Computers & Security 120 (2022) 102805

Contents lists available at ScienceDirect

Computers & Security

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

The nature of security: A conceptual framework for integral-comprehensive modeling of IT security and cybersecurity

Ricardo Villalón-Fonseca

CITIC - ECCI, Universidad de Costa Rica, San José, Costa Rica

ARTICLE INFO

Article history:
Received 28 October 2020
Revised 27 March 2022
Accepted 9 June 2022
Available online 12 June 2022

Keywords:
Information security
Security framework
Security architecture
Cybersecurity architecture
Security model
Security relationship
Security chain
Risk management

ABSTRACT

Cybersecurity is a broadly defined concept comprising security for many different types of elements. Dealing with cybersecurity is a multidimensional problem, and the damage generated by cyberattacks can be very diverse. Reports about cybersecurity show recurrent problems, or increasing on their frequency of appearance, with no clear approach for solving them. Existing models deal with cybersecurity in several different but general ways, and results are not better. Consequently, managing cybersecurity deserves consideration of a new approach. Our approach is based on the nature of security. Security services are modeled around three basic security concepts, namely isolation, interaction, and representation. With these three concepts, a cybersecurity development starts with security objectives for overcoming the cybersecurity challenges, and also has a security representation to achieve integral and comprehensive security results. We propose an architecture-based security conceptual framework having three components, namely a system representation model kind, a security representation model kind, and a security process model kind, to accomplish the security process for a system. The security process is fully guided and supported with security objectives from the beginning to the end. The framework proposes several models, based on data structures for representing the system, the security, and the process itself. The models are scalable to represent systems of any size, from tiny to huge technology infrastructures, and with support for automation of the security process. The scope of the framework is the security of IT systems and cybersecurity, including information, software, virtual resources, hardware, IT devices, money, people, and other related physical objects being represented digitally. The framework was developed while creating a university cloud infrastructure, and consolidated while supporting the security of several national wide software and infrastructure applications for digital signature in Costa Rica. We aim to provide a new and innovative way for doing cybersecurity, by directly targeting the actual security requirements; with a simple, systemic, structured and potentially automated security process, and for achieving integral and comprehensive security solutions.

1. Introduction

Cybersecurity is a broadly defined concept comprising the security of many different types of elements. It has to take care of the security of information, software, computers, virtual resources, electronic devices, physical infrastructures, IT systems, people, and more (Craigen et al., 2014; Maurer and Morgus, 2014; Schatz et al., 2017). Cybersecurity also includes a list of security concepts such as confidentiality, integrity, availability, privacy, vulnerabilities, threats, risk assessment, security requirements, or security controls, which are being managed with currently existing security models for establishing cybersecurity solutions, but the results are not getting better over time (Accenture Security, 2019; ITU Publications, 2018).

Dealing with cybersecurity is, in general, a multidimensional problem. Cyberattacks and the damage they generate can be very diverse (Agrafiotis et al., 2018), with issues such as personal or organizational information stolen or damaged (Plachkinova and Maurer, 2018), information systems and software applications corrupted, stolen or miss-used (FBI, 2018; Stout, 2000), physical assets stolen or destroyed (The Council of Economic Advisers, 2018), reputation of organizations being degraded (Obermaier et al., 2019), or people privacy violated (Dark Readin, 2016).

Additionally, cybersecurity reports show recurrent security problems or increasing on their frequency of appearance, such as unidentified or unattended risks, failure to consider security for the entire system as a whole, or vulnerabilities related to indirect attacks, like supply chain vulnerabilities (Accenture Security, 2019;

E-mail address: ricardo.villalon@ucr.ac.cr

https://doi.org/10.1016/j.cose.2022.102805
0167-4048/

AON, 2019; Urciuoli et al., 2013), with no clear approach for solving them.

Thus, cybersecurity faces a large number of challenges, starting with uncertainty of the definition of the cybersecurity concept itself (Schatz et al., 2017), followed by ambiguity of basic concepts like risk definition (Goman, 2018), problems for specifying, building and managing security for large systems (ECA, 2019; Fischer, 2016), or handling security for specialized cases such as Internet of Things (IoT) (Pan and Yang, 2018). Furthermore, existing models deal with cybersecurity in several different but general ways, and results are not better, but worse (Bradley, 2019; Muprhy, 2018; Townsend, 2019). Consequently, managing cybersecurity deserves consideration of a new approach.

Our approach for developing cybersecurity is based on the nature of security. Security services are modeled around three basic security concepts, namely isolation, interaction and representation. A cybersecurity development has a security representation, an innovative element that establishes security relationships between the components of a system. The security process starts with security objectives and the three new security concepts, to achieve integral and comprehensive security results, for overcoming the cybersecurity challenges.

We propose a security framework supported by architecture concepts, as defined on ISO/IEC/IEEE 420x0 standard series (ISO, 2011; 2019), with three generic and complementary viewpoints, namely system, security and process. The framework establishes a system representation model kind (i.e. a type or category of model, according to ISO/IEC/IEEE 42010), a security representation model kind, and a security process model kind to accomplish the security process of a system. The system representation model kind describes the components of the system, and also the scope of the security process. The security representation model kind describes the security objectives and security relationships between system components, in a way that they become a step by step guide during the entire security process. The security process model kind describes a methodology, with the activities to obtain security requirements and security controls, from established security objectives, security relationships, and the system representation.

The models are based on data structures. They are fully scalable to represent systems of any size, such as very large and complex technology infrastructures, or even system of systems (SoS). The scope of the framework and the models is the security for IT systems and cybersecurity, comprising all type of objects and entities interacting with and within those systems, including information, software, virtual resources, hardware, IT devices, money, people, and other related objects.

The paper is organized as follow. Related Work summarizes relevant existing security models and international standards related to cybersecurity modeling. Security Use Case section presents a basic example with the main ideas of the framework for modeling the security. Next sections explains the details of the framework, including individual sections for the System Representation, the Security Representation, and the Security Process. Real-Life Framework Applications section summarizes real-life example applications for supporting and being supported by the framework. Featured Framework Applications section provides technical aspects and relevant usage scenarios for additional applications that can be supported by the framework. Finally, last section contains the conclusions and future work.

2. Related work

This section presents a summary of relevant models and current standards, related to information security, IT security and cybersecurity. At the end of the section we provide comments about those models, related to the contribution and scope of our framework.

2.1. Security models

During last three decades, there were relevant developments and multiple publications about security models for information and IT resources.

In 1990, McCumber (1991) developed a three-dimensional (i.e. a cube) security model, known as Infosec, for automated information systems. He proposed dimensions for information states (transmission, storage, process), critical information characteristics (confidentiality, integrity, availability), and security measures (technology, policy & practices, education training awareness). The InfoSec model can be used by selecting an entry of the cube for finding vulnerabilities, given an information state and a critical information characteristic. After identifying the vulnerabilities, security measures can be established according to the third dimension of the cube. The model allows to divide the problem of information security into more specific sub-problems, by identifying the characteristics of the information to be secured.

Maconachy et al. (2001) developed an extended version of McCumber model. His approach considered protection of information but also protection of systems, and he named that as information assurance (IA) model. He also renamed critical information characteristics to security services, and added authentication and non-repudiation to the list of services. Furthermore, he added a time dimension to the model, to enable protection, detection and reaction capabilities to the security solutions, that way enabling management of security events over time. Additionally, security measures became technology, operations (to go beyond policies and practices), and people. Usage of the model is similar to McCumber's Infosec. The added time dimension enables the analysis of security events over time.

Reference Model of Information Assurance & Security (RMIAS) (Cherdantseva and Hilton, 2013) is another security model with four dimensions, namely Information System Security Life Cycle, Information Taxonomy, Security Goals, and Security Countermeasures. The purpose of the model is to overcome the limitations of the previously proposed InfoSec and IA models, but also to address additional elements, such as diversity of elements in the system and de-perimeterisation of the environment covered by the protection. RMIAS scope is the information assurance and security, meaning that there are additional objects to be included for security, such as processes, hardware, software, networks or people. The model addresses the security from early stages, by establishing security goals and including some categorization for their definition. The model also includes a time dimension. Information Taxonomy dimension extends information states to a more detailed characterization of information, by adding format, location, sensitivity, and also considerations about ownership of information. The methodology comprises the process of categorizing information, risk analysis where security goals are prioritized, selection of security countermeasures, and tracking of countermeasures for completion of the security life cycle.

Infosec-Tree (Villalón-Fonseca et al., 2014a; 2014b) is an information security model developed as a predecessor of the framework presented on this research. Their approach is to deal with security complexity for large infrastructures, such as modern cloud platforms, or large data centers. They proposed a comprehensive way to represent security controls, beyond check lists and best practice guidelines, by using a hierarchical tree structure to represent the components of a system. Furthermore, graphs are used to show interactions between components. There is also a data structure called triad, to represent security controls for information being protected inside the components, or between communicating components. The model considers a time dimension for handling the security events before, during or after their occurrence. The methodology for applying the model has four steps,

namely gathering security policies and system components, constructing the Infosec-tree structure, determining triads and countermeasures, and establishing the security controls.

2.2. Security standards

There are important national and international standards addressing the construction or evaluation of security. They contain security modeling aspects to be used as baselines or guidelines, during the construction or evaluation of a security process. A summary of relevant standards is presented in this section.

ISO/IEC 27001 (ISO, 2012) is an international standard of requirements for establishing, implementing, maintaining, and continuously improving an information security management system (ISMS), within the context of an organization. The standard includes requirements for the assessment and treatment of information security risks, tailored to the needs of the organization. The topics addressed by the standard are the organizational context, leadership, planning and objectives, required resources and communication, operational aspects, performance evaluation and continuous improvement. The standard focuses on a high level goal of defining a structure, to achieve a continuous improvement on information security for the organization. Two relevant underlying ideas of the standard are the risk management process and the continuous life-cycle, to get an improved information security management.

NIST Cybersecurity Framework (National Institute of Standards and Technology, 2020) is a project from the United States government. It helps organizations to improve critical infrastructure cybersecurity when they are the owners, operators or suppliers of the infrastructure. The framework consists of three parts, namely the Framework Core, the Implementation Tiers, and the Framework Profiles. The Framework Core describes a set of cybersecurity activities, outcomes, and common informative references across sectors and critical infrastructure. The Implementation Tiers provide a way to view and understand how to manage cybersecurity risks, and also to prioritize and achieve cybersecurity objectives. The Framework Profiles are developed from elements defined in the Framework Core because they represent the outcomes based on business needs. The framework also provides the steps for an organization to create a new cybersecurity program, or to improve an existing one.

Common Criteria for Information Technology Security Evaluation (known as Common Criteria or CC), or its corresponding development/maintenance standard ISO/IEC 15408, is an international standard for certification of computer security. According to the CC web site (Common Criteria, 2020), CC objectives are to ensure high-level and consistent evaluation of IT products and protection profiles (PPs), aiming to significantly improve the confidence in the security of those products and the profiles. It is also to improve the availability of evaluated security-enhanced IT products and PPs, for eliminating the burden of duplicating the evaluations, and for continuously improving the efficiency and cost-effectiveness of the evaluation and certification/validation process. CC allows for specifying security functional and assurance requirements (SFR, SAR respectively) in a Security Target (ST), through PPs. Vendors can implement or make claims about the security attributes of their products. Testing laboratories can evaluate the products to determine if they actually meet the claims.

2.3. Related work considerations

Section 2 described important security models and relevant standards, related to IT security or cybersecurity. Most of them comprise key aspects of a security process, like security services, security objectives as a guide/reference, risk analysis, the time dimension (or life cycle) for handling security events, several different structures for organizing, realizing or evaluating the security process. Current models are also enabled for diverse and large systems, because of their facilities for dividing and organizing the process into components or functions.

Our framework moves one step forward, by supporting and not only enabling scalability and automation of the security process, because of the adopted framework models. We provide highly scalable and well defined data structures for supporting the design of the system representation.

The framework provides a model for security representation, a new element not available in any existing (to our best knowledge) model or framework. The approach for representing security is its nature, meaning a way to precisely identify the location of security requirements, either individually, integrally between components, or between systems.

Furthermore, the security process considers, propagates and applies the security objectives, during the whole process, to avoid any deviation or loss from the objectives at any step.

Next section presents a basic security example, showing the main structures and steps composing the framework.

3. Security use case

This section contains a security example, for describing the security concepts, the structures and the security process proposed in the framework. A more detailed characterization of the whole framework is presented at Section 4.

The system of the example is a box containing two glass cups to be transported to its final destination. Fig. 1a) shows the layout of the system as initially implemented, including some labels with information, as security countermeasures, to protect the cups. The first label indicates the number of cups inside the box, the second label is a warning about the fragile content.

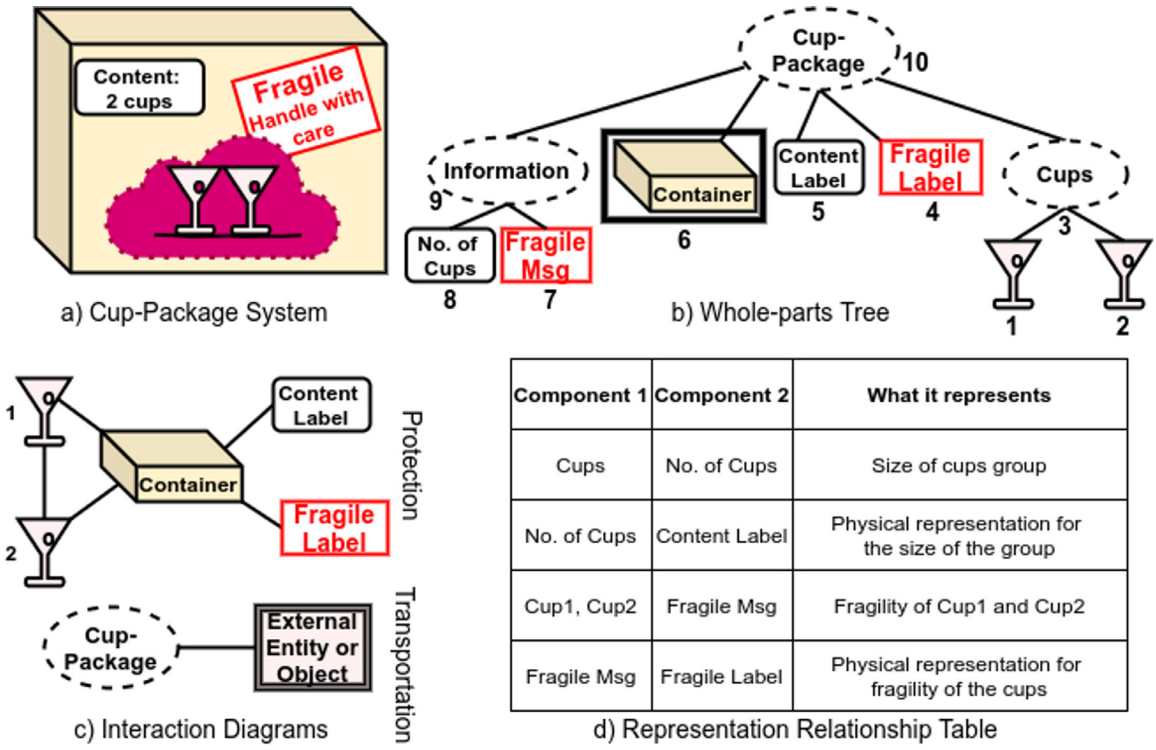
There is a transportation to carry the package (not shown in the figure), which could be a car or any other type of vehicle, and a person in charge of delivering the box, but neither is part of the system to be secured. They are elements with which the system (i.e. the Cup-Package) must interact to accomplish its purpose.

To start the security process, we have to define a representation of the system to be secured, altogether with an initial set of security objectives.

3.1. System representation and security objectives

We need a representation of the system because it establishes not only the scope or boundaries of the system of interest but also the limits of the security process. The system name is `Cup-Package`. It includes the box with the cups, and the labels. System representation has two types of diagrams, namely a whole-parts hierarchical tree, showing the elements composing the system, as shown in Fig. 1b), and also one or more interactions diagrams describing (all) the interactions between components of the system and with external components, as shown in Fig. 1c). These are the models for system representation.

The whole-parts tree is a structure directly obtained from characterizing the components of the system. It represents a real organization of the components by using a whole-and-parts pattern. In the example, the components `Container`, `Content Label`, `Fragile Label`, the `Cups` and `Information` are the components (i.e. parts) of the system `Cup-Package`. `Cup-Package`, `Cups` and `Information` are groups of components, and they are represented with dashed ovals. Groups of components are the only allowed abstraction in a whole-parts tree, as long as each group corresponds to an actual grouping of components in the system.



Component 1	Component 2	What it represents
Cups	No. of Cups	Size of cups group
No. of Cups	Content Label	Physical representation for the size of the group
Cup1, Cup2	Fragile Msg	Fragility of Cup1 and Cup2
Fragile Msg	Fragile Label	Physical representation for fragility of the cups

d) Representation Relationship Table

Fig. 1. Cup-package security use case.

Groups serve the purpose of the security process, as it will be explained later. `Cup-Package` represents the whole system, `Cups` represents the group of cups we want to protect, `Information` represents all the information (i.e. the digital dimension) handled by the system. `Cups` group is shown in the tree because we are interested on protecting the group as a whole but also each individual cup.

Whole-parts tree provides some benefits for system representation. It can be a place for defining component properties, such as the isolation capabilities of the `Container` shown in Fig. 1b). The components belonging to the system may have a name, a description, and additional properties for supporting the security process. The granularity of the described components is reflected by the depth of the tree, and the depth of the tree impacts the level of detail during the security process. In our example, the `Cups` group is shown in the tree, but also each individual cup because we want to establish security at both levels.

As the tree defines the scope of the system, only components belonging to the system are included in the security process. For example, the transportation (i.e. the car, bicycle, etc.) is not included in the tree, then no security controls will be defined for that component, but transportation is shown on the interaction diagrams because of its interaction with the `Cup-Package`, then some recommendations or even external requirements may be suggested about the transport security.

On another hand, interaction diagrams describe interactions between internal components of the system, but also interactions with external entities. Interactions are an important place to look for security requirements in order to achieve the security objectives described later in this section. In the `Cup-Package` example, there are two interaction diagrams, namely Protection and Transportation, as shown in Fig. 1c). Protection diagram shows interactions between individual cups and with the `Container`, but also interactions between the `Container` and each of the labels (made of paper) because they are attached to the box. Transportation diagram shows the interaction of the whole system (`Cup-Package`) with the external transportation, that is not part of the target system.

Once we have a system representation, we need one or more (direct) security objectives to reflect our concerns about the security of the system components. A direct objective is basically a way for specifying what we want about security in the system. Then, we define the following security objective:

> *We want to prevent the cups from being broken or lost while transported to their destination*

Next section describes how to elaborate on this objective, and also derive additional indirect objectives, to obtain a complete security representation for the system.

3.2. Security representation

The security objective stated in Section 3.1 is the first step of a security representation. Next step is to propagate this objective to associate or find all related components of the system (for the security purpose), to integrally support the achievement of the initial objective.

We requested integrity (*broken*) and availability (*lost*) for the group of `Cups`. Then, we need to propagate this objective to the group itself, and for each individual cup (1 and 2). We also asked for integrity and availability of the whole package `Cup-Package`. Then, we define the following (propagated) complete set of security objectives:

DO[1][1] *We want to prevent the Cup 1 from being broken while transported to its destination.*

DO[1][2] *We want to prevent the Cup 1 from being lost while transported to its destination.*

DO[2][1] *We want to prevent the Cup 2 from being broken while transported to its destination.*

DO[2][2] *We want to prevent the Cup 2 from being lost while transported to its destination.*

Table 1
Cup-package - security relationships.

#	Objective 1	Component 1	Service 1	Objective 2	Component 2	Service 2	Relationship
1.	DO[1][1], DO[1][2]	Cup 1	Integrity, Availability	IO[6][1], IO[6][2]	Container	Integrity, Availability	Isolation
2.	IO[6][1], IO[6][2]	Container	Integrity, Availability	————	Cup 1	————	
3.	DO[2][1], DO[2][2]	Cup 2	Integrity, Availability	IO[6][1], IO[6][2]	Container	Integrity, Availability	
4.	IO[6][1], IO[6][2]	Container	Integrity, Availability	————	Cup 2	————	
5.	DO[1][1]	Cup 1	Integrity	DO[2][1]	Cup 2	Integrity	Interaction
6.	DO[2][1]	Cup 2	Integrity	DO[1][1]	Cup 1	Integrity	
7.	DO[1][2]	Cup 1	Availability	————	Cup 2	————	
8.	DO[2][2]	Cup 2	Availability	————	Cup 1	————	
9.	DO[1][1], DO[1][2]	Cup 1	Integrity, Availability	IO[6][1], IO[6][2]	Container	Integrity, Availability	
10.	IO[6][1], IO[6][2]	Container	Integrity, Availability	————	Cup1	————	
11.	DO[2][1], DO[2][2]	Cup 2	Integrity, Availability	IO[6][1], IO[6][2]	Container	Integrity, Availability	
12.	IO[6][1], IO[6][2]	Container	Integrity, Availability	————	Cup2	————	
13.	IO[5][1], IO[5][2]	Content Label	Integrity, Availability	IO[6][1], IO[6][2]	Container	Integrity, Availability	
14.	IO[6][1], IO[6][2]	Container	Integrity, Availability	————	Content Label	————	
15.	IO[4][1], IO[4][2]	Fragile Label	Integrity, Availability	IO[6][1], IO[6][2]	Container	Integrity, Availability	
16.	IO[6][1], IO[6][2]	Container	Integrity, Availability	————	Fragile Label	————	
17.	DO[10][1], DO[10][2]	Cup-Package	Integrity, Availability	————	External Entity	————	
18.	DO[3][1], DO[3][2]	Cups	Availability	IO[5][1], IO[5][2]	Content Label	Integrity, Availability	Representation
19.	IO[5][1], IO[5][2]	Content Label	Integrity, Availability	DO[3][1], DO[3][2]	Cups	Integrity	
20.	DO[1][1], DO[1][2]	Cup 1	Integrity	IO[4][1], IO[4][2]	Fragile Label	Integrity, Availability	
21.	IO[4][1], IO[4][2]	Fragile Label	Integrity, Availability	————	Cup1	————	
22.	DO[2][1], DO[2][2]	Cup 2	Integrity	IO[4][1], IO[4][2]	Fragile Label	Integrity, Availability	
23.	IO[4][1], IO[4][2]	Fragile Label	Integrity, Availability	————	Cup2	————	

DO[3][1] *We want to prevent the Cups from being broken while transported to their destination.*

DO[3][2] *We want to prevent the Cups from being lost while transported to their destination.*

DO[10][1] *We want to prevent the Cup-Package from being broken while transported to its destination.*

DO[10][2] *We want to prevent the Cup-Package from being lost while transported to its destination.*

Notation DO[x][y] allows for identifying specific objectives. Index x represents the component id, as numbered in Fig. 1b). Index y means 1 for integrity, and 2 for availability. Additionally, every objective has three properties, namely:

- A component name, like *Cup 1*.
- An implicitly (or explicit) security service to be achieved, like integrity (for *broken*).
- An event time or timing related to the objective, like *prevent*, for handling the security event before it happens.

At this point, we have a full and detailed set of direct security objectives, reflecting the security concerns for `Cup 1`, `Cup 2`, `Cups`, and `Cup-Package`. Now, we are going to derive (indirect) security objectives for additional system components, to integrally achieve the direct objectives in a comprehensive way. Indirect security objectives can be obtained by identifying security relationships with additional system components, as follow:

- An isolation relationship exists when a component is partially or totally separated, i.e. isolated, by a second component, from other components inside or outside a system, meaning that the second component behaves as a partial or total isolator of the first one, respect to other components. An isolation relationship requires for both components be from the physical world. This relationship can be identified in the whole-parts tree in Fig. 1b), and the description about partial or total isolation between components can be included as additional properties stored at each node of the tree.
- An interaction relationship exists when two components interact or communicate in any way. The function or communication purpose of the interaction is not relevant for the security process, but only the identification of the components participating on the interaction. Interactions are enabled between pairs of physical or digital components, but they don't mix with each other. Interaction relationships are identified from the interaction diagrams, as described in Fig. 1c).
- A representation relationship exists when a component acts on behalf of another component. Components participating on a representation relationship can be of any type, namely physical or digital without restrictions because they enable the joint between different categories of components, such as a digital Id for representing a physical person. Representation relationships are shown in the table of Fig. 1d).

Security relationships are bi-directional. If a relationship is identified from component A to component B, there is also a security relationship from B to A, as a way to achieve integral security for both components. Then, security relationship handling is given in pairs, with one exception. If one of the components is external to the system, only the security objective for the internal component is part of the security process for that system.

Intuitively, a security relationship between components A and B allows for identifying a potentially additional (indirect) security requirement for component B given a (direct) security objective for component A, to support the objective established for A. An isolation relationship shows the isolation a component provides to another component. An interaction relationship means that, for the pair of interacting components, if we have a security objective for one of the components, then we should check for an additional security objective for the other component. A representation relationship shows an implicit security requirement, as opposed to explicit, if one of the components represents the other in some way.

Table 1 shows the security relationships for `Cup-Package` system. Lines 1–4 show isolation relationships between each cup (`Cup1`, `Cup2`) and the `Container`. Lines 1 and 3 show that given a security objective (integrity and availability) for each cup, it worths to consider the security (also integrity and availability) for the `Container`. On the other side, lines 2 and 4 show that given a security objective for the `Container`, there is no contribution if we consider the security of any cup to achieve the security of the container.

Isolation relationships can be described by storing isolation information that relates each cup and the container. Then, when considering automation of the security process, these relationships

Table 2
Cup-package - complete (direct and indirect) objective list.

#	Objective	Component	Service	Description	Rationale
1.	DO[3][1]	Cups	Integrity	We want to prevent the cups from being broken while transported to their destination	Direct objective
2.	DO[3][2]	Cups	Availability	We want to prevent the cups from being lost while transported to their destination	Direct objective
3.	DO[1][1]	Cup 1	Integrity	We want to prevent the cup 1 from being broken while transported to its destination	Group objective DO[3][1] was propagated downward
4.	DO[1][2]	Cup 1	Availability	We want to prevent the cup 1 from being lost while transported to its destination	Group objective DO[3][2] was propagated downward
5.	DO[2][1]	Cup 2	Integrity	We want to prevent the cup 2 from being broken while transported to its destination	Group objective DO[3][1] was propagated downward
6.	DO[2][2]	Cup 2	Availability	We want to prevent the cup 2 from being lost while transported to its destination	Group objective DO[3][2] was propagated downward
7.	DO[10][1]	Cup-Package	Integrity	We want to prevent the Cup-Package from being broken while transported to its destination	Group objective DO[3][1] was propagated upward
8.	DO[10][2]	Cup-Package	Availability	We want to prevent the Cup-Package from being lost while transported to its destination	Group objective DO[3][2] was propagated upward
9.	IO[4][1]	Fragile Label	Integrity	We want to prevent the Fragile Label from being damaged while transported to its destination	For supporting objectives DO[1][1], DO[2][1]
10.	IO[4][2]	Fragile Label	Availability	We want to prevent the Fragile Label from being removed while transported to its destination	For supporting objectives DO[1][1], DO[2][1]
11.	IO[5][1]	Content Label	Integrity	We want to prevent the Content Label from being damaged while transported to its destination	For supporting objective DO[3][1]
12.	IO[5][2]	Content Label	Availability	We want to prevent the Content Label from being removed while transported to its destination	For supporting objective DO[3][2]
13.	IO[6][1]	Container	Integrity	We want to prevent the Container from being broken while transported to its destination	For supporting objectives DO[1][1], DO[2][1], IO[4][1], IO[5][1]
14.	IO[6][2]	Container	Availability	We want to prevent the Container from being lost while transported to its destination	For supporting objectives DO[1][2], DO[2][2], IO[4][2], IO[5][2]

could be identified automatically if enough characterization of the components is stored inside the nodes of the whole-parts tree. The establishment the second objective (i.e. the Service 2 column and its corresponding Objective 2 column) for the second component of the relationship involves a security decision, about establishing or not the second objective, which should be decided by the developer of the security process.

Lines 5–17 of Table 1 follow a similar reasoning for interaction relationships, and they are obtained by iterating for each interaction of each component, from the interaction diagrams shown on Fig. 1c). Integrity focuses on the damage of the component, as opposed to availability that focuses on the lost of the component. Interactions happen between the cups (`Cup 1`, `Cup 2`) and the `Container`. Each label only interacts with the `Container`. Interaction relationships unveil considerations for protecting the cups, but they also show an additional indirect requirement for protecting the labels, because they provide information about the number of contained cups, and their fragility.

Lines 18–23 of Table 1 show details of the representation relationships on Fig. 1d). `Content Label` represents the size of the `Cups` group, and `Fragile Label` represents the fragility property of each cup. Representation relationships unveil an additional requirement for protecting the labels. If any of the labels get damaged or removed, it may increase the possibility of transporting the cups without fragility considerations. A modified or removed `Content Label` could lead to a lost of the contained cups.

We can create the complete list of objectives, as shown on Table 2, by putting together the direct objectives (DO[x][y]) with the indirect objectives (IO[x][y]) obtained from the column Objective 2 on Table 1. Now, there is confidence for moving towards a risk identification and analysis step.

3.3. Risk assessment

Risk assessment is realized as proposed in ISO/IEC 27005:2018 standard (ISO, 2018b) with three main activities, namely risk identification, risk analysis, and risk evaluation. Risk identification consists of identifying applicable vulnerabilities and threats, regarding to the set of defined security objectives. This way, every security solution will be aligned with real but also relevant requirements.

Table 3 shows the identified vulnerabilities and threats for the objectives defined in Table 2. Last column includes a rationale explaining how direct and indirect objectives are related.

How different objectives relate to each other is an additional but relevant result that we call security chains, because they behave as chained security requirements. Security chains are explained later, and they are the way for doing integral and comprehensive security for a system.

Risk analysis and evaluation can be done in several different ways, through some risk analysis methodology available in the industry. For the `Cup-Package` example, we took the decision of mitigating every identified risk, or at least to propose some security consideration. This decision is appropriate because the goal of this example is to explain how the security framework behaves, independently of any risk analysis approach. Table 4 shows the identified risks, and all of them will be mitigated to some extent.

3.4. Security requirements and controls

Last part of the security process is to define security requirements for mitigating the selected risks, as shown in Table 4. A security requirement establishes what to do for risk mitigation. Moreover, security controls allows to enforce the requirements. Table 5 shows the security requirements and controls for the `Cup-Package` system.

The obtained security requirements and the controls are integral but also comprehensive regarding to the initial security objectives. The security controls included in the initial system, such as the container and the labels, are implicitly considered because the security process associates them to the objectives.

The security process also considers missing, or complementary controls, while trying to get an integral security. For example, requirements 1 and 2 (related to the cups) from Table 5 are established to satisfy initial direct objectives, but requirements 3 and 4

Table 3
Cup-package - vulnerabilities and threats by objective.

#	Objective	Component	Service	Vulnerability	Threat
1.	DO[1][1]	Cup 1	Integrity	Cup 1 is fragile and may break	Container or Cup 2 may hit Cup 1 during transport
2.	DO[1][2]	Cup 1	Availability	Cup 1 may be lost/stolen	Somebody may take/steal the cup
3.	DO[2][1]	Cup 2	Integrity	Cup 2 is fragile and may break	Container or Cup 1 may hit Cup 2 during transport
4.	DO[2][2]	Cup 2	Availability	Cup 2 may be lost/stolen	Somebody may take/steal the cup
5.	DO[3][1]	Cups	Integrity	Any cup(s) may break	Cups may hit each other or with the Container during transport
6.	DO[3][2]	Cups	Availability	Any cup(s) may be lost/stolen	Somebody may take/steal any cup(s)
7.	DO[10][1]	Cup-Package	Integrity	Cup-Package may be damaged	Somebody/something may hit the whole package during transport
8.	DO[10][2]	Cup-Package	Availability	Cup-Package may be lost/stolen	Somebody may take/steal the whole package
9.	IO[6][1]	Container	Integrity	Container may be damaged	Somebody/something may hit the Container during transport
10.	IO[6][2]	Container	Availability	Container may be lost/stolen	Somebody may take/steal the Container
11.	IO[5][1]	Content Label	Integrity	Content label may be modified or damaged	Somebody/something may damage or change value of content label during transport
12.	IO[5][2]	Content Label	Availability	Content label may be removed	Somebody/something may remove/destroy the content label
13.	IO[4][1]	Fragile Label	Integrity	Fragile label may be damaged	Somebody/something may damage or leave the fragile label illegible during transport
14.	IO[4][2]	Fragile Label	Availability	Fragile label may be removed	Somebody/something may remove/destroy the fragile label

Table 4
Cup-package - risks by objective.

#	Objective	Component	Service	Risk
1.	DO[1][1]	Cup 1	Integrity	Cup 1 may break because Container or Cup 2 may hit it during transport
2.	DO[1][2]	Cup 1	Availability	Cup 1 may be lost/stolen because somebody may take it out of the box
3.	DO[2][1]	Cup 2	Integrity	Cup 2 may break because Container or Cup 1 may hit it during transport
4.	DO[2][2]	Cup 2	Availability	Cup 2 may be lost/stolen because somebody may take it out of the box
5.	DO[3][1]	Cups	Integrity	Any cup(s) may break because they hit each other or with the Container during transport
6.	DO[3][2]	Cups	Availability	Any cup(s) may be lost/stolen because somebody may take them out of the box
7.	DO[10][1]	Cup-Package	Integrity	Cup-Package may be damaged because somebody/something may hit the whole package during transport
8.	DO[10][2]	Cup-Package	Availability	Cup-Package may be lost/stolen because somebody may take the whole package
9.	IO[6][1]	Container	Integrity	Container may be damaged because somebody/something may hit it during transport
10.	IO[6][2]	Container	Availability	Container may be lost/stolen because somebody may take it
11.	IO[5][1]	Content Label	Integrity	Content label may be modified if somebody/something damage or change the content value during transport
12.	IO[5][2]	Content Label	Availability	Content label may be removed/destroyed by somebody/something
13.	IO[4][1]	Fragile Label	Integrity	Fragile label may be damaged or left illegible by somebody/something during transport
14.	IO[4][2]	Fragile Label	Availability	Fragile label may be removed/destroyed by somebody/something

(related to the labels and the container) support the security, because they aim to mitigate risks not directly but related to the security defined in the initial objectives.

Next section define and describe the Security Framework supporting the `Cup-Package` example. The main security concepts and contributions of this research are also presented.

4. Security framework

We define a Security Framework as an architecture-based fundamental structure, for supporting the execution of a security process on a system.[1] The framework is described by using the key concepts of an architecture description, according to ISO (2011), including but not limited to the terms stakeholders, concerns, viewpoints, model kinds, and models.

[1] The term *entity* or *entity of interest* may be used for a more generic application of the framework, but *system* is used instead for more clarity into the cybersecurity context.

Security concerns of stakeholders presented on Section 1 are reflected in the proposed architecture with three integral and complementary viewpoints, namely:

- **system viewpoint** for describing or representing the system of interest (i.e., the target of the security),
- **security viewpoint** for describing the security goals and related aspects, and
- **process viewpoint** for establishing a clear methodology to define the security of the system.

The framework comprises a security basis, and three model kinds for supporting each corresponding viewpoint with generic models, that can be used or applied to specific security scenarios.

Security basis is its nature and its origin, because it helps to answer questions such as what security is, where it is found, when we need it, and how to use it. Nature of security approach aims to get a precise, clear and complete specification of the security, related to a well defined set of security objectives on a system, that way responding to real security requirements.

Table 5
Cup-package - security requirements and controls.

#	Objective	Component	Service	Requirement	Control
1.	DO[1][1], DO[2][1], DO[3][1]	Cup 1, Cup 2, Cups	Integrity	Every cup **must** be protected from hitting against other cups (or any other object) inside the box, or against the box itself, while being transported to its final destination	Install a soft padded protection around every cup, such that they do not hit to each other, and they do not move or hit the box
2.	DO[1][2], DO[2][2], DO[3][2]	Cup 1, Cup 2, Cups	Availability	Cups **must** be protected so that they cannot be taken out of the box, while being transported to its final destination	Install a lock in the box, proportional to the level of robustness of the container
3.	IO[5][1], IO[5][2], IO[4][1], IO[4][2]	Content Label, Fragile Label	Integrity, Availability	Labels (Content and Fragile) **should** be protected from breaking or being removed from the box, accidentally or intentionally, or they should be replaced by labels that cannot be easily removed.	Install a protection to the labels, such as a transparent adhesive tape that is suitable for installation in the box, or replace the labels with different material that cannot be easily removed, for example, ink labels.
4.	IO[6][1], IO[6][2], DO[10][1], DO[10][2]	Container, Cup-Package	Integrity, Availability	The carrier **should** have insurance or a contract for the transport service that ensures integrity and availability of the package during transport of the package to its final destination	Sign insurance or a transport contract that includes the integrity and availability of the components to be transported

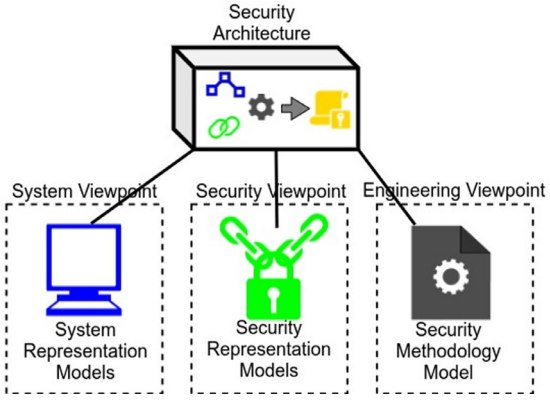


Fig. 2. Security architecture overview.

The model kinds specified by the viewpoints are shown on Fig. 2. First model kinds are for system representation. Second model kinds are for security representation, to support the definition of the security objectives to be achieved. Third model kind is for representing the methodology of the security process. A preliminary description of them follows.

- **System representation model kinds** provide a description of a system, with its components and interactions. Information included in the representation focuses on the properties of components that contribute to the security process.
- **Security representation model kinds** provide a description of security objectives and security relationships between components of a system. Security objectives are a set of statements to describe what should be done about security. Security objectives may consider organization policy and objectives. If organization policy and objectives are not available, they may be directly obtained from a rationale and established from security services (such as integrity, confidentiality, availability and more), with regard to the components of the system. Additional security objectives can be derived in a systemic or even automated way, based on the initial objectives and the security relationships between the system components.
- **Security process model kind**, or simply **security process**, is defined as a series of steps, in order to define and implement a set of security controls, for making a system more secure regarding to initial security objectives. The security process defines a step by step methodology for constructing the security of a system.

Following sections provide a detailed description of the three model kinds.

5. System representation model kinds

A system representation is a description of a system, including components and their interactions, to support a security process. By **system** we mean a set of things working together as parts of a mechanism or an interconnecting network (Oxford University Press, 2010). Examples of systems include small electronic devices like smart cards, software applications of any type, computer systems and networks with different sizes, communication and security devices, specialized-device applications, virtualization platforms, modern data centers, large infrastructure systems of Internet Service Providers (ISP), distributed infrastructure of global companies around the world, spatial technology systems, complex IT systems, system of systems, and more.

A system representation can become complex, even for medium sized systems, then it should be generic enough and scalable to be applied to systems of different types and sizes. Aspects of the system not related to security are not be included in the representation. For example, a sequence of interactions (like a conversation

or protocol) describing a functionality between two components is not represented in detail, but only a single interaction to describe the security aspects of that functionality.

A system representation has a basic abstraction called **node**, to represent a single component of the system. Most operations of the security process are orchestrated around the node. Building on the node, the system representation has two larger abstractions, namely:

- **Components representation** is an abstraction to represent all the components and entities participating in the system. The components representation is implemented as a **whole-parts tree structure**, by using a hierarchical graph to show the organization of a system with its parts.
- **Interactions representation** is an abstraction to represent the functions/operations/actions realized inside or by the system. The term interaction emphasizes a generic relationship between pairs of components, as opposed to a more specific type of function or relationship. The interactions representation is implemented with **interaction diagrams**, by using undirected graphs, to show how the components and entities of the system interact with each other.

Following, there is a detailed description of a node, as well as the whole-parts tree and interaction diagrams.

5.1. Node representation

A node is a data structure representing a component of a system, at any level of detail. For example, a node can represent the entire system or any of its parts. It is also a structure to keep most of the relevant information required when managing a security process. A node includes a description of the component. There is also a category for classifying the component to support the definition of security requirements. For example information, software, physical device, physical infrastructure or people, are valid categories.

A node is also the basic working unit to create larger data structures, such as a whole-parts tree to contain all components of the system, or interaction diagrams to represent the dynamic aspects of the system. Information used for implementing the security representation is also stored into the node.

The following list describes basic elements for a node structure. Additional elements may be defined when developing an automated version of the security process.

- **Basic information** contains an identification, name and short description of the component being represented, a type or category of the component, intrinsical characteristics of the component for identifying vulnerabilities, threats and risks. There are also structural characteristics of the component respect to its position into the system, and any other information related to the description of the component that may be appropriate to support the security process.
- **Security Objectives** allow to establish what should be done about security for the component. It contains a list of direct or indirect objectives as described in Section 6.2.3, using the format defined in Section 6.2. Every objective is expected to have a name, a description, what is the target security service, where is the security service required (usually the node itself or a set of nodes containing it), when the security service should be managed, altogether with additional complementary information for supporting the objective.
- **Security Representation** contains a list of security relationships for the component, i.e. existing relationships with other components for security purposes. The security process of Section 7 explains how to obtain a full security representation. Every security relationship contains the security objectives for the related components, the type of relationship, and any other complementary information to describe the relationship.
- **Vulnerability information** contains a list of vulnerabilities associated with the component. A vulnerability usually has an identification, name, short description, full description (intended audience, credit, contact information, revision history), affected products, severity, impact, remediation, and event time or relation with the life-cycle of the system.
- **Threat information** may include a name, a description, vulnerabilities that may be exploited by this threat, and adverse actions that may be taken against it.
- **Risk identification and assessment** is realized when vulnerabilities are presented to threats, and according to a previously defined set of security objectives. A risk definition may include a name, the related vulnerabilities and threats, its likelihood, impact, management context, recommendations for mitigation, location, time interval, and more.
- **Security requirements** describe what to do for mitigating the identified risks. They may include a name, description, the target security service, the life-cycle relationship, and any other detail describing the requirement.
- **Security controls** describe how to support the security requirements. They may include a name, a description of the control, the target security service, a categorization, what or who is going to execute the control, and any other information related to the implementation or management of the control.

5.2. Whole-parts tree structure

A whole-parts tree is a hierarchical data structure representing the components of a system. It describes the system components by breaking it down into its parts. A whole-parts tree is organized with an initial root node, representing the entire system, and a set of children nodes to represent its parts. Each child node can be a simple leaf node, or it can be a composed node. Each composed node can be recursively described as having a set of children nodes, representing its sub-parts, and so on.

Nodes of the tree represent objects, components or entities that belongs to or participate into the system. Each child node have only one directly connected parent node, except for the root node that does not have a parent. The path from the root node to a child shows all the ancestors from which that child is (direct or indirectly) a part.

Each component (node) may be described by its parts with at least one level of granularity, but it can go deeper depending on the granularity of the required security for the component.

Grouping of components is the only allowed abstraction for defining a node of a whole-parts tree. A group of components that shows the composition of a system part is appropriate for describing a node of the tree.

A useful and interesting property of the proposed tree structure is that, if we had a system represented by a uniform depth tree, each level of the tree would be a full representation of the entire system, but described with different levels of granularity. This property yields a powerful tool for establishing general, specific, or even multiple simultaneous security controls, with different levels of granularity, like in a layered layout of the system.

Nodes of the whole-parts tree represent the composition of the system, but they can be also used to describe the isolation security relationships, for physical components, required for a security process, as explained in Section 6.

5.3. Interaction diagrams

An interaction diagram is a representation of interactions between the components of a system. An interaction represents two objects having some contact of any type, while executing a function of the system. An interaction happens between two internal components of the system, or between a component into the system with an external entity, i.e. an entity not being part of the system. Interactions between two external entities, that do not belong to the system, are not considered part of the security process and consequently are not represented in any diagram.

An interaction is a good abstraction for constructing security because it is generic enough to avoid any miss-interpretation between the security and other specific functionalities that does not contribute to the security process, such as a dependency relationship, a specialized functional action, among others. Furthermore, an interaction allows to identify security requirements for the source, the target or both ends of the interaction.

Nodes of an interaction diagram must belong to a related whole-parts tree. External entities are the only nodes not belonging to the whole-parts tree that can be used on interaction diagrams, to show interactions from components of the system with the external world. An interaction diagram also describes communication channels between pairs of components, as a consequence of their functional relationships in the system.

Each interaction diagram may represent all the interactions for a single component, but it may also represent a use case of the system. Then, there could be multiple (probably many) diagrams to represent all interactions in the system.

An interaction diagram is implemented as an undirected graph, where nodes represent components of the system, and edges represent the interactions or communication channels.

Interactions and communication channels are different from each other. Interactions are defined between pairs of components, and they interact through the function they execute. Communication channels show a third moving component, like information or a virtual resource, while being transferred between a pair of communicating nodes.

While an interaction is represented as a single edge between two nodes, a communication channel may be represented in two ways. It may be represented as a single edge between the nodes like a normal interaction with the moving component as a label for the edge, or it can be drawn with a new additional node for representing the component moving between the communicating nodes, when the channel requires a more detailed security description.

Finally, interaction diagrams provides de input for identifying the interaction relationships required by the security process, as explained in Section 6.

5.4. System representation rationale

This section describes the benefits of using a whole-parts tree structure and interaction diagrams for modeling a system representation.

- The whole-parts tree and interaction diagrams are complementary to each other, by representing components and functions from a security perspective. They support a consistent and complete security solution.
- Heterogeneous types or categories of components can be represented, to provide strong support for complex security scenarios, like in cybersecurity. This model allows different types of components, such as digital and physical components, that way considering the (security) representation relationships for enabling the bridge between the digital and the real physical world, during the security process.
- Components can be described with different levels of granularity, to adapt for general security, or very specific security requirements.
- The model is fully scalable because of the recursive layout of the tree structure. It can be applied from small specialized systems, such as a simple cryptographic card or a mobile device, up to complex systems such as distributed data center infrastructures located around the world, or SoS.
- The model supports a natural way for identifying security requirements (i.e. the security relationships), with the corresponding benefits, as explained in Section 6.
- The whole-parts pattern is related to mereology (Calosi and Graziani, 2014), favoring the implementation of formal operations in the model.
- The model supports modular and flexible definition of security for specific components. It allows identification of components belonging to the system but not being included in the boundaries of the security process. It also allows to represent external components, not belonging to the system but interacting with the system.
- The model dynamically supports system changes, such as added or removed interactions, or new additional security requirements.
- Interaction diagrams expose security relationships, and puts aside other types of existing relationships, such as functional operations, non-functional relationships or dependencies, that do not contribute for identifying or solving the security problem.
- Interaction diagrams allows to represent simple interactions but also communications channels.
- The system representation is enabled for security of complex systems including virtual or real resources, processes, virtual machines or real-world assets. This property allows to precisely compare differences of security levels between real vs virtual systems.
- The model is enabled for components moving or migrating between other components of the system, and requiring highly dynamic security solutions, as it happens in environments of cloud computing.

Next section describes the security representation model kinds, which complement the system representation by providing the security related data structures required for the security process.

6. Security representation model kinds

A **security representation** is a description of security objectives altogether with security relationships for the components of a system. Security objectives are a backbone for guiding what should be done during a security process, and the entire security representation is a backbone for the methodology guiding how to execute the process.

A security representation produces a set of directly defined security objectives, for selected components of a system. But it also allows to derive indirect security objectives, obtained from security relationships between system components, to support an integral and comprehensive security process.

This section starts with a description of how security requirements arise. We call that the nature of security. Security objectives are described in detail, followed by security relationships, and security chains as a derived result of the security relationships.

6.1. Nature of security

We call the **nature of security** to the source or origin of the security needs. It is described by the concepts isolation, interaction and representation.

Isolation is the condition of being alone; in the model it is related to a component that is partially or totally separated by a second component from other components inside or outside the system.

Interaction is referred to things that communicate with or react to each other; in the model it is related to pairs of components that communicate, or execute a function in which they react to each other. The function itself is not relevant, but the interaction.

Representation is a way for someone or something to be shown or described; in the model it is related to how a component is partially or totally described by another component.

Nature of security give rise to the system representation, i.e. whole-parts tree and interaction diagrams, presented in Section 5. Then, system diagrams provide specific ways to look for security, regarding to security objectives.

6.2. Security objectives

A **security objective** is a proposal to set up security services for components or entities belonging to a system, at some points or intervals over time. Security objectives state what should be done about security, and they are the backbone of any security process. They behave like a security plan, and also become a reference point to establish and evaluate (assure) the implemented security controls.

Theoretically, security objectives may be initially obtained from organizational objectives. They may derive from the main guidelines to be met by the systems of an organization. Security has a clear purpose when security objectives are supported by organizational objectives, and this may help to reduce unexpected security events and their impact, as proposed by Goman (2018).

When organizational policies are not available, initial security objectives should be, at least, in line with business expectations, to avoid unexpected results on the security. Incorrect or unnecessary security objectives may lead to the implementation of unnecessary security controls, waste of resources, but also the lack of relevant security controls.

A security objective should be specified as a structured sentence, containing a clear security goal, but also indicating where and when it should be fulfilled. This structure should be preserved for every objective defined in the system. In the model, a security objective contains the following elements:

What has to be done about security, meaning what security service is being required. This is the main contribution of the security objective to a security process. A list of security services with corresponding descriptions is shown in Section 6.2.1.

Where the security service is required, related the component(s) of the system where the security process is to be executed. The component(s) referenced by an objective have to be a subset of the components included in the whole-parts tree.

When the security service is required, for referencing a point or time interval when the security service is required, as described in Section 6.2.2.

Additional elements that complement the specification of the objective. The elements comprise a variety of things, such as specifying a subject when defining a security service for an object of the system (i.e. a component referenced in the where clause of the objective).

Following, there is a description of security services and the time dimension for security events. Later, a definition of direct and indirect objectives introduces the section for security relationships.

6.2.1. Security services

Most security models use different terms (Bishop, 2004; Cherdantseva and Hilton, 2013; Maconachy et al., 2001; McCumber, 1991; Stallings and Brown, 2012) to describe what is provided by security, such as critical information characteristics, security goals, security aspects, security services, security objectives, among others.

We selected the concept **security service**, as defined in ITU (1991), to describe the key aspects that ensure a suitable security. Following is a non-exhaustive list of security services with a corresponding description, that can be applied to the components of a system during a security process.

confidentiality property of data that indicates the extent to which these data was not available or disclosed to unauthorized individuals, processes, or other entities (ISO, 2015).

integrity property of accuracy and completeness (ISO, 2018a).

availability property of data or resources being accessible and usable on demand by an authorized entity (ISO, 2015).

non-repudiation ability to prove the occurrence of a claimed event or action and its originating entities (ISO, 2018a).

privacy freedom from intrusion into the private life or affairs of an individual when that intrusion results from undue or illegal gathering and use of data about that individual (ISO, 2015).

authentication provision of assurance that a claimed characteristic of an entity is correct (ISO, 2018a).

authorization is the specification of access policies (Josang, 2017), that are enforced with access controls. Access control means to ensure that access to assets is authorized and restricted based on business and security requirements (ISO, 2018a).

audit systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled (ISO, 2018a).

Security services are strongly related to the nature of security. A formal rigorous definition of security services in terms of the three core security concepts proposed in Section 6.1 is left for future work.

6.2.2. Time of security events

The time of security events is a complementary element of a security representation, and a key aspect to define security objectives. Event times are represented as time lines, relative to the occurrence of the security events. Time slots are used to represent the moments before, during or after the occurrence of an event. The event time allows to use different strategies for dealing with security during the system life-cycle.

Event time intervals are the time dimensional complement for the system representation of Section 5. Time intervals are relative to the occurrence of an event. They are defined as the time to prevent, detect, or recover from a security event, as follow.

Prevent is the time interval before the occurrence of a security event, i.e., the time to act before something outside security objectives happen.

Detect is the time interval during the occurrence of an in-progress security event, i.e., the time to usually stop or block an on-going event.

Recover is the time interval after the occurrence of an event, for identifying or repairing damages, i.e., the time to restore the system to its normal state.

Time intervals are a main component of the security objectives. They are also considered as a strategy for risk assessment during the security processes.

6.2.3. Types of security objectives

We categorize security objectives in two types, depending on how they are established and defined during the security process, namely:

Direct Security Objective is a security objective established by those responsible of a component or subset of components in a system, by means of a direct statement, according to the format defined in Section 6.2.

Indirect Security Objective is a derived security objective. It is obtained during the security process, to support the achievement of a direct security objective. The main goal of indirect objectives is to complement the achievement of an integral security process. Indirect objectives support the direct objectives, by securing additional components that have security relationships with the direct target component.

6.2.4. Security objectives rationale

The proposed approach for security objectives has the following benefits during a security process.

- Security objectives are supported by organizational policies and objectives, i.e. business goals, which increases the probability of a successful security process, because security controls will focus on real requirements.
- The structured format (what, where, when) of the objectives supports definition of security goals in space and time.
- Security objectives apply to any type or size of components, described by the whole-parts tree and interaction diagrams, enabling diversity and all granularity levels for dealing with security.
- Comprehensive and automatic identification of indirect objectives is possible, with support of the security relationships defined in Section 6.3.
- The proposed security service list is appropriate for assessing and management the security of most modern complex systems, such as government or corporate infrastructures, cybersecurity scenarios, SoS and more.

6.3. Security relationships

A **security relationship** is a relationship of security requirements between a pair of components. It is characterized by a first component having a direct security objective, and a second component where an additional (indirect) security objective may be defined.

The second objective is related to the achievement of the first one, then it is a complementary goal looking toward an integral security solution. The relevance of the second objective will always depend on a decision about mitigating the risks associated with the second component.

How the second objective supports the first one depends on the type of security relationship between the components. There are three types of security relationships, and they derive from the three sources of security needs:

- **Isolation relationship**: one component is related to another component because of their structure or location into the system.
- **Interaction relationship**: one component is related to another component because they interact in any way.
- **Representation relationship**: one component is related to another component because one of them behave as a representation of the other.

A security relationship could be created when adding a new component to the system, such as adding a new security control for securing a existing component(s). In this case, the second security objective, corresponding to security for the new component and for supporting the security of the first component, may follow a regular assessment process to mitigate its associated risks, or it may be associated with residual risks.

6.3.1. Isolation relationship

An isolation relationship is an explicit structural security relationship between two physical components of a system, where a component is partially or totally separated, i.e. isolated, by a second component, from other components located inside or outside a system. The second component behaves as a partial or total isolator of the first one, respect to other components.

Common scenarios of isolation relationships between two components are:

- when a component is or behaves as a container of another component.
- when a component is located next to other and behaves as a divider or separator element, but not necessarily as a container.

Isolation relationships depends on whether the required security service can take advantage or not from an isolation structure in the system. Examples of components having an isolation relationship are: an electronic device and its corresponding enclosing case, a hard disk and the contained magnetic-fields, a fiber-optic cable and the contained light signal, among others.

Isolation relationships may be systemically identified in the whole-parts tree of a system, if enough descriptive information about the organization of the tree components is provided.

6.3.2. Interaction relationship

An interaction relationship is an explicit security relationship between two components of a system. They are related when they interact in any way with each other, because of their function into the system. Given two components there are countless ways for interacting, so any type of activity that involves both components should be considered an interaction.

Interaction is the key concept behind interaction security relationships. Examples of interaction relationships are a person and a keyboard for providing information to a software application, two processes communicating remotely through a network, a function calling another function in a software application to execute some action.

Interaction security relationships may be systemically identified from the interaction diagrams of a system, by iterating for every interaction of every node for identifying additional security objectives.

6.3.3. Representation relationship

A representation relationship is an implicit security relationship between two components of a system. They are related because both elements represent to each other in several different ways. The scope of the representation may be variable because there are many ways for defining or expressing a representation relationship.

Representation examples are a person and its identifying information, an electronic relay and its logical status stored in the

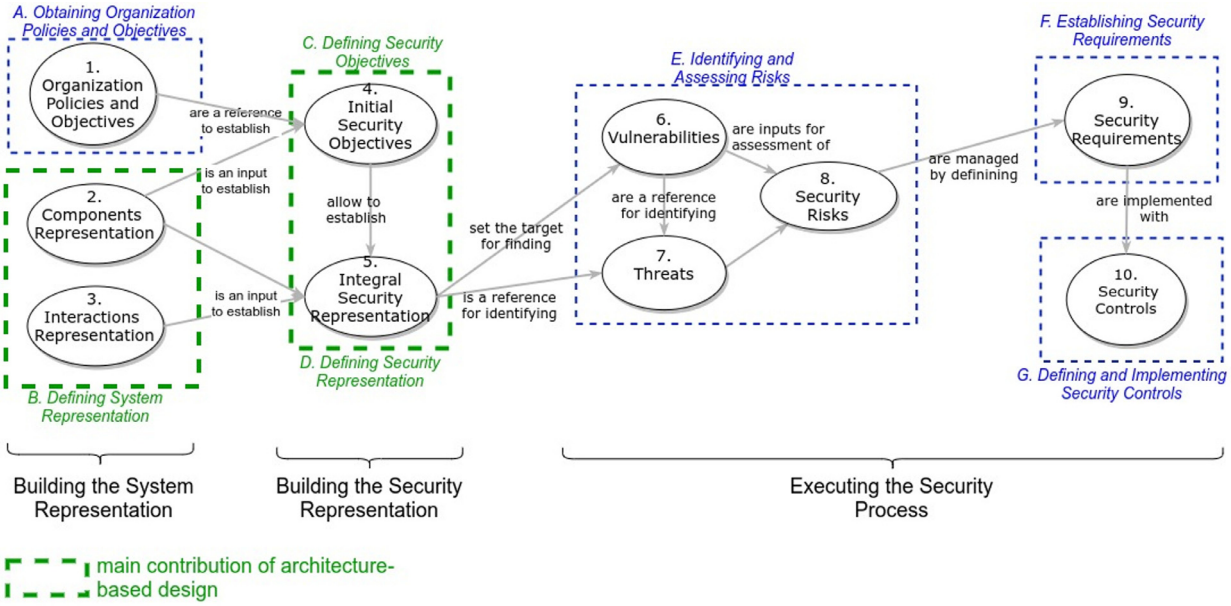


Fig. 3. Security process model - conceptual map.

memory of a controlling software application, real money and its equivalent value stored in a digital device, a clear text information and its encrypted version.

Due to the implicit characterization of representation relationships, they must be individually established between pairs of components of a system, but they could be systemically identified and managed during a security process, if enough information is provided about the system components.

6.3.4. Security chains

Security relationships produce chains when two or more components are related by security objectives. A **security chain** is the representation of an integral security of an initial component and its security relationships. Given a direct security objective for a component, a security chain shows the list of expected security objectives for additional components in the system, related to the security of the first component, to enforce an integral solution for the initial objective.

A full set of security chains for an entire system, i.e. the list of direct security objectives and their corresponding indirect objectives, describes a **comprehensive and integral security representation**, and it can be used to implement a complete security process for the system.

6.4. Security representation rationale

A security representation is a novel way for modeling security. This section provides reasoning for justifying the new approach.

- The security basis arise from properties for the components of a system, allowing to identify real requirements based on objectives.
- The approach provides a representation of security that is independent from other types of requirements for a system, such as functional or other non-functional requirements, but at the same time it is coherent with them for the entire life-cycle of the system.
- Security relationships and chains allows to specify integral and comprehensive security for a system, with an organized and potentially automatic methodology.
- For complex systems, the security specification supports a modular assignment of security responsibilities by parties, according to the system structure.
- For systems of any size, the security representation is algorithmically enabled to support all stages of the life-cycle of the system, including but not limited to development, production, monitoring, management, or improvement.
- The timing of security events, altogether with the other data structures, enables managing dynamic security for changing systems, even when security requirements change over time.

7. Security process model kind

The Security Process is a methodology for executing a security process for a system. A security process usually starts by collecting the business policy, accompanied with organization objectives that support the policy. High-level security objectives may be established from organizational objectives. The security objectives altogether with a system representation are used to obtain a full security representation for the system, with the purpose of guiding a risk assessment process. The risk assessment produces a set of security requirements for mitigating the selected risks. Finally, security controls can be defined and implemented to enforce the requirements.

Fig. 3 shows a conceptual map describing the methodology. Green dashed boxes enhance the main contribution of the architecture-based approach to the security process. The methodology steps are explained following.

A. Obtaining Organization Policies.
B. Defining the System Representation.
C. Defining Security Objectives.
D. Defining Security Representation.
E. Identifying and Assessing Risks.
F. Establishing Security Requirements.
G. Defining and Implementing Security Controls.

7.1. Obtaining organization policies

Organization policies and objectives are the main guidelines for any business, and they can be useful during a security process for an IT system.

In our model, organization/business policies are different from the security requirements derived later in this methodology. Organization policies are goals defined by the business owner, as a desired target. There is not necessarily a formal process that guarantees its achievement, but the intention of the head to establish business goals.

On the other side, security requirements are derived from a rigorous risk assessment process, for achieving a set of security objectives. Security requirements are a result of risk assessment, for which a level of effectiveness is expected. This section refers to organization policies, as opposed to security requirements.

Collecting organization policies and objectives is a key initial step before doing a security process. Organization policies establish the main guidelines of the resulting security. If there are no available business policies, security objectives could be defined based on a rationale, but hopefully related to the business activity.

7.2. Defining a system representation

A description of the components belonging to the system is a requirement for security modeling. A list of components defines the scope of the security process. Furthermore, characterizing the components allows to identify and describe the target components under responsibility of the security. Characteristics of the components are stored into the nodes of the whole-parts tree, and the interaction diagrams. This section describes how to create the diagrams for representing the system.

7.2.1. Creating the whole-parts tree

Construction of the whole-parts tree starts by defining a root node to represent the whole system. The second level of the tree contains the main elements composing the whole system, usually the largest parts or groups of components. Next levels are the subparts or sub-components of the second level components.

The process of decomposing parts on its sub-parts can be repeated several times. Not necessarily all nodes at any iteration are to be decomposed. Then, on the resulting tree, not every branch has the same level of depth, and not every part of the system is described with the same level of granularity. A detailed decomposition produces deeper tree branches, and it enables more specific security requirements for the corresponding components. A more general description of components will produce less deep tree branches; therefore, security will be applied in a less specific way. So, granularity of the whole-parts tree will impact the rest of the security process, and also the level of detail for the resulting security.

Construction of the whole-parts tree may be done in several different ways, such as using a bottom-up or a top-down approach for describing the components. Some components may be described from general to specific or vice versa. Later, they can be combined to create the whole tree.

A whole-parts tree represents a real structure and composition of a system. No abstractions about the components should be elaborated when constructing the system representation, but the real composition and implementation, with the exception of defining groups. Groups are a natural and intuitive way of applying a whole-parts pattern. For example, the root node is a group representing all components of the system. Several instances of the same type of component, such as a virtual machine farm, may be also grouped for applying common security requirement to all of them.

The following guidelines can be helpful when constructing a whole-parts tree:

- Gather a list of the system components. Include components of any category, such as IT hardware, network components, software, virtual resources like virtual machines or virtualized objects, information, and also people as long as they participate as part ofthe system. In general, any component or entity that is considered as an internal or composing part of the system should be included in the tree.
- Describe every component by its parts, as desired or required, by creating a corresponding sub-tree. For every component include a characterization of its properties, and aspects related to its structure. Include information about physical components that behave as a separator or isolator, for example when a component is a container, or a boundary like a computer case.
- Check for correctness of the whole-part structures. There are common mistakes such as considering digital components (for example, operating systems and software applications) as containers. Software components or virtual resources are not containers but they interact with each other.
- Define groups of components as a tool for constructing the whole-parts tree. Groups are the only allowed abstraction when building the tree. They allow to represent a set of components as one element, for applying security requirements to the group.
- Components like information, software and other digital resources may have a special treatment. They could be considered more than once. For example, a software application may be stored into file in a hard disk, but it could also be a running process inside a computer with CPU and memory. Then, representation of some components may be duplicated to satisfy different security requirements at different locations of the system. They may also be represented inside communication channels, when dealing with the security of transferred components. Consider representation relationships for these components.

7.2.2. Creating interaction diagrams

Interaction diagrams are complementary to the whole-parts tree. We may identify internal interactions between pairs of components belonging to the system, or interactions between internal components and external entities.

Internal interactions are part of the security process. For interactions involving an external entity, only the internal side of the interaction is part of the security process. The external entity by itself, or the external side of the interaction, is not part of the security, and it is not considered.

Interaction diagrams take some concepts from the object-oriented methodology proposed by Jacobson et al. (1992), but they also have some differences.

- The scope of an interaction diagram can be like a use case from Jacobson's methodology. But the interaction diagrams are different because they are graphs, with nodes representing components and edges representing interactions, as opposed to the vertical bars (components) and arrows between bars (interactions) from Jacobson version.
- Interaction diagrams in our model are simpler. They only need one connecting edge between each pair of nodes for each set of security requirements, to represent interacting components, as opposed to one arrow for each possible interaction between each pair of components.
- Edges of interaction diagrams usually represent simple interactions, but they may also be communication channels to represent other components being transferred. When a communication channel has complex security requirements, it may become a node located between the communicating nodes, so security requirements may be specified to an appropriate level of detail.

Consider the following guidelines when constructing an interaction diagram:

- For the general case, an interaction diagram may be based on a use case, like proposed by Jacobson et al. (1992), but using the symbols and other details described in this section. There should be a use case for each functional scenario comprising the components with a set of security requirements. Include enough use cases to consider all interactions for all system components.
- A simpler version for representing all interactions of a system would be by creating a star graph for each system component. The node representing the component would be the center of the star, and all direct interactions with other components around it. All system interactions may be fully represented with this basic interaction pattern for each component. This way is not probably very intuitive, but it is a valid representation, then representation with use cases is the suggested option.
- Interaction diagrams should consider interactions with external entities, not belonging to the system. Interactions between internal components allow to establish internal security, but interactions with external entities allow to define security related to the external world.

7.3. Defining security objectives

Defining security objectives for a system is a requirement for the effectiveness of the security process. Direct security objectives are defined first, indirect security objectives are derived later, as explained in next section.

Direct security objectives can be obtained from organizational (i.e. business) policies and objectives, or they can be defined from a rationale, when no organizational policies are available. To define an initial set of security objectives, review the system components for targeting the ones being relevant or sensible to the policy guidelines. Then, establish objectives for the selected components by using the objective structure described in Section 6.2. With support of the whole-parts tree, it is possible to scroll down from the selected components (nodes) to associate the direct objectives to the sub-components in a more specific way, to establish additional (probably more specific) direct objectives.

After the initial direct objectives identification, the security process is ready for next step, to obtain a complete security representation.

7.4. Defining a security representation

A security representation defines indirect objectives for additional components, to support the direct objectives of the targeted system components. Indirect security objectives arise from the existing security relationships (isolation, interaction, representation) between the components of the system, as described in Section 6.

When chaining security relationships, every indirect objective of a relationship can become the direct objective for the next relationship (link) of the chain. An iterative process for each relationship, with support of the whole-parts tree and the interaction diagrams, allows to obtain all the security chains, and consequently an entire security representation. The complete list of direct and indirect objectives is the input for the assessing risks process.

7.5. Assessing risks

Assessing risks is a multi-step activity comprising risk identification, risk analysis, and risk evaluation, as described in ISO (2018b). Risk assessment identifies the applicable threats and vulnerabilities if they exist or can exist, as a way for identifying and assessing the risks of the previously defined list of security objectives.

7.5.1. Finding vulnerabilities and threats

The more technical part of a security process deals with finding vulnerabilities and threats for the targeted components of the system, in the context of the security objectives. Technical threats and vulnerabilities can be obtained from available online vulnerability databases, provider support systems, online communities, digital forums, and more.

A vulnerability with no associated threats may not be considered in the security process, but it could be identified and monitored for changes.

7.5.2. Identifying and assessing risks

The purpose of risk identification is to determine the causes of a potential loss, and to understand how, where and why a loss can happen, related to the security objectives, but also in the scope and boundaries of the whole-parts tree, and the interaction diagrams of the system.

Vulnerability and threat pairs can be used to identify risks. The risks can be selected or not for mitigation with different criteria, such as the probability of occurrence, the impact for the system, or by using any available methodology for risk evaluation.

Security controls installed before the risk analysis are included in the security process, because they are considered as part of the security chains. It should be noted that an incorrectly implemented or malfunctioning control, or a control being used incorrectly, can itself be a source of vulnerability.

7.6. Establishing security requirements

A security requirement (or security policy) is a statement about what is and what is not allowed in a system for supporting the security process (Bishop, 2018). Security requirements are the result of the risk analysis step. There are multiple ways for expressing requirements (Bishop, 2018), from low to high level and depending on the requirement domain.

For establishing security requirements, every security objective should be considered altogether with the corresponding risks, to define the desired actions for mitigation. It is usually common that multiple risks can be addressed with a single requirement.

7.7. Defining and implementing security controls

A security control is a method, tool or procedure to enforce a security requirement (Bishop, 2004). For defining and implementing security controls, every security requirement should be considered at least once. It is usually common that multiple requirements may be enforced with a single mechanism, but also multiple mechanisms could be used to enforce a requirement in a multi-layered way.

We described the model kind for a security process that can be applied to a broad number of IT environments. Following section show examples of real applications where the framework has been already applied.

8. Real-life framework applications

This section describes a digital signature infrastructure system that was used to support the creation and evaluation of the framework. The framework was also used to provide feedback for validating the security of that system.

At the end of the section, there is a summary of two additional infrastructure/software example applications, that were also worked during the construction and evaluation the framework. These applications are also real and running-in-production IT platforms.

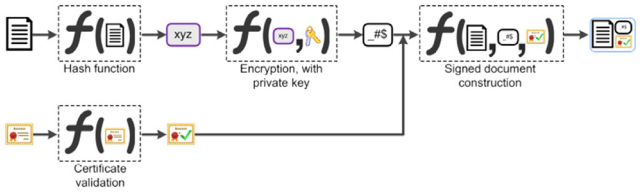


Fig. 4. Digital signature creation use case. Source: (Mora, 2017), English translated.

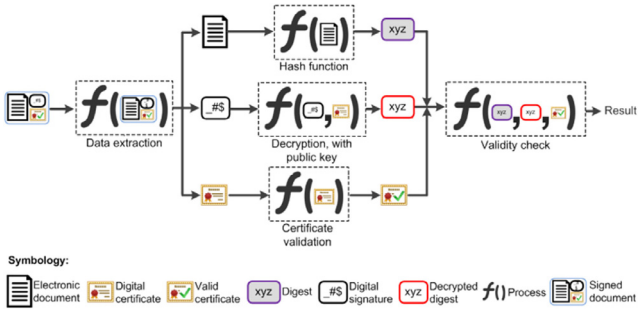


Fig. 5. Digital signature verification use case. Source: (Mora, 2017), English translated.

8.1. Software applications for the SNCD

The National Digital Signature Certification System (SNCD by its acronym in Spanish, Sistema Nacional de Certificación Digital) was created in Costa Rica for providing digital signature capabilities to the population at a national level. The security framework described in previous sections was used to provide academic support and for validating the security of software applications with digital signature capabilities into the SNCD.

The SNCD is owned by the Ministerio de Ciencia, Tecnologa y Telecomunicaciones. The Root and Issuing Certificate Authorities (CA) are hosted and managed by the Banco Central de Costa Rica. By request of the CA managers, a research academic project was started, with the goal of providing additional academic and technical support to the security of the SNCD (i.e., the digital certification system). The research included a graduate master project (Mora, 2017), a publication with preliminary results (Villalón-Fonseca et al., 2016), and a validation of the results with collaboration of the CA managers.

Following, there is a description of the main steps to support the security of software applications with digital signature capabilities running on the SNCD system, according to Mora (2017), and including additional considerations of the framework that were developed at a later stage of that research. The example follows the framework structure described on Sections 4–6, and the methodology of Section 7.

8.1.1. System representation

The system representation is constructed from a reference software application that comprises the minimum required operations needed for digital signature functionality. The reference application is derived from four use cases, namely:

- Digital signature creation (Fig. 4) for signing a digital document.
- Digital signature verification (Fig. 5) for verifying the signature of a digitally signed document.
- Digital signature conversion format (Fig. 6) for adding a certified time stamp to a digitally signed document.
- Authentication with digital signature (Fig. 7) for executing an authentication process using a digital signature as credential.

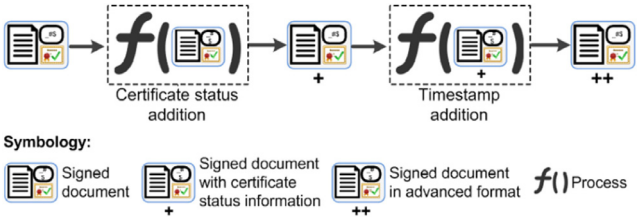


Fig. 6. Digital signature conversion format use case. Source: (Mora, 2017), English translated.

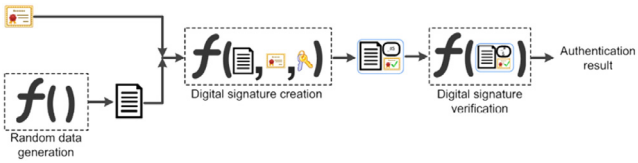


Fig. 7. Digital signature authentication use case. Source: (Mora, 2017), English translated.

Each use case is represented with a high-level work flow, describing the involved information and the action steps (indicated with *f()* functions), as shown on Figs. 4–7.

In addition to the uses cases, the reference application is modeled using the two physical architectures shown on Fig. 8, namely centralized (such as a standalone application) and distributed (such as a client-server application). These architectures allow to consider different risk evaluation options, for similar application functionality but different implementations, such as having local or remote communications between different software components.

The whole-parts tree and interaction diagrams created for the Digital Signature Creation Use Case of Fig. 4 are shown on Fig. 9. Similar diagrams are available for the remaining use cases.

8.1.2. Security representation

The security representation starts with a high-level goal, to ensure the non-repudiation of the digital certification system (i.e., the SNCD) through ensuring non-repudiation of the digital signatures and the related operations being managed by the software applications.

The main direct objective can be propagated to specific system components, by considering components of the four digital signature use cases. Propagated (but still direct) objectives comprise non-repudiation for the signed information or documents, the advanced signature format data, or the authentication data.

Indirect security goals (i.e., security chains) are derived as a result of the security relationships, for integrity, confidentiality, authentication or authorization on additional components that relates to the initial (direct) targets, such as additional security objectives for the Secure Cryptographic Device (SCD) on Fig. 9.

Thus, the complete set of security objectives is as follow:

- **Main direct objective** is to keep the non-repudiation of the SNCD (i.e., non-repudiation of its digital signatures and related operations).
- **Propagated direct objectives** are to keep the non-repudiation of signed information or documents into the SNCD, during the following digital signature operations
  - signing a document
  - verifying a document
  - changing a signature format
  - authenticating a user
- **Derived indirect objectives** are obtained from interaction and representation relationships of the information being processed with a digital signature operation. Table 6 shows relevant representation relationships for some system components. Each

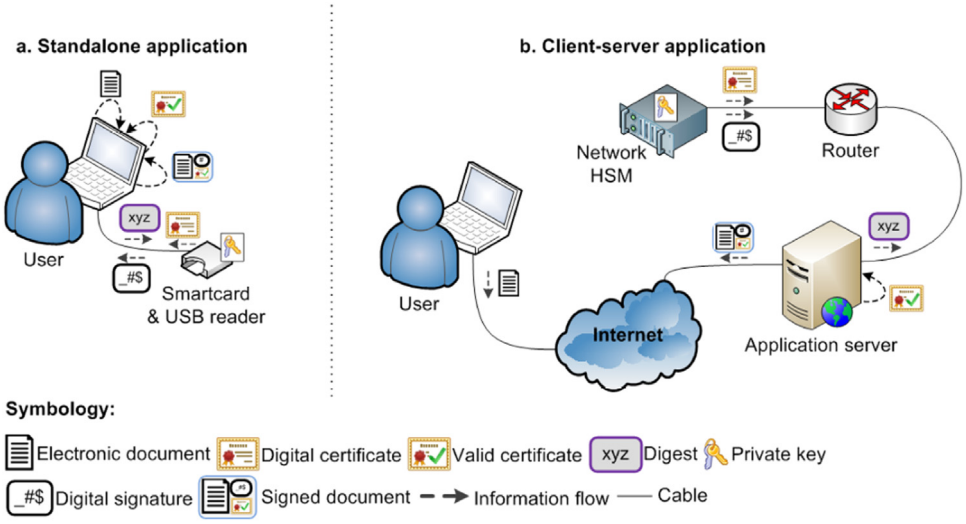


Fig. 8. Digital signature - physical architectures. Source: (Mora, 2017), English translated.

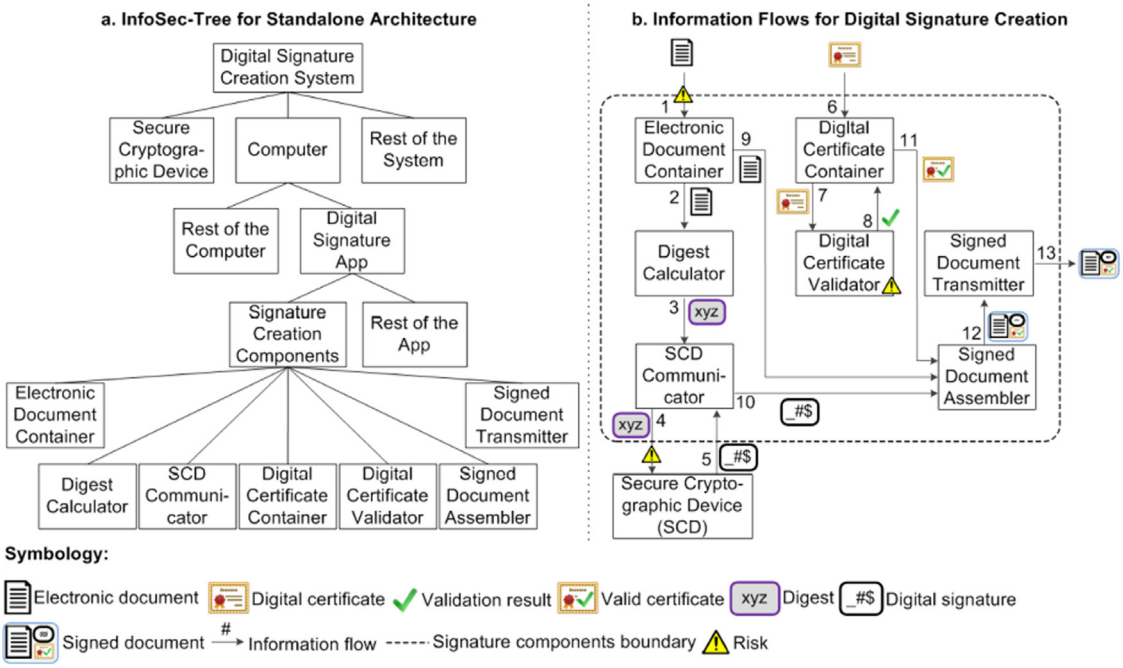


Fig. 9. Digital signature - system representation. Source: (Mora, 2017), English translated.

Table 6
Digital signature - representation relationships.

#	Component 1	Component 2	Representation
1.	original document	copy of document	document content
2.	original document	document hash	document content
3.	original document	encrypted document hash	document content
4.	authentication data	authentication data hash	data used for authentication
5.	user private key	user digital certificate	user identification

and every interaction and representation relationship should be considered to identify additional objectives for supporting the achievement of the initial non-repudiation goal. The iterative indirect objective identification process can be summarized, for brevity of space, as follow:

- to keep confidentiality and integrity, as appropriate, for information and software applications involved into the digital signature operations
- to achieve valid authentication and authorization for the participating subjects or its representations

The software application for handling digital signatures can be very complex, but the security relationships (i.e., interactions and representations, for this case) provide a simple iterative way for identifying the additional (indirect) security objectives. The additional goals follow some patterns along the application, to support

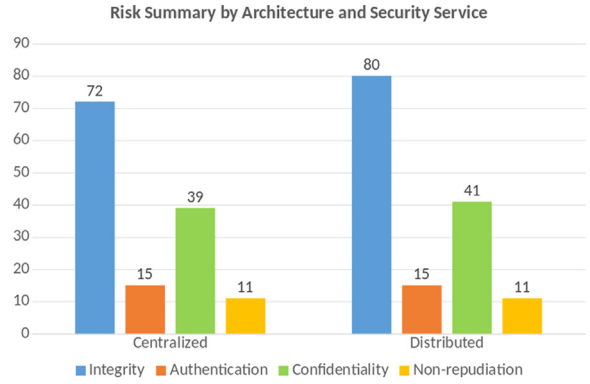


Fig. 10. Digital signature - risk summary. Source: (Mora, 2017), English translated.

the initial established non-repudiation objective, as described following.

Indirect objectives for integrity and confidentiality are iteratively defined for information components representing other information components, such as the digitally signed representation of a document that requires integrity for its non digitally signed version, or a hash string representing a summary of the entire document. A similar process can be followed for identifying additional integrity and confidentiality goals for software functions, starting with the components where a non-repudiation objective has been defined, and following sequences (chains) of interactions between software functions (i.e., functions calling another functions, local or remotely) described on the interaction diagrams.

Indirect authentication and authorization goals can be identified by considering representation relationships for the subject id, such as its digital certificate, and the interactions with the software functions requiring some authentication or authorization context.

8.1.3. Risk identification and assessment

Risk identification takes the security objectives defined for the system components as its input. The next step is the identification of existing vulnerabilities and threats to establish the corresponding risks, by iterating over all the security relationships. The risks were described with structured template sentences, containing all the expected elements for a risk. Seventeen risk templates were defined. Following there are two examples, were text in square brackets *[]* represent variable values:

- **Integrity risk template**: *[A vulnerability]* allows *[a threat source]* to modify *[an information component]* while transmitted from *[component A]* to *[component B]*.
- **Confidentiality risk template**: *[A vulnerability]* allows *[a threat source]* to reveal *[an information component]* while stored into *[a component]*.

Centralized and distributed architectures were also considered because of the differences on the risk evaluation. The result of iterating for all components having security objectives is a table with 228 risks, organized into 82 risks for the centralized architecture, 90 risks for the distributed architecture, and 56 architecture independent risks. Fig. 10 shows the risk summary, for both architectures and categorized by security service.

For risk assessment two factors were considered, the probability of occurrence, and the impact of the risk. For establishing the risk probability there were two considerations, namely aspects related to the vulnerability and aspects related to the threat, each of them with values in the range of [0–9]. For establishing the risk impact there were technical aspects and business aspects, also with values in the range of [0–9]. Tables 7 and 8 describe the specific aspects considered for probability and impact, respectively.

Table 7
Digital signature - aspects for risk probability.

Aspect category	Aspect
Vulnerability	Easy of Discovery, Easy of Exploitation
Threat	Threat Source Ability, Reward, Necessary Resources

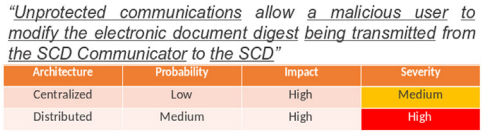


Fig. 11. Digital signature - risk example. Source: (Mora, 2017), English translated.

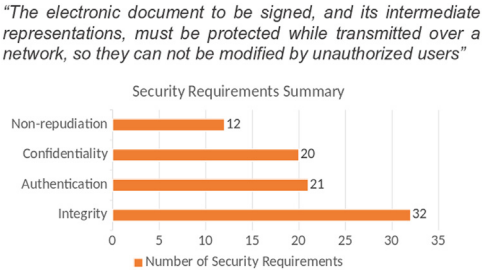


Fig. 12. Digital signature - security requirements. Source: (Mora, 2017), English translated.

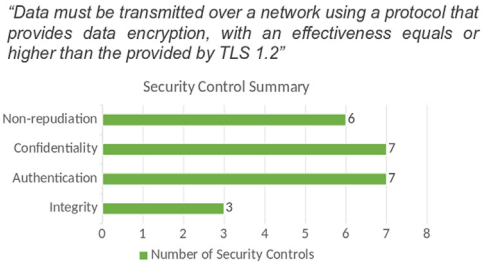


Fig. 13. Digital signature - security controls. Source: (Mora, 2017), English translated.

All aspects were weighted evenly (i.e., averaged) for calculating the probability and the impact. Other calculation methods were tested, such as assigning different weights to some aspects, but there were no significant differences on the results so the final way used for impact and probability was the average of the considered aspects. Similarly, the final severity of the risk was calculated by averaging its probability and its impact. The severity range of [0–9] was also evenly divided into five sub-ranges with qualitative values, namely very low, low, medium, high, and very high. Risks with final severity value of medium, high and very high were selected for mitigation with a security control, risks with severity very low and low were only appropriately reported and assumed. On the final results, 152 risks were selected for mitigation, the remaining 76 risk were selected to be assumed. Fig. 11 shows an example of a risk with its evaluation decision.

The last step of the security process is to group similar risks to define security requirements or policies to be supported with security controls. The selected 152 risks were grouped into 85 security requirements, and the requirements supported with 23 security controls. Fig. 12 shows an example of a security requirement and the summary of them. In the same way, Fig. 13 shows an example of a security control and a summary of the remaining controls to be implemented.

As a final comment, it can be seen how a single security objective (non-repudiation of the SNCD) allowed to derive security

Table 8
Digital signature - aspects for risk impact.

Aspect category	Aspect
Technical Factors	Interruption of a Security Service
Business Factors	Interruption of Business Activity, Economic Loss, Loss of Reputation

objectives, risks, security requirements and controls for additional system components, for producing the integral and comprehensive security solution.

8.2. University academic cloud

The framework developed during this research was initially created for defining the security of the Computational Academic Cloud (NAC by its acronym in Spanish, Nube Académica Computacional) at the University of Costa Rica (UCR).

The UCR is the largest public institution dedicated to higher education in Costa Rica. It was founded on 1940, and it has about 12 campus nation-wide. By the year 2021, it has about 46 schools and 48 research units, and also there are more than 40,000 active students with 383 undergrad and 260 grad programs.

The NAC started as a research project on 2013, led by the researcher/professor Ph.D. Ricardo Villalón with the expectation of providing IT support for the main institution activities. On the year 2017, some initial services were deployed in production for supporting several courses and other academic activities, such as supporting the learning management system. On the year 2020, the consequences of world-wide covid-19 pandemic launched the NAC exponentially, for supporting remote-work on academic and research activities. The NAC is (at the time of writing this manuscript) the largest academic cloud platform nation-wide.

Some initial concerns about architecting the security of the NAC were:

- How to deal with security complexity of the physical infrastructure, but also with the virtual infrastructure (i.e. virtual machines), and the information.
- Is there any difference between the security process of physical and virtual infrastructure.
- What is the difference about implementing security for information and security for software.
- About complexity, is there any way to divide the whole cloud platform into smaller well-defined components that can be secured independent and incrementally.
- Is there any way to precisely describe the security relationship between people, physical components, software, information, and any other type of system components.

These and other concerns gave rise to the establishment of the whole-parts tree, as a way for recursively dividing a large system into smaller ones, but at the same time supporting a systemic, comprehensive and organized process for managing the security process. On the same line, interaction diagrams arise as a way for representing security from a functional perspective, and at different granularity levels. Preliminary results of this work are published at Villalón-Fonseca et al. (2014a).

The whole-parts tree for the NAC is divided into several initial subtrees, namely:

- **Hardware devices** comprising servers, storage and networking, described recursively into smaller components according to its structure.
- **Software components** is a very large sub-tree, initially having the hypervisors of the physical servers, but also about 50 cloud management services, each composed of clusters of virtual machines (VM). Each cluster was described by its VMs, and so on, up to the granularity of software applications or application sub-components, as required. In-house developed applications usually had a more granular component description.
- **Information components** is also a very large sub-tree with significant differences regarding physical devices and software, because information components do not interact with each other, but they interact with software.

Table 9
Digital signature application - security representation.

Element	Value
Business Direct Goals	3
Direct Objectives	19
Indirect Objectives	25
Security Relationships	136

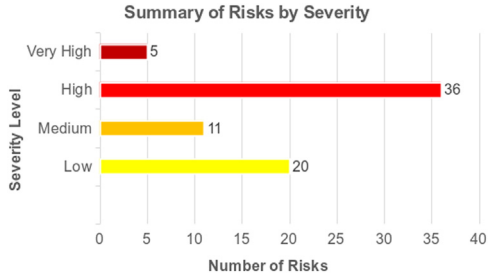


Fig. 14. Digital signature application - security risks by severity. Source: (Gonzlez, 2021).

Even though the NAC is a very dynamic platform, with many variations on its structure or functionality, in the long-term the security framework has been good enough to support that dynamism, with a modular management of security that works at any granularity level. Due to changes on institutional goals, the platform had a re-engineering phase of the management and production services by the beginning of 2019, and the security process supported that big change, because of the systemic and organized approach.

8.3. Digital signature software applications

This section summarizes a more recent real example that is complementary to the project of securing non-repudation on the SNCD. The goal was to develop a security guide for a software application having digital signature functionality running on desktop computers and mobile devices over the Internet. The approach is a multi-dimensional about cybersecurity, by initially considering several security services, namely integrity, confidentiality and authentication to be applied to the system components. Non-repudiation issues were already managed on the previous project.

By applying the security process proposed by the framework, from three initial security goals, the number of elements established for the security representation is shown on Table 9.

The analysis produced the risk distribution shown on Fig. 14. All risks above medium severity were selected for mitigation, that way producing the security requirements and controls shown on Table 10.

Table 10
Digital signature application - security requirements and controls. Source: (Gonzlez, 2021).

Element	Value
Security Services	3
Security Requirements	45
Security Controls	25

9. Featured framework applications

This section describes additional relevant IT platforms and applications, in the context of using the proposed framework for cybersecurity.

9.1. Software applications

Software applications are essential on any IT platform, whether large or small, complex or simple. Modern software applications are created with many programming paradigms, languages or software frameworks.

This diversity of applications can be represented with a whole-parts pattern, and with interaction diagrams. The components of the system are the data and functional structures of the software, whether objects, data structures, modules, functions, or other. Function calls and data handling represent interactions between components.

Representation relationships between the source code of a software application, and the corresponding binary code allow to systematically establish security controls for source code, by defining security objectives for the binary representation, and vice versa.

It is possible to automate the construction of whole-parts tree and interaction diagrams, by building a tool inside the integrated development environments (IDE), for parsing the source code to create a system representation.

9.2. Data center and large infrastructures

Modern data centers and the infrastructure of large organizations are ideal examples for applying the framework. The scalable design of the whole-parts tree and the interaction diagrams allows to divide a large security process, while keeping it integrated.

The framework works appropriately in centralized infrastructures in a single physical location, but it also works for distributed data centers around the world. The whole-parts tree is agnostic to the components physical location, and it considers the physical security aspects through the isolation and interaction relationships.

Communication channels between components give rise to interaction diagrams, independently of complexity of the communication. Interaction diagrams are also agnostic to the location or distance between components.

Systemic and automated versions of the whole-parts tree and interaction diagrams could be constructed with support of existing network protocols. For example, Simple Network Management Protocol (SNMP), or other related protocols may support system representation of large infrastructures, that would be hard to achieve with manual procedures.

9.3. Cloud and virtual infrastructures

Cloud technologies and virtual infrastructures produce a major change in security implementation, related to traditional data centers.

The security relationships defined for physical resources are different in the corresponding virtual resources. For example, physical hardware resources are more related to isolation and interaction relationships, but virtual hardware resources are more related to interaction and representation relationships, like software.

The framework allows to appropriately identify the differences in the nature of the components, and consequently in the construction of the security for virtual and cloud environments.

9.4. Mobile infrastructures

Mobile infrastructures have a high level of interaction within the real world. Information stored in mobile devices presents important security challenges, due to the strong relationship with the state of real objects and people.

Representation relationships are the glue to achieve an integral security between the real and the digital world, including the issues associated with people's privacy.

That way, security relationships are key elements in security solutions with mobile technologies.

9.5. Privacy protection

Privacy of people is a complex problem worldwide, because of the challenge generated with modern technologies.

The framework provides representation relationships to comprehensively model the security of information and other resources related to people, in the context of an IT system.

Sensitive applications, which represent and interact with people, can develop a security system with all the required level of detail, by using the framework structures and the security relationships.

9.6. Cryptographic systems

Cryptography has proven to be an essential tool in the security of IT systems. From the perspective of the framework, cryptographic algorithms generate representations of information or other digital resources, that are immune to security threats.

The encrypted versions of the data have representation relationships with their original (and probably vulnerable) versions. For example, the security of a cryptocurrency system could be treated through representation relationships between the real-world currency and the corresponding secure encrypted representations.

9.7. Smart cities and internet of things

Smart cities and the Internet of Things (IoT) are and will be permanent topics of technological work during the 21st century. IoT systems have many heterogeneous components, specialized environments, scalability requirements and consequently the security is complexity. All these aspects are directly supported by the framework.

The framework is also prepared to evolve in several different ways to support emerging security requirements.

10. Conclusions and future work

This research contributes to the academy with a security framework based on concepts of the nature of security, namely isolation, interaction and representation relationships identified for the components into a system. The framework is supported with a strong architecture description, having three viewpoints to represent the system, the security goals, and the security process. The main contributions of the architecture approach are a robust system representation, that is appropriate to develop a security solution, but also a well-defined objective-based security representation.

The scope of the framework is the security for IT systems and cybersecurity, including information, software, virtual and physical

resources, IT devices, money, people, and a variety of physical objects and their corresponding representation in the digital world. Real life examples are presented, but also some featured applications where the framework might be used to improve the security.

There is also an important practical contribution. The models for the system, the security and the process provide a systemic, comprehensive and with-automation-capabilities way for identifying security risks and for defining security controls. The architecture approach supports the main aspects of the system security life-cycle, including conceptualization, elaboration, and management, which allow to generate evidence and documentation of the security solution, but also to technically support the implementation of the security controls. The proposed models are scalable to represent a wide range of systems, from devices, mobile applications or software systems to cloud infrastructures, distributed data centers, smart cities, Internet of things solutions, and more.

Additionally, the security relationships established by the nature of security provide precise locations into the system to look for expected or even unexpected risks, related to the security objectives. The risk identification process is iterative and well defined by the security relationships, which allows to focus on the security process itself, but it also simplifies the required intellectual effort to identify all system places to look for risks, or on interactions with external entities.

Regarding the integration of the framework with existing security solutions, the modularity of the system representation and the security relationships enable a simple and intuitive integration with other existing academic or commercial security solutions. The security objectives consider the space (i.e., the system) and time dimensions, but also the security services, for aligning with most existing and well defined security solutions, such as international or even proprietary security standards. Furthermore, the security goals behave as a guide for coupling the security with existing solutions, and the solutions could be improved with the identification of additional indirect security objectives, when constructing the security chains proposed in the framework.

As a concluding remark about scope and limits, the framework and the security architecture focus on solving the complexity of the security for large systems, by clearly and specifically describing the places where to look for the security. On the other hand, there are aspects not currently being approached in detail by the framework, but considered appropriately. The security process does not describe detailed steps for collecting policies and goals at the organization level. There are no specialized or technical considerations for identifying vulnerabilities or threats at the places indicated by the security relationships. The security process does not propose a methodology for risk assessment either. All these aspects and related issues can be solved by taking techniques and solutions already available in the industry and the academy.

10.1. Future work

The framework described on this research can be improved or extended in several different ways.

The security chains derived from the security relationships can be improved by establishing security patterns, by relating the security services of the components in a chain. For example, if a targeted information component has a confidentiality requirement, additional indirect objectives can lead to confidentiality and integrity of the software application managing the information, but also similar objectives for the operating system administering the computer resources.

The presented framework supports a systemic handling of the security process with automation capabilities. This could be achieved by developing a software application for implementing the entire security process, and actually for managing the entire security life cycle, including system conceptualization, elaboration, management, and more.

A current pending job of the security industry is the assurance (i.e., evaluation) of security solutions for complex systems. There are international certifications processes for assuring the security of IT products, such as the Common Criteria Certification (Common Criteria, 2020), but the assurance step is still a work to be done for complex IT systems, or even for system of systems (SoS). The models and methodology described with this framework may be used to support scalability, and consequently for security and assurance of the security for complex systems or SoS.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ricardo Villalón-Fonseca: Investigation, Writing – review & editing.

Acknowledgments

Thanks to the Centro de Investigaciones en Tecnologas de la Información y Comunicación (CITIC) and the Escuela de Ciencias de la Computación e Informtica (ECCI) at the Universidad de Costa Rica (UCR), for supporting this research under project number 834-B9-095. The most sincere thanks to my colleague and friend Miguel Carballo Chavarría and also for the staff at the División de Servicios Tecnológicos of the Banco Central de Costa Rica, who provided their support and comments for validating the framework with real life scenarios. Last but not least, thanks to Alejandro Mora for providing the material and figures of his master graduate project for supporting the digital signature real-life example.

References

Accenture Security. The cost of cybercrime. 2019. https://www.accenture.com/_acnmedia/pdf-96/accenture-2019-cost-of-cybercrime-study-final.pdf.

Agrafiotis, I., Nurse, J., Goldsmith, M., Creese, S., Upton, D., 2018. A taxonomy of cyber-harms: defining the impacts of cyber-attacks and understanding how they propagate. J. Cybersecur. 1–15. doi:10.1093/cybsec/tyy006.

AON. Cyber security risk report. 2019. https://www.aon.com/getmedia/4c27b255-c1d0-412f-b861-34c5cc14e604/Aon_2019-Cyber-Security-Risk-Report.aspx.

Bishop, M., 2004. Introduction to Computer Security. Addison-Wesley Professional.

Bishop, M., 2018. Computer Security Art and Science, second ed. Pearson.

Bradley T. The standard cybersecurity model is fundamentally broken. 2019. https://www.forbes.com/sites/tonybradley/2019/10/07/the-standard-cybersecurity-model-is-fundamentally-broken/6f9c26711894.

Calosi, C., Graziani, P., 2014. Mereology and the Sciences. Springer, p. 371. doi:10.1007/978-3-319-05356-1.

Cherdantseva, Y., Hilton, J., 2013. A reference model of information assurance & security. In: IEEE Proceedings of ARES.

Craigen, D., Diakun-Thibault, N., Randy, P., 2014. Cybersecurity. Technol. Innov. Manag. Rev. 4 (10), 13–21.

Common Criteria. 2020. https://www.commoncriteriaportal.org/ccra/index.cfm.

Dark Readin, 2016. 412 Million Users Exposed in Adult Friend Finder. Penthouse Breach. https://www.darkreading.com/attacks-breaches/412-million-users-exposed-in-adult-friend-finder-penthouse-breach-/d/d-id/1327478.

ECA, 2019. Challenges to Effective EU Cybersecurity Policy. European Court of Auditors. https://www.eca.europa.eu/Lists/ECADocuments/BRP_CYBERSECURITY/BRP_CYBERSECURITY_EN.pdf.

FBI. The morris worm: 30 years since first major attack on the internet. 2018. https://www.fbi.gov/news/stories/morris-worm-30-years-since-first-major-attack-on-internet-110218.

Fischer, E., 2016. Cybersecurity Issues and Challenges: In Brief. Congressional Research Service.

Goman, M., 2018. Towards unambiguous IT risk definition. In: Central European Cybersecurity Conference, p. 4.

Gonzlez, A., 2021. Creación de una gu-a de Aseguramiento de la Información Para Aplicaciones de Software en el Sistema Nacional de Certificación Digital en Internet. Universidad de Costa Rica, San José, Costa Rica.

ISO. ISO/IEC 42010 systems and software engineering - architecture description. 2011.
ISO. ISO/IEC 27001 information technology security techniques information security management systems. Requirements2012.
ISO. ISO/IEC 2382:2015 information technology - vocabulary. 2015.
ISO, 2018a. ISO/IEC 27000 information technology. "Security techniques " Information security management systems Overview and vocabulary.
ISO, 2018b. ISO/IEC 27005 information technology. " Security techniques " Information security risk management.
ISO. ISO/IEC 42020 systems and software engineering - architecture processes. 2019.
ITU. Data Communication Networks: X.800 Open Systems interconnection (OSI); Security, Structure and Applications. Security Architecture for Open System Interconnection for CCITT Applications1991.
ITU Publications. Global cybersecurity index (GCI). 2018. https://www.itu.int/dms_pub/itu-d/opb/str/D-STR-GCI.01-2018-PDF-E.pdf.
Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G., 1992. Object-Oriented Software Engineering A Use Case Drive Approach. Addison-Wesley.
Josang, A., 2017. A consistent definition of authorization. 13th International Workshop on Security and Trust Man-agement (STM 2017). Oslo.
Maconachy W., Schou C., Ragsdale D., Welch D. Proceedings of the 2001 IEEE Workshop On Information Assurance and Security United States Military academy. 2001.
Maurer, T., Morgus, R., 2014. Compilation of Existing Cybersecurity and Information Security Related Definitions. New America, Washington.
McCumber, J., 1991. Information systems security: A comprehensive model. The Pentagon.
Mora, A., 2017. Definición de un Proceso de Aseguramiento de la Información Para los Componentes Tecnológicos, Que Utilizan Certificados y Firma Digital en una Aplicación de Software Dentro del Sistema Nacional de Certificación Digital. Universidad de Costa Rica, San José, Costa Rica.
Muprhy, C., 2018. Why cybersecurity fails. The Digital Transformation People. https://www.thedigitaltransformationpeople.com/channels/cyber-security/why-cybersecurity-fails/.
National Institute of Standards and Technology. Framework for improving critical infrastructure cybersecurity. Version 112020;. doi:10.6028/NIST.CSWP.04162018.
Obermaier F., Obermayer B., Wormer V., Jaschensky W. About the Panama papers. 2019. https://panamapapers.sueddeutsche.de/articles/56febff0a1bb8d3c3495adf4/.
Oxford University Press. Oxford Distionary of English. Print Publication Date: 2010. Print ISBN-13: 9780199571123. Published online: 2010. Current Online Version: 2015; 2010.
Pan J., Yang Z. Cybersecurity challenges and opportunities in the new "EdgeComputing + IoT" world. SDN/NFV-enabled Security Mechanism2018;. doi:10.1145/3180465.3180470
Plachkinova, M., Maurer, C., 2018. Teaching casesecurity breach at target. J. Inf. Syst. Educ. 29 (1), 11–20.
Schatz, D., Bashroush, R., Wall, J., 2017. Towards a more representative definition of cyber security. J. Digit. Forensics, Secur. Law (12) 2–8. doi:10.15394/jdfsl.2017.1476.
Stallings, W., Brown, L., 2012. Computer Security Principles and Practive, second ed. Pearson.
Stout, D., 2000. Youth Sentenced in Government Hacking Case. The New York Times. https://www.nytimes.com/2000/09/23/us/youth-sentenced-in-government-hacking-case.html.
The Council of Economic Advisers. The cost of malicious cyber activity to the U.S. economy. 2018. https://www.whitehouse.gov/wp-content/uploads/2018/03/The-Cost-of-Malicious-Cyber-Activity-to-the-U.S.-Economy.pdf.
Townsend, K., 2019. Failures in Cybersecurity Fundamentals Still Primary Cause of Compromise: Report. Security Week. https://www.securityweek.com/failures-cybersecurity-fundamentals-still-primary-cause-compromise-report.
Urciuoli, L., Mannisto, T., Hintsa, J., Khan, T., 2013. Information & security:. Int. J. 29 (1), 51–68. doi:10.11610/isij.2904.
Villalón-Fonseca, R., Mora-Castro, A., Bartels-Gonzlez, R., Carballo-Chavarra, M., Marn-Raventós, G., 2016. Promoting quality e-government solutions by applying a comprehensive information assurance model: use cases for digital signature. In: Mata, F., Ponts, A. (Eds.), ICT for Promoting Human Development and Protecting the Environment. WITFOR 2016. IFIP Advances in Information and Communication Technology, vol 481. Springer.
Villalón-Fonseca, R., Solano-Rojas, B., Marn-Raventós, G., 2014a. An applied methodology for information security and assurance: a study case for cloud computing. In: Proceedings on ICITST.
Villalón-Fonseca, R., Solano-Rojas, B., Marn-Raventós, G., 2014b. Infosec-tree model: an applied, in-depth, and structured information security model for computer and network systems. J. Internet Technol. Secur. Trans. 3 (3), 300–310.

Ricardo Villalón-Fonseca received his Master degree from the University of Costa Rica, and the Ph.D. degree from the University of New Mexico, United States. His research interests include information security, cybersecurity, cloud computing, digital signature, and evolutionary computation. He is the creator and technical leader of the academic cloud at the University of Costa Rica. He is a member of the National Standardization Body of Information Technology (CTN 27) of Costa Rica (INTECO). He actively contributes to the International Standardization Organization (ISO) as a member of several working groups, such as ISO/JTC1/SC27/WG3 or ISO/JTC1/WG13, among others.