definition 4. Feature model for CPS and IoT. Let CPS_IoT_{FMF} be a formalised FM consisting of the tuple $\langle F, R\rangle$, where F is a set of Boolean variables $\{f_{A_1}, f_{A_2}, \ldots, f_{A_n}, f_{S_1}, f_{S_2}, \ldots, f_{S_m}\}$ that represent the features and R as a set of relations r_i that are a tuple (S,BR), where the scope $S=(f_i, f_j)$ is a tuple of variables that participate in a relation and $BR \subseteq D_{f_i} \times D_{f_j}$ are the tuples satisfying the relation and d_{f_k} is the domain of the values of the feature f_k, where f_i and $f_j \in \{f_{A_1}, f_{A_2}, \ldots, f_{A_n}, f_{S_1}, f_{S_2}, \ldots, f_{S_m}\}$.

The available CPS_IoT_{FMF} contains almost 30 features among asset types and security features and more than 100 relations. Derived from the complexity, the automatic feature model analysis (Galindo et al., 2019a) in an efficient way is necessary. The CPS_IoT_{FMF} is accessible for public use through the catalogue provided by our tool CyberSPL.[1] .

Following with CARMEN steps, how to automate the mapping between the security requirement sr_1 (as the shown in Fig. 4) and the FM for CPS_IoT_{FMF} (as shown in Fig. 5) is detailed.

4.3. Step 3: Mapping between security requirements and features models

The metamodel provides a mechanism to define an instantiation of security requirements (sr_i) but it lacks mechanisms to verify the correctness of the instances concerning a high variability of options and restrictions to be configured for CPS. For this reason, we have proposed in the previous step the use of FMs (see Step 2) to describe the variability of configurations and to provide reasoning mechanisms. However, to provide interoperability between the security requirement metamodel instances (see Step 1) and the FMs (see Step 2), we propose to use model weaving techniques (Fabro and Valduriez, 2009). Model weaving is a Model-Driven Engineering technique that enables the creation of abstract links between model elements to customise the model-to-model transformation. The main goal of our weaving approach is to obtain an artefact from an instance of the security requirement metamodel (sr_i) and the FM which enables us to verify the correctness of the instance (sr_i).

An overview of the model weaving in our particular case is presented in Fig. 7. The model weaving enables to customise of the model-to-model transformation. For this reason, the model-to-model transformation (cf., M-M symbol) is fed from the security requirement metamodel, the FM, and the weaving templates. The result of the model weaving is a customised model-to-model transformation that enables the generation of a Security Configuration as an artefact to be verified and diagnosed. Thus, some elements of the model weaving are taken from the security requirements and others from the feature model, being necessary some templates (i.e., weaving templates) generate some parts of the output model (i.e., Security configuration). The weaving templates capture different transformation patterns between the security requirement elements and the FM elements, pointing to both the security requirement and the FM.

To illustrate the mapping an step-by-step example based on the sr_1 requirement is given in Figs. 8–10 . Bear in mind, there are entities in the sr_1 instantiation that can be automatically mapped to the features in the FM. However, other features need to be inferred since the entity or property in the sr_1 are not equivalent to any entity or relation in the FM and vice versa.

Fig. 8 represents a partial model weaving which focuses on one asset from sr_1. In this example, we highlighted in grey colour the concepts Asset Type and Temperature Sensor to be mapped, while the rest of the concepts have been skipped in white colour. As commented, there is a direct mapping between the Sensor and Asset Type objects, and the features Sensor, Temperature, and Asset from FM in yellow colour. Also as we can see, there are other features from FM as Device or CPS_IoT, that are inferred due to the intrinsic relation w.r.t the automatic mapped features. In this case, the use of weaving templates to describe this pattern enables to infer that relationship among the element from both models.

The complete mapping for the rest of asset types from sr_1 to the features of FM is shown in Fig. 9.

The complete mapping is given in Fig. 10 where the security features of sr_1 are also mapped to the feature of the FM.

As aforementioned, the weaving model enables to generate a complete model-to-model transformation and it returns the Security Configuration (sc) as the resulting artefact to be verified and diagnosed against the FM. Therefore, for each sr_i we can obtain an equivalent sc_i. The security configuration (White et al., 2014) is a specific arrangement of features according to the specification of a security requirement.

Definition 5. Security configuration as the equivalent of security requirement. Let sr^i be a security requirement, CPS_IoT_{FMF} the formalised feature model, and sc_i the equivalent security configuration resulting for the weaving using ω_t as a weaving template:

$$(2)(sr_i \overset{\omega_t}{\bowtie} CPS\_IoT_{FMF}) \rightarrow sc_i$$

For instance, the following sc_1 is the resulting artefact from the mapping of sr_1:

sc_1={CPS_IoT, Asset, Device, Sensor, Temperature, Controller, microController, Infrastructure, Communications, Protocol, BLE, RFID, Network, WLAN, Security, Enforce Communications, Property, Confidentiality, Level, High, Constraint, HTTPS, Cipher, Camellia}

;1;

Thus, sc_1 represents a particular selection of features for the CPS_IoT_{FMF} according to the sr_1 requirement. This security configuration is the artefact that can be verified against CPS_IoT_{FMF}.

4.4. Step 4: Verification and diagnosing security configurations

As explained in previous sections, for each sr_i a security configuration sc_i is derived through the mapping application. This sc_i represents a specific selection of features as shown in Fig. 11. As proposed in the CARMEN process, each sc_i must be verified according to the FM that represents the possible correct security configurations, i.e, CPS_IoT_{FMF}.

[1] https://n9.cl/models

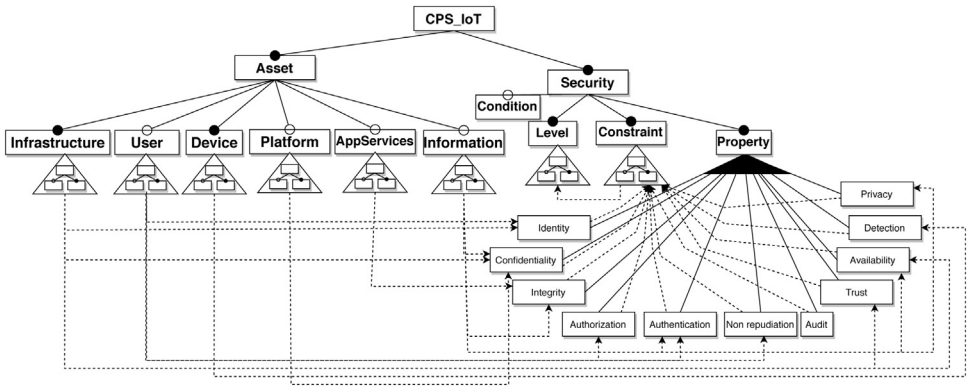

Fig. 6. Feature model for CPS and security requirements.

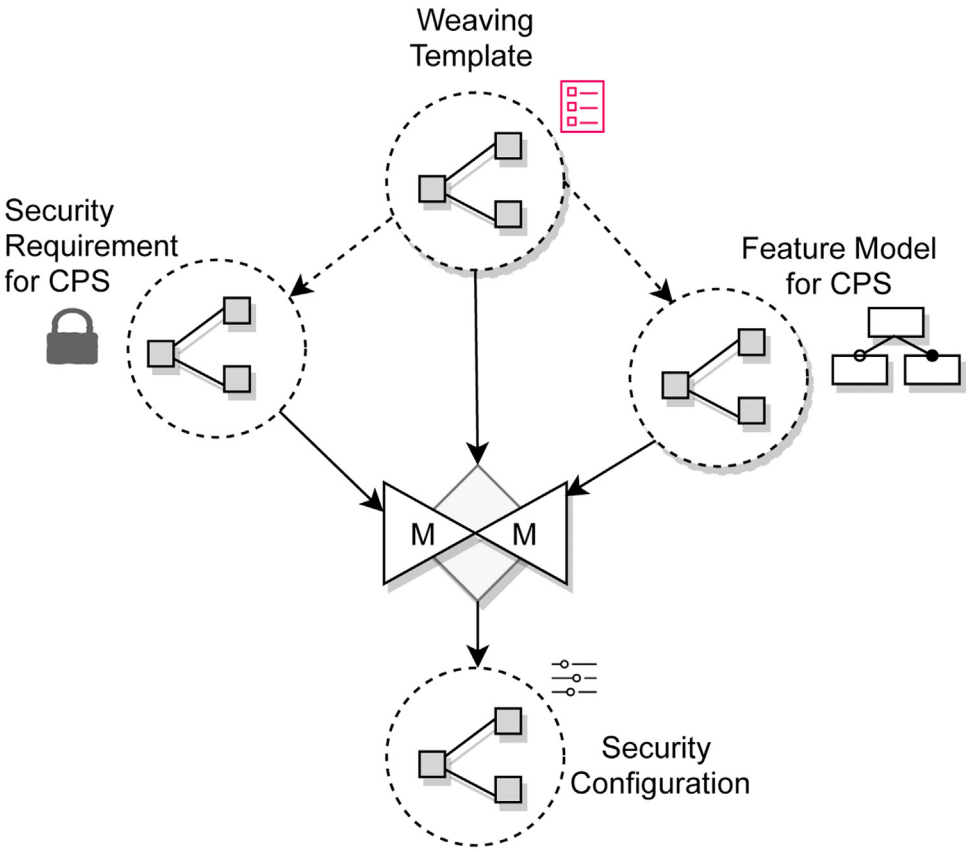

Fig. 7. Weaving models for generation of security configurations.

The verification of a sr_i is based on the definition of a valid configuration in the security context as introduced in Varela-Vaca et al. (2019a). Thus, a configuration sc_i, represents an assignment of features for certain *FM*. For instance, a security configuration $sc_i = \{CPS\_IoT = true, Asset = true, Infrastructure = true, Network = false, Communications = true, \dots\}$ represents an assignment for the reduced model in Fig. 5, where missed features are assigned to *false* value. The same sc_i can be represented without the Boolean values but the same semantic, thus, $sc_i = \{CPS\_IoT, Asset, Infrastructure, Communications, \dots\}$.

The verification can conclude that the sc_i is *valid* (i.e., correct) whether the selection of assigned features satisfies all the FM relations, or *invalid* otherwise. We revisited the definition of valid configuration (Varela-Vaca et al., 2019a) to adapt it for the context of the verification of a security configuration (sc_i) by considering it as a configuration to be checked.

Definition 6. Verification of security configurations. Let (*FM*, sc_i) be the tuple that represents the feature model, $FM : (F, R)$ (see Definition 4), and the security configuration, sc_i, respectively. Let sc_i be a configuration assignment of n asset types and security features $\{f_{A_1}, f_{A_2}, \dots, f_{A_n}, f_{S_1}, f_{S_2}, \dots, f_{S_m}\}$ according to *FM*. Thereby, the sc_i is verified as valid when all the included features in the requirement satisfy the relation of *FM*:

$$(3) verify(FM, sc_i) = \textbf{valid} \Leftrightarrow \{\forall r_j \in R \mid r_j(sc_i) \equiv true\}$$

For instance, any sc_i assignment that includes *Confidentiality* but no *Enforcement Communications* and one of the features *HTTPS, SSL/TLS or Tunnelling* will be *invalid* due to the relations between those features are unsatisfied.

When a sc_i is verified as invalid, there are some failures (no correct assignment to features) in that configuration. The minimal diagnosis represents the explanation which turns into an invalid

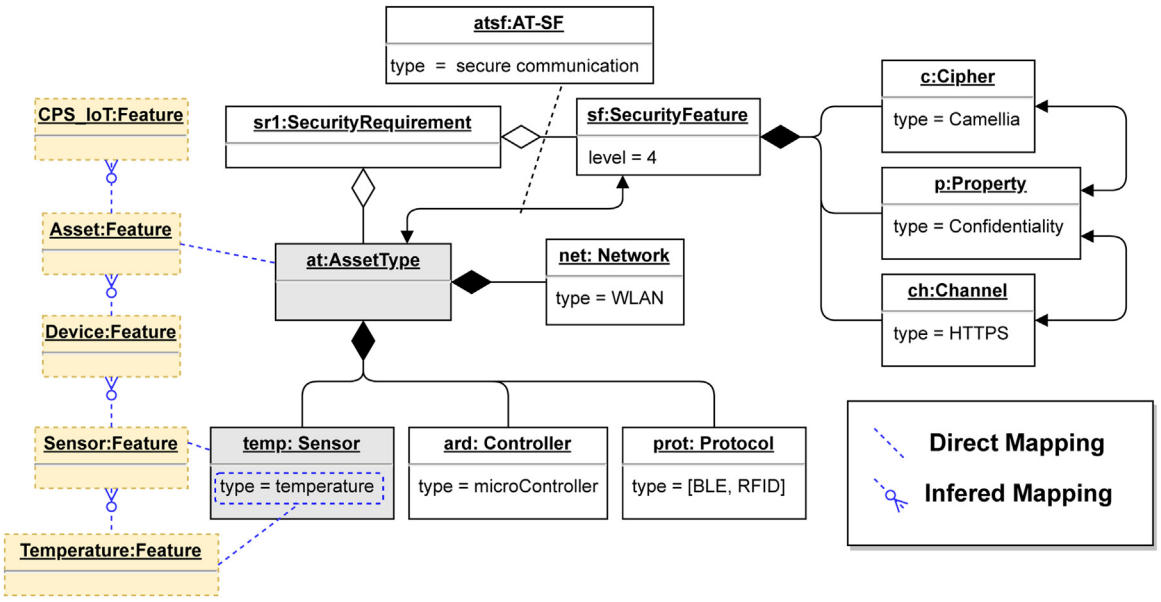

Fig. 8. Example of weaving for mapping sr_1.

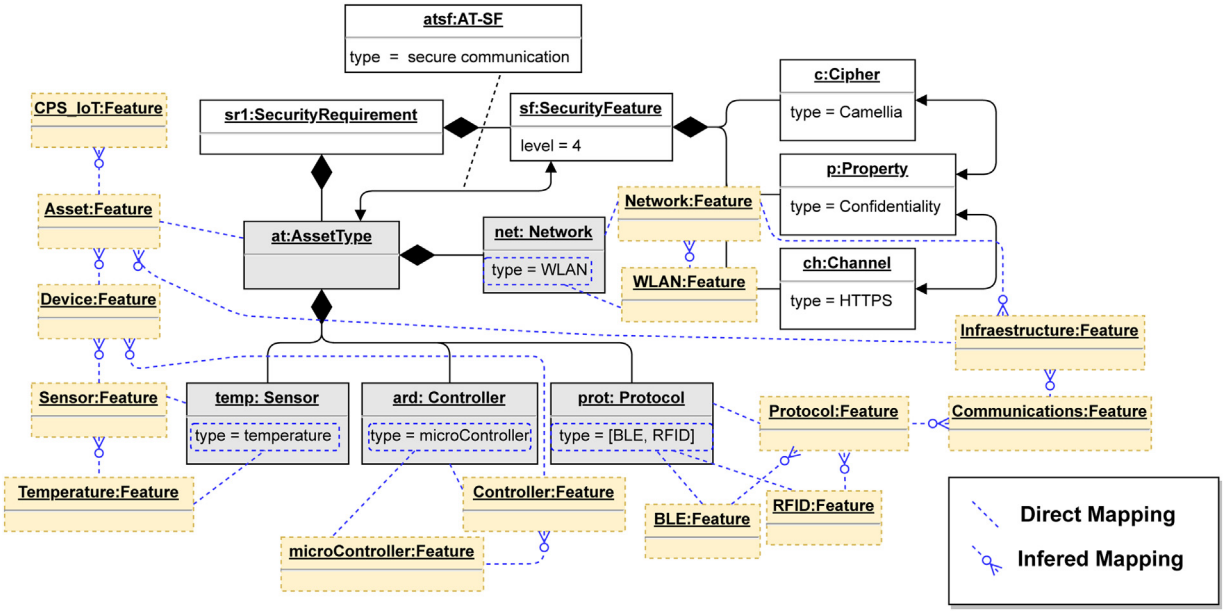

Fig. 9. Example of weaving for mapping sr_1.

configuration into a valid one. The two possible modification operations available are the elimination ($\angle^i_-$) or addition ($\angle^i_+$) of features into sc_i. For simplification reasons, a configuration could be also represented as a set of features, whose belonging to the set implies a *true* value in the assignment. In Fig. 5 a configuration with the set of features $sc_i = \{CPS\_IoT, Asset, Infrastructure, Communications, Protocol, MQTT, WLAN, \dots\}$, not including *Network* is invalid. This security configuration can be transformed into valid just including (1) *Network* or (2) eliminating *WLAN*. Additionally, in option (2), *Sensor* and μController must also be included. We have revisited the diagnosis of feature models configurations in (White et al., 2010) towards the minimal diagnosis of a security configuration.

Definition 7. Diagnosis of Security Misconfigurations. Let sc_i an invalid security configuration, $\triangle$ is the minimal diagnosis as the set of modification of features (i.e., inclusions or eliminations operations) to obtain a valid sc^m_i with regard to *FM*:

$$(4) verify(FM, sc_i) = \textbf{invalid} \xrightarrow{\triangle_{\min}} verify(FM, sc^m_i) = \textbf{valid}$$

The operations to transform sc_i into sc^m_i must be minimal, that implies:

$$sc_i \xrightarrow{\angle^1_-} sc^1_i \xrightarrow{\angle^2_+} \cdots \xrightarrow{\angle^{m-1}_+} sc^{m-1}_i \xrightarrow{\angle^m_+} sc^m_i \quad (5)$$

We assume that just one changed is produced in every modification operation ($\angle^i_*$). Thus, the difference in the number of elements in a sc^k_i w.r.t the next sc^{k+1}_i after a ($\angle^i_*$), will result in 1. Furthermore, we assume that operations are applied at invalids intermediate security configuration until a valid one is reached:

$$\forall_k sc^1_i \cdots sc^{m-1}_i, \quad verify(FM, sc^k_i) = invalid \wedge ||sc^k_i| - |sc^{k+1}_i|| = 1 \quad (6)$$

Following with the previous example, the diagnosis of $\{CPS\_IoT, Asset, Infrastructure, Communications, Protocol, MQTT, WLAN, \dots\}$ could be: the diagnosis for the option (1) results into a

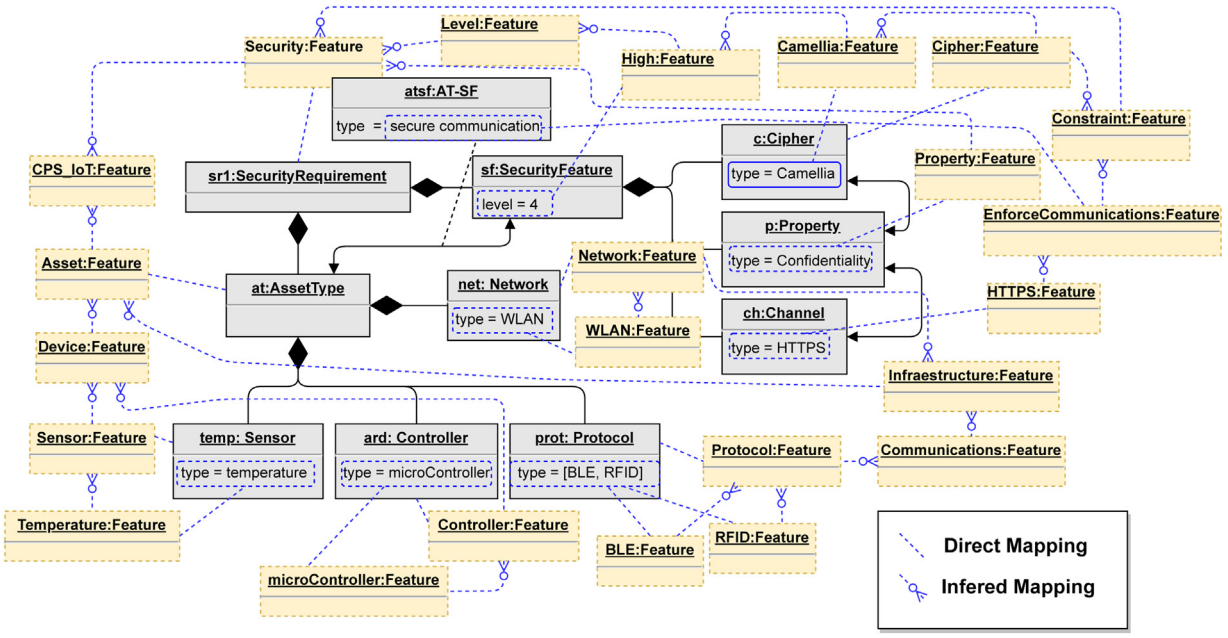

Fig. 10. Example of weaving for mapping sr_1.

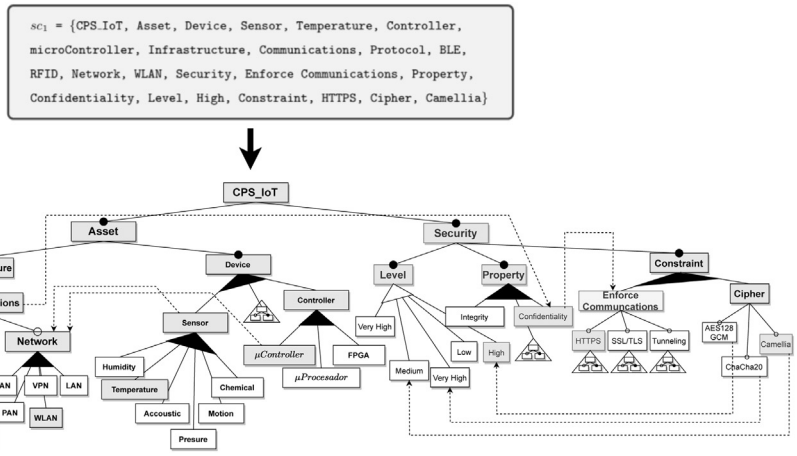

Fig. 11. Selection of features according to security configuration in FM.

cost of 1, thus, to including a feature ($\triangle^1_{sc_1} = \{\angle^1_+, \text{Network}\}$); and the diagnosis of the option (2) results into a cost of 3, thus, to eliminating one feature and including two features ($\triangle^2_{sc_1} = \{\{\angle^1_-, \text{WLAN}\}, \{\angle^2_-, \text{Sensor}\}, \{\angle^3_+, \mu\text{Controller}\}\}$). Regarding both possible diagnosis, the minimal diagnosis is $\triangle_1$ with the minimal cost. The detection of the minimal diagnosis is supported by our tool thanks to the FAMA reasoner (Benavides et al., 2007).

When a CPS has several security misconfigurations the union of all the minimal diagnosis, e.g., $\{\triangle_1, \triangle_2, \ldots, \triangle_n\}$ composes the diagnosis of the CPS.

5. Experimentation

To complement the approach, we define various security requirements to illustrate how to follow the CARMEN framework process for hydroponic farming (see Section 3). The experimentation is composed of two security requirements that are instantiated with our metamodel (see Fig. 3).

Security requirement sr_2**:** The short-range sensors of temperature, light and humidity are connected to an Arduino controller via Bluetooth. The transmitted information acts under the HTTP client/server protocol but the transmitted information must be secured by applying the SSL/TLS cryptographic protocol over HTTP, ensuring confidentiality. This information is stored encrypted in a local webserver with HDFS and HBASE, ensuring integrity (Fig. 13). On the one hand, we have the stored information that must guarantee integrity, so we have a set of assets with database and datastore that are related to the security feature (sf_1) with a very high-security level (cf., level = 5) whose property is integrity (defining a type of encryption). The other part of the security requirement defines that the communication must be confidential (with a high level, level = 4), so the security feature (sf_2) is defined which has the property of confidentiality and a secure communication channel HTTPS. This set of assets (at_1) includes the sensors (temperature, luminosity, humidity), the controller (microcontroller), the protocol (BLE) and network (WLAN), with which they communicate with each other.

We can use the same viewpoint shown in Fig. 12. In this use case, we generate two different security configurations sc_{2a} and sc_{2b} after applying the mapping to the CPS_IoT_{FMF} is as follows:

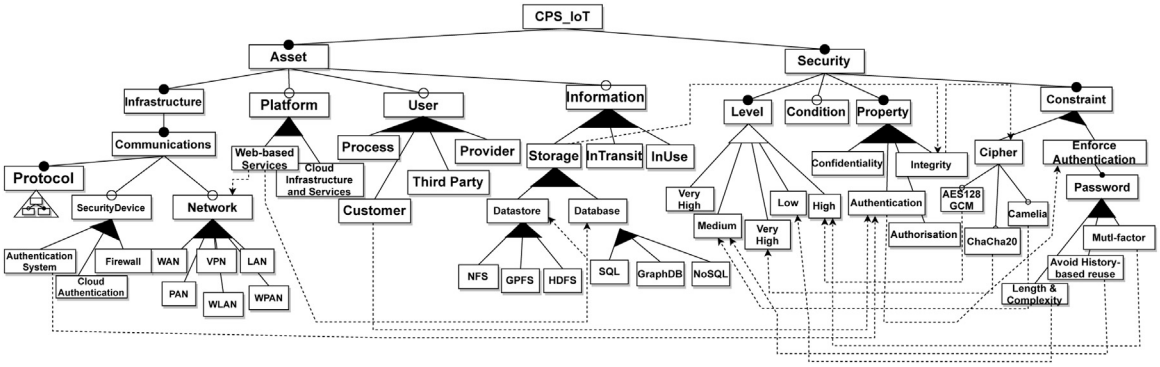

Fig. 12. (Partial) Feature model for CPS and IoT viewpoint.

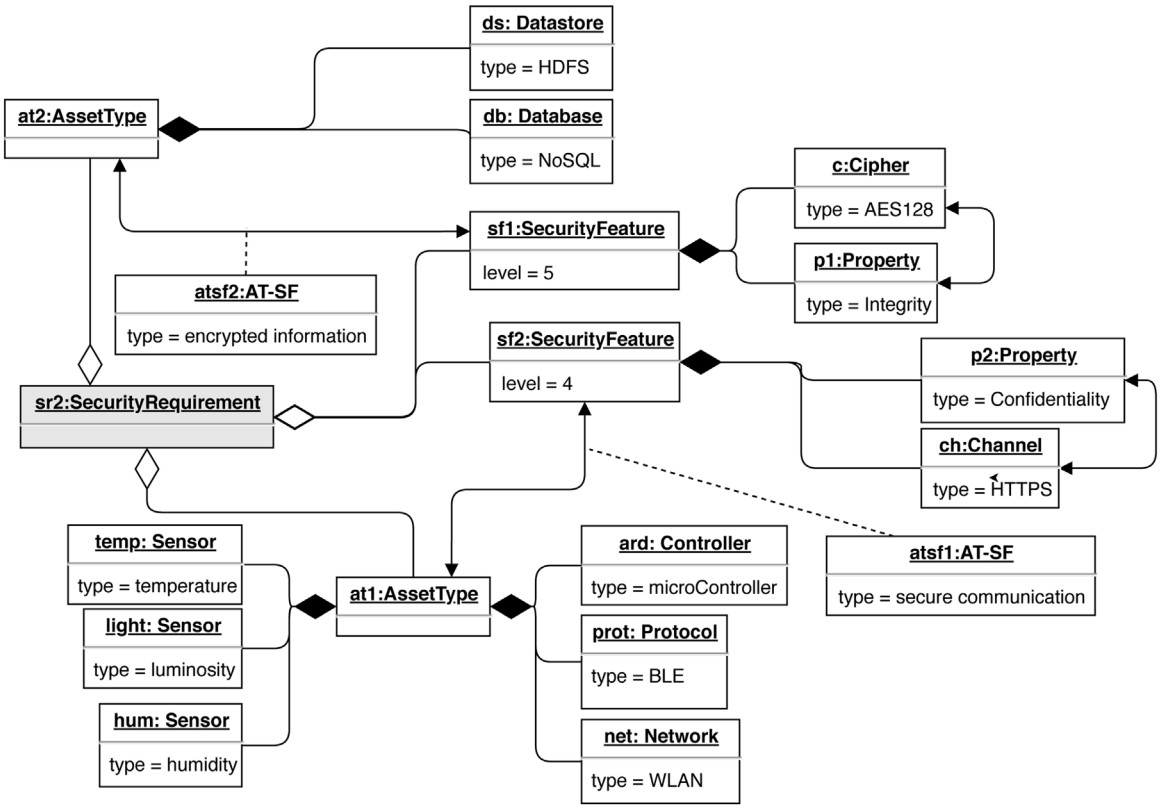

Fig. 13. Security requirement sr_2.

sc_{2a}={CPS_IoT, Asset, Device, Sensor, Humidity, Temperature, Controller, microController, Infrastructure, Communications, Information, Storage, Datastore, HDFS, Database, NoSQL, Protocol, BLE, Wifi, Security, Enforce Communications, Property, Integrity, Confidentiality, Level, VeryHigh, Constraint, SSL_TLS, Cipher, AES128GCM}
;1;
sc_{2b}={CPS_IoT, Asset, Device, Sensor, Humidity, Temperature, Luminosity, Controller, microController, Infrastructure, Communications, Information, Storage, Datastore, HDFS, Database, NoSQL, Protocol, BLE, Wifi, Network, WLAN, Security, Enforce Communications, Property, Integrity, Confidentiality, Level, VeryHigh, Constraint, SSL_TLS, Cipher, AES128GCM}
;1;

In the process of verification, the sc_{2a} configuration is verified as *invalid*, whereas the sc_{2b} configuration is *valid*. After applying the diagnosis, the minimal diagnosis consists of the following possible options:

$$\triangle^1_{sc_{2a}} = \{\{\angle^1_+, High\}, \{\angle^2_-, VeryHigh\}\} \quad (7)$$

$$\triangle^2_{sc_{2a}} = \{\{\angle^1_+, ChaCha20\}, \{\angle^2_+, AES128GCM\}\} \quad (8)$$

In both options, the minimal diagnosis consists of 2 operations. The first diagnosis set ($\triangle^1_{sc_{2a}}$) offers the option to change the Very High (level = 5) to High (level = 4) level since the use of AES128GCM is considered as High. On the other hand, the second diagnosis set ($\triangle^2_{sc_{2a}}$) offers to change the AES128GCM encryption function in favour of ChaCha20 as the function considered as Very High level of security.

Security requirement sr_3**:** The nutrient injector is a critical device for the hydroponic farming and therefore integrity, confidentiality and authentication must be guaranteed in all actions

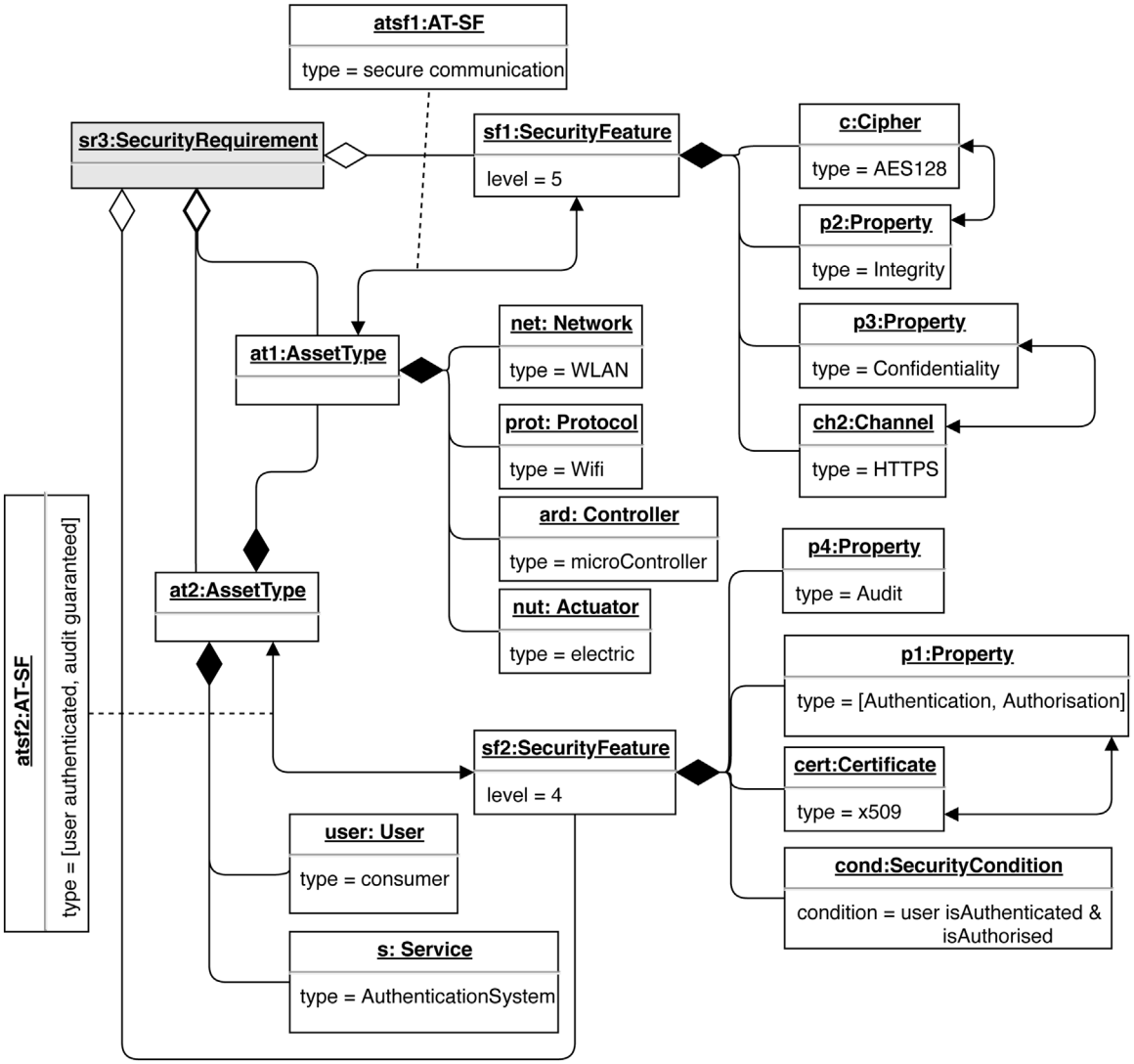

Fig. 14. Security requirement sr_3.

performed with it. Therefore, only a user with a valid ×509 digital certificate will be in charge of sending orders to the device and this access will be registered (Fig. 14). In this requirement we have two clearly differentiated sets of assets (at_1 and at_2), and, in turn, one of them contains the other one. On the one hand, the part of the injector and the Wifi communication that enters and leaves the device (asset type at_1), which must guarantee a secure communication through security feature sf1, which has a very high level of security ensuring confidentiality and integrity (with secure channel HTTPS and encrypted communication using AES128GCM). On the other hand, the set of assets at_2 refers to the user who must access it (to use injector is necessary to include at_1) through an authentication service and who must comply with the security feature sf_2. Sf_2 guarantees, with a high level of security, compliance with authentication and authorisation through a user's ×509 certificate and must comply with correct user status as a condition for access. Besides, it guarantees the registration of all the accesses of any assets at_2 (i.e., the user who wants to access) made by authentication service, thereby complying with the Auditing property.

The security configurations generated from the requirement are divided into two sets, according to the level of security:

sc_{3a}={CPS_IoT, Asset, Device, Actuator, Electric, Controller, microController, Infrastructure, Communications, Protocol, Wifi, Network, WLAN, Security, EnforceCommunications, Property, Integrity, Confidentiality, Level, VeryHigh, Constraint, SSLTLS, Cipher, AES128GCM}
;1;

sc_{3b}={CPS_IoT, Asset, Device, Actuator, Electric, Controller, microController, Infrastructure, Communications, Protocol, Wifi, Network, WLAN, User, Customer, SecurityDevice, AuthenticationSystem, Infrastructure, Security, Property, Authentication, Authorisation, Audit, Level, High, Constraint, EnforceAuthentication, Condition, Certificate, x509}
;1;

Both configurations are verified as *invalid*. The diagnosis for sc_{3a} is the same than for sc_{2a}:

$$\triangle^1_{sc_{3a}} = \{\{\angle^1_+, High\}, \{\angle^2_-, VeryHigh\}\} \tag{9}$$

$$\triangle^2_{sc_{3a}} = \{\{\angle^1_+, ChaCha20\}, \{\angle^2_+, AES128GCM\}\} \tag{10}$$

On the other hand, the diagnosis for sc_{3b} is as following:

$$\triangle^1_{sc_{3b}} = \{\{\angle^1_+, EnforceCommunications\}, \{\angle^2_+, Confidentiality\}, \{\angle^3_+, Password\}, \{\angle^4_+, Multifactor\}\} \tag{11}$$

The minimal diagnosis consists of 4 modifications. Similar to other requirements, the use of communication infrastructure requires the enforcement of confidentiality in any way. Further, the enforcement of the authentication requires the selection of a password policy mechanism according to the High level, thus, a Multifactor password system.

To conclude, the whole diagnosis for the hydroponic farming is the union of all the minimal diagnosis: $\{\triangle^1_{sc_{2a}}, \triangle^2_{sc_{2a}}, \triangle^1_{sc_{3a}}, \triangle^2_{sc_{3a}}, \triangle^1_{sc_{3b}}\}$

6. Conclusions and future work

In recent years, security by design has been widely demanded in many areas but for CPS in particular. However, security requirements associated with a CPS implies a very high number of characteristics and potential configurations that makes the assessment and diagnosis of security requirements very difficult. To fulfil the gap, we present the CARMEN framework to guide users in the definition of correct security requirements for CPS and to enable the diagnosis of these requirements. CARMEN includes the development of a metamodel that supports the definition of the security requirements for CPS environments. CARMEN makes use of feature models based on SPL to represent the possible correct configurations, mapping them with the security requirements to verify if they are correct or diagnose them otherwise. To specify the security requirements, we have defined a common grammar to describe the security requirements for the CPS through the use of an object diagram that gathers the possible elements involved. To perform the diagnosis of the requirements, feature models are used, which will allow us to verify the configurations of each of the requirements. The developed solution allows the expert to describe the requirements, associating them to feature models and establishing a system diagnosis. It has been evaluated by means of a case study of hydroponic cultivation associated with Agriculture 4.0, as this is one of the fields that has grown the most due to the COVID-19 pandemic, which has required more autonomous systems and closer to the cities. Although CARMEN follows the "security by design" philosophy that allows building new systems considering security as a requirement in its analysis, design and construction, its quality will depend on a correct definition and identification of the security requirements and the expertise in the modelling of these requirements. CARMEN is driven by security requirements, so in order to protect against a type of attack, we must first identify it using security engineering techniques, such as abuse cases, attack trees, security use cases, etc. to correctly identify the security requirement, being sure that the resulting security configuration, after applying CARMEN, contains the necessary features to protect the system from that type of attack previously identified by the security requirement. For the future, the approach presented in the paper can be extended in many and diverse directions, but we plan to explore the following ideas: (1) to extend the current models (meta and feature) to cover other properties such as safety one; or to support other guides and standards related to more specific protocols and security controls for CPS; (2) to extend the types of reasoning that can be applied to the combination of the feature models and the security requirements for CPS, expanding the stages of the framework to be used as a compliance tool, increasing the automation of some of its stages, and verifying them, and; (3) to explore the application of CARMEN in other real-time environments of CPS by enabling runtime verification of security requirements.

Conflict of interest

None declared.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgements

This work has been funded by the projects AETHER-UCLM (Ministerio de Ciencia e Innovación, PID2020-112540RB-C42), AETHER-US (PID2020-112540RB-C44/AEI/10.13039/501100011033), COPERNICA (Consejería de Transformación Económica, Industria, Conocimiento y Universidades, Junta de Andalucía, P20_01224), and GENESIS (Consejería de Educación, Cultura y Deportes, Junta de Comunidades de Castilla La Mancha, Fondo Europeo de Desarrollo Regional FEDER, SBPLY/17/180501/000202).

References

Abendroth, N.P., Kleiner, B.A., 2017]. Cybersecurity Policy for the Internet of Things. Guidelines, Microsoft https://www.microsoft.com/en-us/cybersecurity/content-hub/cybersecurity-policy-for-iot/.

Arciniegas, J.L., Due nas, J.C., Ruiz, J.L., Cerón, R., Bermejo, J., Oltra, M.A., 2006]. Architecture reasoning for supporting product line evolution: an example on security. Software Product Lines, 327–372.

Arrieta, A., Sagardui, G., Etxeberria, L., 2015]. Cyber-Physical Systems Product Lines: Variability Analysis and Challenges.

Arrieta, A., Wang, S., Sagardui, G., Etxeberria, L., 2016]. Search-based test case selection of cyber-physical system product lines for simulation-based validation. In: Mei, H. (Ed.), 20th International Systems and Software Product Line Conference, SPLC 2016, Beijing, China, September 16–23. ACM, pp. 297–306, http://dx.doi.org/10.1145/2934466.2946046.

Ashibani, Y., Mahmoud, Q.H., 2017]. Cyber physical systems security: analysis, challenges and solutions. Comput. Secur. 68, 81–97, http://dx.doi.org/10.1016/j.cose.2017.04.005.

Avižienis, A., Laprie, J.-C., Randell, B., 2004]. Dependability and its threats: a taxonomy. In: Jacquart, R. (Ed.), Building the Information Society. Springer US, Boston, MA, pp. 91–120.

2018]. Baseline Security Recommendations for IoT. https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot.

Batory, D., 2005]. Feature models, grammars, and propositional formulas. In: 9th International Conference on Software Product Lines, SPLC'05. Springer-Verlag, Berlin, Heidelberg, pp. 7–20, http://dx.doi.org/10.1007/11554844_3.

Beek, M.H.t., Fantechi, A., Gnesi, S., 2018]. Product line models of large cyber-physical systems: the case of ertms/etcs. In: 22nd International Systems and Software Product Line Conference - Vol. 1, SPLC'18. Association for Computing Machinery, New York, NY, USA, pp. 208–214, http://dx.doi.org/10.1145/3233027.3233046.

Benavides, D., Segura, S., Trinidad, P., Cortés, A.R., 2007]. Fama: tooling a framework for the automated analysis of feature models. VaMoS 2007, 01.

Benavides, D., Segura, S., Ruiz-Cortés, A., 2010]. Automated analysis of feature models 20 years later: a literature review. Inform. Syst. 35 (6), 615–636, http://dx.doi.org/10.1016/j.is.2010.01.001.

Biffl, S., Eckhart, M., Lüder, A., Weippl, E., 2019]. Introduction to security and quality improvement in complex cyber-physical systems engineering. Security and Quality in Cyber-Physical Systems Engineering, 1–29.

Bramberger, R., Martin, H., Gallina, B., Schmittner, C., 2020]. Co-engineering of safety and security life cycles for engineering of automotive systems. ACM SIGAda Ada Lett. 39 (2), 41–48.

Brambilla, M., Cabot, J., Wimmer, M., 2017]. Model-Driven Software Engineering in Practice, 2nd ed. Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers, http://dx.doi.org/10.2200/S00751ED2V01Y201701SWE004.

Carter, B., Adams, S., Bakirtzis, G., Sherburne, T., Beling, P., Horowitz, B., Fleming, C., 2019]. A preliminary design-phase security methodology for cyber-physical systems. Systems 7 (2), 21.

Colombo, A.W., Veltink, G.J., Roa, J., Caliusco, M.L., 2020]. Learning industrial cyber-physical systems and industry 4. 0-compliant solutions. 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Vol. 1, 384–390.

CPS Public Working Group, 2016]. CPS PWG Cyber-Physical Systems (CPS) Framework Release 1.0, Framework. National Institute of Standards and Technology (NIST) https://s3.amazonaws.com/nist-sgcps/cpspwg/files/pwgglobal/CPS_PWG_Framework_for_Cyber_Physical_Systems_Release_1_0Final.pdf.

Czarnecki, K., Helsen, S., Eisenecker, U., 2004. Staged configuration using feature models. In: Nord, R.L. (Ed.), Software Product Lines. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 266–283.

Dechter, R., 2003. Constraint Processing. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

ETSI, 2020]. Cyber Security for Consumer Internet of Things: Baseline Requirements. Standard ETSI EN 303 645. European Telecommunications Standards Institute (ETSI), France https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf.

Fægri, T.E., Hallsteinsen, S., 2006]. A software product line reference architecture for security. Software Product Lines, 275–326.
Fabro, M.D.D., Valduriez, P., 2009]. Towards the efficient development of model transformations using model weaving and matching transformations. Softw. Syst. Model. 8 (3), 305–324, http://dx.doi.org/10.1007/s10270-008-0094-z.
Galindo, J.A., Benavides, D., Trinidad, P., Gutiérrez-Fernández, A.-M., Ruiz-Cortés, A., 2019a]. Automated analysis of feature models: Quo vadis? Computing, http://dx.doi.org/10.1007/s00607-018-0646-1.
Galindo, J.A., Benavides, D., Trinidad, P., Gutiérrez-Fernández, A.-M., Ruiz-Cortés, A., 2019b. Automated analysis of feature models: Quo vadis? Computing 101 (5), 387–433.
Geismann, J., Gerking, C., Bodden, E., 2018]. Towards ensuring security by design in cyber-physical systems engineering processes. In: Proceedings of the 2018 International Conference on Software and System Process, ICSSP'18. Association for Computing Machinery, New York, NY, USA, pp. 123–127, http://dx.doi.org/10.1145/3202710.3203159.
Group, I.W., et al., 2015]. New Security Guidance for Early Adopters of the IoT. Tech. Rep.
Group, I.W., et al., 2016a]. IoT Security Guidelines. Tech. Rep.
Group, I.W., et al., 2016b]. Future-Proofing the Connected World: 13 Steps to Developing Secure IOT Products. Tech. Rep.
Group, I.W., et al., 2016c]. Industrial Internet of Things Volume g4: Security Framework, iic:pub:g4:v1.0:pb:20160919. Tech. Rep.
Group, I.W., et al., 2017]. Identity and Access Management for the Internet of Things-Summary Guidance. Tech. Rep.
Iglesias, A., Iglesias-Urkia, M., López-Davalillo, B., Charramendieta, S., Urbieta, A., 2019]. Trilateral: Software product line based multidomain iot artifact generation for industrial cps. In: 7th International Conference on Model-Driven Engineering and Software Development, Vol. 1. SCITEPRESS-Science and Technology Publications, Lda, pp. 64–73.
Information Technology Laboratory, 2016]. NIST SP 800-183 – Networks of 'Things'. Standard NIST SP 800-183. National Institute of Standards and Technology (NIST), http://dx.doi.org/10.6028/NIST.SP.800-183.
Information Technology Laboratory, 2019]. Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks: Nistir 8228. Report NISTIR 8228. National Institute of Standards and Technology (NIST) https://www.nist.gov/news-events/news/2019/06/considerations-managing-internet-things-iot-cybersecurity-and-privacy-risks.
ISO Central Secretary, 2018]. ISO/IEC 30141:2018 – Internet of Things (IoT) – Reference Architecture. Standard ISO/IEC 30141:2018. International Organization for Standardization, Geneva, CH https://www.iso.org/standard/65695.html.
ISO Central Secretary, 2021a]. ISO/IEC CD 27402.2 – Cybersecurity – IoT Security and Privacy – Device Baseline Requirements. Standard ISO/IEC CD 27402.2. International Organization for Standardization, Geneva, CH https://www.iso.org/standard/80136.html.
ISO Central Secretary, 2021b]. ISO/IEC DIS 27400 – Cybersecurity – IoT Security and Privacy – Guidelines. Standard ISO/IEC DIS 27400. International Organization for Standardization, Geneva, CH https://www.iso.org/standard/44373.html.
Kenner, A., Dassow, S., Lausberger, C., Krüger, J., Leich, T., 2020]. Using variability modeling to support security evaluations: virtualizing the right attack scenarios. In: VaMoS'20: 14th International Working Conference on Variability Modelling of Software-Intensive Systems, Magdeburg Germany, February 5–7, 2020, http://dx.doi.org/10.1145/3377024.3377026, pp. 10:1–10:9.
Kim, B.-J., Lee, S.-W., 2020]. Understanding and recommending security requirements from problem domain ontology: a cognitive three-layered approach. J. Syst. Softw. 169, 110695, http://dx.doi.org/10.1016/j.jss.2020.110695.
Lezzi, M., Lazoi, M., Corallo, A., 2018]. Cybersecurity for industry 4.0 in the current literature: a reference framework. Comput. Ind. 103, 97–110, http://dx.doi.org/10.1016/j.compind.2018.09.004.
Mörth, O., Emmanouilidis, C., Hafner, N., Schadler, M., 2020]. Cyber-physical systems for performance monitoring in production intralogistics. Comput. Ind. Eng. 142, 106333, http://dx.doi.org/10.1016/j.cie.2020.106333.
Mellado, D., Mouratidis, H., Fernández-Medina, E., 2014]. Secure tropos framework for software product lines requirements engineering. Comput. Stand. Interfaces 36 (4), 711–722.
Mokalled, H., Pragliola, C., Debertol, D., Meda, E., Zunino, R., 2019]. A comprehensive framework for the security risk management of cyber-physical systems. Resilience of Cyber-Physical Systems, 49–68.
OneM2M, 2017]. oneM2M – Standards for M2M and the Internet of Things. Standard oneM2M. OneM2M https://www.onem2m.org/.
2021]. OWASP Internet of Things Project. OWASP https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=Main.
Peisert, S., Margulies, J., Nicol, D.M., Khurana, H., Sawall, C., 2014]. Designed-in security for cyber-physical systems. IEEE Secur. Privacy 12 (5), 9–12, http://dx.doi.org/10.1109/MSP.2014.90.
Peldszus, S., Strüber, D., Jürjens, J., 2018]. Model-based security analysis of feature-oriented software product lines. 17th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, 93–106.
Pirbhulal, S., Gkioulos, V., Katsikas, S., 2021]. A systematic literature review on rams analysis for critical infrastructures protection. Int. J. Crit. Infrastruct. Protect. 33, 100427, http://dx.doi.org/10.1016/j.ijcip.2021.100427.
Rehman, S.U., Gruhn, V., 2018]. An effective security requirements engineering framework for cyber-physical systems. Technologies 6 (3), 65.
Rehman, S., Gruhn, V., Shafiq, S., Inayat, I., 2018]. A systematic mapping study on security requirements engineering frameworks for cyber-physical systems. International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, 428–442.
Riel, A., Kreiner, C., Messnarz, R., Much, A., 2018]. An architectural approach to the integration of safety and security requirements in smart products and systems design. CIRP Ann. 67 (1), 173–176, http://dx.doi.org/10.1016/j.cirp.2018.04.022.
Shaaban, A.M., Gruber, T., Schmittner, C., 2019a]. Ontology-based security tool for critical cyber-physical systems. In: 23rd International Systems and Software Product Line Conference – Vol. B, SPLC'19. Association for Computing Machinery, New York, NY, USA, pp. 207–210, http://dx.doi.org/10.1145/3307630.3342397.
Shaaban, A.M., Gruber, T., Schmittner, C., 2019b]. Ontology-based security tool for critical cyber-physical systems. 23rd International Systems and Software Product Line Conference-Vol. B, 207–210.
Sion, L., Van Landuyt, D., Yskout, K., Joosen, W., 2016]. Towards systematically addressing security variability in software product lines. 20th International Systems and Software Product Line Conference, 342–343.
Souag, A., Salinesi, C., Mazo, R., Comyn-Wattiau, I., 2015]. A security ontology for security requirements elicitation. International Symposium on Engineering Secure Software and Systems, 157–177.
Span, M., Mailloux, L.O., Mills, R.F., Young, W., 2018]. Conceptual systems security requirements analysis: aerial refueling case study. IEEE Access 6, 46668–46682.
ur Rehman, S., Allgaier, C., Gruhn, V., 2018]. Security requirements engineering: a framework for cyber-physical systems. 2018 International Conference on Frontiers of Information Technology (FIT), 315–320.
Varela-Vaca, Á.J., Gasca, R.M., Ceballos, R., Gómez-López, M.T., Bernáldez Torres, P., 2019a]. CyberSPL: a framework for the verification of cyber security policy compliance of system configurations using software product lines. Appl. Sci. 9 (24), http://dx.doi.org/10.3390/app9245364.
Varela-Vaca, Á.J., Galindo, J.A., Ramos-Gutiérrez, B., Gómez-López, M.T., Benavides, D., 2019b]. Process mining to unleash variability management: discovering configuration workflows using logs. 23rd International Systems and Software Product Line Conference-Vol. A, 265–276.
Varela-Vaca, Á.J., Gasca, R.M., Carmona-Fombella, J.A., Gómez-López, M.T., 2020a]. AMADEUS: towards the automated security testing. In: Lopez-Herrejon, R.E. (Ed.), SPLC'20: 24th ACM International Systems and Software Product Line Conference, Montreal, Quebec, Canada, October 19-23, Vol. A. ACM, http://dx.doi.org/10.1145/3382025.3414952, pp. 11:1–11:12.
Varela-Vaca, Á.J., Rosado, D.G., Sánchez, L.E., Gómez-López, M.T., Gasca, R.M., Fernández-Medina, E., 2020b. Definition and verification of security configurations of cyber-physical systems. In: Katsikas, S., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Kalloniatis, C., Mylopoulos, J., Antón, A., Gritzalis, S., Meng, W., Furnell, S. (Eds.), Computer Security. Springer International Publishing, Cham, pp. 135–155.
White, J., Benavides, D., Schmidt, D., Trinidad, P., Dougherty, B., Ruiz-Cortes, A., 2010]. Automated diagnosis of feature model configurations. J. Syst. Softw. 83 (7), 1094–1107, http://dx.doi.org/10.1016/j.jss.2010.02.017, sPLC2008.
White, J., Galindo, J.A., Saxena, T., Dougherty, B., Benavides, D., Schmidt, D.C., 2014]. Evolving feature model configurations in software product lines. J. Syst. Softw. 87 (1), 119–136, http://dx.doi.org/10.1016/j.jss.2013.10.010.
Yaacoub, J.-P.A., et al., 2020]. Cyber-physical systems security: limitations, issues and future trends. Microprocessors Microsyst. 77, 103201, http://dx.doi.org/10.1016/j.micpro.2020.103201.
Zhu, Q., Sangiovanni-Vincentelli, A., 2018]. Codesign methodologies and tools for cyber-physical systems. Proc. IEEE 106 (9), 1484–1500.
Zunino, C., Valenzano, A., Obermaisser, R., Petersen, S., 2020]. Factory communications at the dawn of the fourth industrial revolution. Comput. Stand. Interfaces 71, 103433, http://dx.doi.org/10.1016/j.csi.2020.103433.