Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

A SYSTEMATIC LITERATURE REVIEW ON THE APPLICABILITY OF SECURITY PATTERNS

TRABAJO FIN DE MÁSTER

Beatriz García González

2018

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos

TRABAJO FIN DE MÁSTER

A SYSTEMATIC LITERATURE REVIEW ON THE APPLICABILITY OF SECURITY PATTERNS

Autor

Beatriz García González

Director

Julio César Caiza Ñacato

Ponente

José María del Álamo Ramiro

Departamento de Ingeniería de Sistemas Telemáticos

2018

Acronyms

ATL - ATLAS Transformation Language
CBSE - Component-Based Software Engineering
CIA - Confidentiality, Integrity, Availability
DDS - Design Document Specification
EuroPLoP - European conference on Patterns Languages of Programs
GDPR - General Data Protection Regulation
GoF - Gang of Four
GPS - Global Positioning System
ID - IDentifier
IDE - Integrated Development Environment
I/E criteria - Inclusion/Exclusion criteria
IEEE - Institute of Electrical and Electronics Engineers
IS - Information System
MDS - Model-Driven Security
OCL - Object Constraint Language
PbD - Privacy by Design
PLoP - Conference on Pattern Languages of Programs
RAM - Reusable Aspect Models
RQ - Research Question
SA - Security Attributes
SCRIP - SeCurity patteRn Integration Process
SCRI-PRO - SeCurity patteRn Integration aPPROach
SCRIsTUDIO - SeCurity patteRn Integration sTUDIOS
SDLC - Software Development Life Cycle
SEMDM - Software Engineering: Metamodel for Development methodologies
SI - Sistema de Información
SLR - Systematic Literature Review
SoSPa - System of Security design Patterns
SPAR - Security Pattern Application Rule
SR - Security Requirements
SRS - Software Requirement Specification
STRIDE - Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service and Elevation of privileges
UML - Unified Modeling Language
URL - Uniform Resource Locator
TFM - Trabajo Fin de Máster

Resumen

Hoy en día existe la necesidad de desarrollar sistemas de información (SI) amigables con la seguridad y la privacidad. Un sistema con un diseño seguro puede ayudar a prevenir ataques que vulneren la integridad y alteren el correcto funcionamiento del mismo, como ocurrió en mayo del 2017 con el ataque *WannaCry*. Por su parte, un sistema que proporcione privacidad puede garantizar que sus usuarios estén informados y que puedan controlar el tratamiento que recibirán sus datos personales. Esto incluye información sobre qué datos se recogen por cada aplicación, quién es el responsable de ello y cómo se procesarán. El escándalo de *Cambridge Analytica* el pasado marzo, es un ejemplo de lo que ocurre cuando hay violaciones de privacidad. Por eso, el diseñar sistemas amigables con la privacidad ayudaría en gran medida a garantizar la privacidad de los usuarios.

Los patrones de diseño han sido considerados una buena herramienta para diseñar sistemas amigables con la seguridad y la privacidad. Sin embargo, aunque se ha avanzado en el mundo relativo a los patrones de privacidad, aún hacen falta más resultados empíricos que garanticen la aplicabilidad de los patrones de privacidad y sus beneficios. Teniendo en cuenta la cercanía del mundo de la seguridad, y para evitar "*reinventar la rueda*", el objetivo de este estudio es llegar a conocer cuáles son los mecanismos que han sido empleados en el dominio de seguridad para facilitar la aplicabilidad de los mismos.

Considerando que no se han realizado estudios sistemáticos formales para analizar la aplicabilidad de patrones de seguridad, y dado que el realizar un estudio de este estilo daría mayor solidez al querer exportar conocimiento entre dominios; en este Trabajo Fin de Máster (TFM) se desarrolla una revisión sistemática de literatura (SLR, del inglés *Systematic Literature Review*) para conocer los mecanismos utilizados para aplicar patrones de seguridad durante la fase de diseño de sistemas de información. El proceso desarrollado ha seguido las buenas prácticas habituales recomendadas para el desarrollo de este tipo de investigación secundaria.

La estrategia de búsqueda que se ha empleado para el desarrollo de la SLR ha sido el uso de bases de datos científicas. En concreto, se ha escogido Scopus, pues es la base de datos científica más grande con artículos revisados por pares. Una vez elaborada la cadena de búsqueda del estudio, esta se introdujo en Scopus el 12 de marzo del 2018, obteniendo un total de 160 resultados. Estos 160 resultados fueron filtrados aplicando unos criterios de inclusión y exclusión, los cuales se dividieron en dos: criterios de inclusión/exclusión automáticos, que pueden realizarse de manera automática a través de las herramientas de Scopus; y los manuales, los cuales necesitan del investigador para ser llevados a cabo. Una vez se aplicaron los criterios de inclusión/exclusión automáticos, como por ejemplo que todos los resultados fuesen artículos revisados por pares y que estuviesen documentados en inglés, se obtuvieron 159 documentos que satisfacían ambos criterios. Posteriormente, dichos 159 documentos, fueron sometidos a los criterios de inclusión/exclusión manuales, en los cuales se comprobaba, entre

otras cosas, que el documento reporte un mecanismo para mejorar la aplicabilidad de patrones de seguridad al diseñar sistemas de información. Una vez aplicados los criterios manuales de inclusión/exclusión, 18 documentos fueron seleccionados para seguir con el estudio. Sobre estos 18 documentos, se aplicaron también criterios de calidad, cuyo objetivo era determinar la calidad del contenido de dichos documentos. Al final de esta fase, se seleccionaron 17 documentos, sobre los cuales se trabajó en las fases posteriores.

La extracción de datos de los 17 documentos finalmente seleccionados, se realizó en dos iteraciones. En la primera iteración, los datos a extraer se definieron en base a las preguntas de investigación (RQ, del inglés *Research Question*). Como resultado de la primera iteración, se observó que los trabajos que investigaban la aplicabilidad de los patrones de seguridad la abordan desde tres facetas distintas: selección, estructura y holística. Por ello, para la segunda iteración se tuvieron en cuenta los resultados del análisis de la primera y se establecieron los criterios de extracción por cada una de las facetas definidas. Los datos extraídos incluían información básica de los documentos (autores, fecha de publicación, etc.), los mecanismos reportados y mapeados como elementos de metodologías, los impactos de los resultados al usar dichos mecanismos y las descripciones de los diferentes procesos, entre otros.

Una vez todos los datos fueron extraídos de manera correcta, se realizó un análisis comparativo por cada una de las facetas, que ayudó a responder a las preguntas de investigación definidas al principio del estudio.

Finalmente, como resultado de este estudio, se establecieron las conclusiones pertinentes, donde las más relevantes fueron que la aplicabilidad de patrones de seguridad es vista por los distintos autores de diferentes formas, las cuales se han designado como facetas. Además, dependiendo de cada faceta, los mecanismos encontrados son de diferente tipo y tienen niveles de madurez distintos.

Los mecanismos más maduros han sido encontrados en la faceta de selección, y por ello serían susceptibles a ser replicados en el dominio de la privacidad. No hay que olvidar tampoco el resto de mecanismos reportados en las facetas de estructura y holística, pues ambas también reportan detalles importantes, como estar asociados a una característica exclusiva del ámbito de la seguridad. Por ello, se deben considerar igualmente para proponer nuevos mecanismos que ayuden a facilitar la aplicabilidad de patrones de privacidad.

Palabras clave: revisión sistemática de literatura, aplicabilidad, patrones de seguridad, diseño, sistemas de información.

Abstract

Nowadays there is an increase of interest to design security and privacy-friendly information systems (IS). A security-friendly system could help the system avoiding attacks which damage its integrity and correct operation, like *WannaCry* did in May 2017. A privacy-friendly system could guarantee that the users are informed and can control the treatment their personal data will receive, regarding which data is going to be collected, who is the responsible on it, and what procedure is going to take place. *Cambridge Analytica* scandal in March 2018, is an example of privacy violations. Hence, privacy-friendly systems could help to guarantee users' privacy.

Design patterns have been considered a good tool for designing security and privacy friendly systems. Nevertheless, although there have been improvements in the privacy domain related to privacy patterns, there is still a lack of empirical studies to foster the applicability of privacy patterns and their benefits. Considering the closeness of the security domain and avoiding "*reinventing the wheel*", the aim of the research is to get to know which mechanisms have been used in the security domain to facilitate the applicability of security patterns.

Considering the lack of formal systematic studies to analyze the applicability of security patterns, and given that developing such a kind of study would help to export knowledge between domains, this Master Final Project develops a systematic literature review (SLR) in order to know which mechanisms have been used by security experts to apply security patterns while designing information systems. The process has followed the recommended guidelines of good practices for developing this type of secondary study.

The employed search strategy to develop this SLR was the use of Scopus database, as it is the largest peer-reviewed scientific database. Once the search string for the study had been defined, it was introduced into Scopus on May 12th 2018, and 160 results were obtained. These 160 studies were put through a set of inclusion/exclusion criteria, which can be divided in two parts: automatic inclusion/exclusion criteria which can be checked by Scopus' tools; and manual inclusion/exclusion criteria, checked by the researcher. Once automatic inclusion/exclusion criteria were applied, 159 papers were selected. After that, automatic inclusion/criteria were applied to these 159 papers and 18 were selected. Later, quality assessment criteria were applied to the 18 selected papers in order to determine how well documented they were. At the end of this phase, 17 papers were finally selected for the study.

Data extraction was done in two iterations. Within the first iteration, data to extract were defined according to the research questions defined at the beginning of the study. As a result of the first iteration, the applicability concept had changed, as the applicability of security patterns could be divided into three facets: selection, structure and holistic. Hence, for the second iteration, the analysis results for the first one were considered, and data extraction criteria were established for each of the defined facets. Extracted data included basic information about the papers (authors, publication date,

etc.), the reported and mapped mechanisms as methodology elements, the impact of applying those mechanisms and their description, among others.

Once all data were extracted correctly, a comparative analysis for each facet was implemented. This analysis helped to answer the research questions that were defined at the beginning of the study.

Finally, as a result of this study, some conclusions were established. The most relevant ones were that the applicability of security patterns is seen in different ways by the authors. These different ways were named as facets. Further, depending on the facet, the found mechanisms were different and so were their maturity level.

The most mature found mechanisms were those in the selection facet, and they could be replicated in the privacy domain. Nevertheless, mechanisms found within the structure and holistic facet, also reported important details, such as an exclusive security feature associated to security patterns. Hence, these mechanisms should be also considered to propose new mechanisms that ease the applicability of privacy patterns.

Keywords: systematic literature review, SLR, applicability, security patterns, design, information systems.

Contents

List of Figures

List of Tables

1

Introduction

Privacy by Design (PbD) claims that privacy should be taken into account just from the start of the system's design and not be considered as an additional element once this system has been developed. PbD promotes a future vision which ideally requires that the assurance of privacy becomes the organization predetermined *modus operandi* [1]. Hence, privacy would be incorporated into information systems by default, becoming one of the most important priorities of the companies.

Design patterns (further explanation in section 2.1) have been considered as a good approach for designing security [3] [4] [5], and privacy friendly systems [6] [8]. Despite there have been many contributions regarding privacy patterns, as new proposals [6] [11], catalogs [8] [9] [10], relationships [12] and achieving privacy systems proposals [2], this field is yet lacking of empirical evidence regarding their applicability and benefits [22]. Therefore, considering the closeness of the security domain to the privacy one, the aim of this study is to learn how security patterns were applied while designing information systems.

The applicability of security patterns have been study in the past, but no formal systematic studies were developed. Some of the studies that were found while exploring the security domain were considered old enough, as they were developed in 2009, and it is well-known that the world of information systems and technology changes very fast, so their content could be outdated. Thus, it is necessary to develop a systematic formal study to know how researchers have reported the mechanisms about applicability of security patterns and how designers have dealt with the application of them. The study carried out is a systematic literature review, which is deeper explained in section 2.3.

2

Theoretical framework

2.1.- Design patterns

According to Alexander et al. in [15], "a pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million over times without ever doing it the same way twice". The idea of patterns was also applied in software development, as developers have come across problems that have repeatedly appeared. Design patterns, the term coined to refer to the idea of patterns in software development, are a concept from software engineering which describes recurring solutions to common problems in software design. Design patterns were defined by the Gang of Four (GoF) in [16].

Nowadays there are two important communities where people can notice how many researchers are working on patterns in the software world: PLoP (Conference on Pattern Languages of Programs) [17] and EuroPLoP (European conference on Patterns Languages of Programs) [18]. Both communities organize annual conferences on patterns and pattern language, where academics and practitioners participate sharing their knowledge in the field of software development among others.

A design pattern is divided into different sections. Depending on the author who describes the pattern, they consider three sections (context, problem and solution) [19] or five (context, problem, forces, solution and consequences) [20] and even there are more approaches. The commonly different sections that could be considered by the authors are described next:

- **Context:** environment where a problem is taking place.
- **Problem**: describes the problem that appears repeatedly in the given context. It first describes a general problem specification which is then complemented by a set of forces.
- **Force**: a force is any aspect of the problem that needs to be considered when applying a solution.
- **Solution**: shows how to solve the repeating problem. According to Bushmann in [20], in software architecture every pattern solution is divided into two aspects:

    1. Static aspect: it specifies a certain structure and a spatial configuration of the elements in the pattern.
    2. Dynamic aspect: it is related with the run-time behavior.

- **Consequences**: they are related with the impacts of applying a certain solution.

These differences when defining patterns, will be significant when authors try to normalize a unique template where all security patterns will be defined in the same way.

Security patterns are those design patterns for designing security-friendly systems, and they are our main focus of study.

2.2.- Software Development Life Cycle (SDLC)

When it is talked about software, it does not only refer to the computer programs, but also its procedures, documentation and data required for the operation of a computer [13]. There is an engineering discipline called software engineering interested in the aspects related with the production of software. In software engineering, a software development process is a systematic approach which describes a set of activities that lead all the people implied to the development and deployment of software products [43].

In software engineering, Software Development Life Cycle (SDLC) is the process of designing, developing and testing a new software product [43]. The main goal of SDLC is producing high quality software that satisfies or even exceeds the clients' expectations, and what is even more important, in an optimized time and price [43]. The SDLC defines a methodology to improve the quality and the development of the product and it has six phases [43]:

1. **Requirements analysis**: it is the most important phase during the SDLC, and it is about user requirements elicitation done by some members of the developer's team with inputs from the client, sales department, market surveys, and so on. This information is then used for taking in mind an idea of the project and test its feasibility in an assorted environment.

2. **Defining requirements**: its aim is to clearly identify and define product requirements in order to have the client's approval. This is done by a document called Software Requirement Specification (SRS), in which product requirements are defined and how the programmers can develop them during its life cycle.

3. **Design the product**: the SRS is a guide to follow when finding the best architecture that fits the project. Based on this document, more than one design is proposed, and each of them is documented in the Design Document Specification (DDS). This DDS will be later revised and checked by all the stakeholders, and they will choose the one they think better adapts to the project based on some aspects like risk assessment, product robustness, design modularity, etc. However, designers do not have a first good idea, so they develop the design in an iterative way, which means that they can backtrack to correct the errors and enhance the overall project. At the same time, this phase divides itself in several activities. These activities depend on the product, for example, there will be projects that will not need any database, so, designers cast aside the activity related with the database design. However, there are four main activities in the design phase which are the following [14]:

a) Architectural design: the architects of the system have to identify the global system structure, main modules or components with their proper relations, how these components are distributed, among other important aspects.

b) Interface design: clear definition of interfaces among the components of the system, with no ambiguity, in order to facilitate others developers to use these components without knowing how they had been developed.

c) Component design: the operation of each component is designed in this phase. Each component of the system is taken and it is designed how they will operate.

d) Database design: developers have to think about the data system structure and how the data will be represented in a database. Here, the developer has to choose between one of the several databases that nowadays exist (relational database, graph database, document database, etc.).

4. **Developing the product**: the programming code is generated according to the DDS. If the design phase is done in a correct way, then the development of the code can be done without too much hassle.

5. **Testing the product**: the product is tested by a group of people to whom the beta test is provided, so they can report any error of it. Once these errors are reported, developers have to solve them and it will be tested again.

6. **Market deployment and maintenance**: once the product has been tested and it is ready to be deployed, it is launched in the proper market. The deployment of the product is done in a progressive way, in phases. First, the product is launched in a limited segment of the market (as well as deployment, launching is done progressively) and tested once again in the real business world. Then, based on the feedback, the product can be launched as it is to the market or even improved according to the segment users proposals. Once the product is finally in the market, maintenance tasks are done to keep their performance right.

The study will be focused on the design phase, as the applicability of security patterns occurs during this phase. Furthermore, within the design phase, it is thought that the two most relevant activities for the study are architectural and component design.

2.3.- Systematic Literature Review (SLR)

A systematic literature review (SLR) is a way of identifying, evaluating and interpreting all available research results that are relevant to a particular research question or area [21]. Unlike literature reviews, which have little scientific value, systematic reviews synthesize all the existing work in a manner that is fair.

According to Kitchenham in [21], there are three main phases when developing a systematic review, which in turn, are divided in several stages:

1. **Planning the review**: before carrying out the systematic review, it is necessary to know how the study will be developed. This phase is divided in the following stages:

   a) Identification of the need for a review: it is important that researchers identify the reasons why undertaking a systematic review is necessary. Knowing that, the conclusions of the review could be more accurate.

   b) Commissioning a review (optional) [1]: when an organization does not have enough expertise in the field of knowledge, it will commission researchers to perform a systematic review of the topic.

   c) Specifying the research questions: this is the most important part of any systematic review. The research questions drive all the systematic review methodology, so they must be meaningful and important to practitioners.

   d) Developing a review protocol: it specifies the activities that will be developed to carry out the systematic review. The review protocol helps to reduce the possibility of researcher bias. The protocol helps to document what is being planned:

   - The background highlights the motivation for the survey.
   - The research questions that the researcher is intended to answer. Defining the right ones will lead the researcher to obtained more accurate results at the end of the review.
   - The search strategy that will be followed to search primary studies. There are several search strategies that can be followed, such as databases, snowballing searches or consultations with experts in the field. For this study we have only considered Scopus database.
   - The search string that helps finding the most relevant results related with the review. It is defined by breaking down the research questions into individual aspects.
   - The study selection criteria that will help the researcher to determine which papers will be included in, or excluded from the review.
   - The study selection procedure that describes how the selection criteria will be applied.
   - The study quality assessment checklist to evaluate the content of the papers that will be considered for the review.
   - The data extraction strategy which defines how the information required to answer the defined research questions will be obtained. It is helpful to know how relevant information from each paper will obtained.
   - The data analysis is a detailed description of the findings in line with the research questions.
   - The project timetable should be defined.

   e) Evaluation of the review protocol (Optional): the review protocol is a critical element, so researchers must agree with it. There are two possibilities to evaluate the review protocol. If the review is being developed by a group of researchers, all of them must agree on a procedure to evaluate the protocol.

[1] For this study, this activity was not considered.

On the contrary, if there is only one researcher or student, which is the case of this project, she should present the defined protocol to her advisor in order to get feedback. In any of both cases, there must be an agreement at least in the following points:

- Search strings are appropriately derived from the research questions.
- The extracted data will properly address the research questions.
- The data analysis procedure is suitable to answer the research questions.

2. **Conducting the review**: once the planning phase has been designed, the review can start. This phase is divided in the following stages:

   a) Identification of research: the principal aim of a systematic review is to find as many primary studies relating to the research questions as possible. This must be done in a rigorous way. Tasks that should be considered to develop this stage are described below:

   - Define a search strategy: that identifies the search method that will be applied for developing the study. Once the search strategy is chosen, search string must be defined. One thing that can be done to generate the search string is to break down the research questions into individual facets. Then, write down a list of synonyms, abbreviations, and alternative spellings. Sophisticated search strings are built using boolean *ANDs* and *ORs*.
   - Define tools for bibliography management and document retrieval: it is of quite importance to have both in order to manage all the references obtained from the literature search.
   - Define means for documenting the search: due to the process of performing a systematic review it must be transparent and replicable. The review must be documented in an accurate way for readers to be able to assess the thoroughness of the search. The search should be documented as it happens and the changes made should be justified and noted. Those unfiltered results should be saved and retained for possible reanalysis.

   b) Selection of primary studies: selected papers should provide direct evidence about the research questions, according to the selection criteria already defined in the plan.

   c) Study quality assessment: in addition to the inclusion/exclusion criteria, it is considered critical to assess the quality of the papers found, in order to provide still more detailed inclusion/exclusion criteria. To do so, it is necessary to define the quality assessment criteria in order to *measure* the quality of each selected paper.

   d) Data extraction: identifies and extracts relevant information from the selected papers by addressing the research questions. Data extraction include recording ideas, concepts, contributions and findings of each of the selected studies.

   e) Data synthesis: collects the extracted data, its comparison, and its further analysis.

3. **Reporting the review**: it is the final phase of a systematic review and involves writing up the process that has been followed and the results of the review. These stages should be considered during this phase:

    a) Specifying dissemination mechanisms: the ways for reporting the findings could include writing a paper, a report, a master final project, and so on.

    b) Formatting the main report: the ways for describing the structure of the document.

    c) Validating the report (Optional): in order to avoid bias in the results, all the phases could be validated at their end. The validation method must be defined by the researcher.

3

Motivation and aims

Privay by Design (PbD) implies that privacy must be considered since the earlier phases when designing an information system. One approach gaining popularity is to use privacy patterns to achieve it. Nevertheless, the scope of privacy patterns is relatively new, and there is still a lack of empirical solutions that proves their applicability and benefits [22].

Taking into account the closeness of security domain to the privacy one, the aim of this research is to find which mechanisms have been used in the security domain for facilitating the applicability of security patterns in order to replicate them in the privacy domain. The study must have a high level of validation, so the reported mechanisms could be replicated in the privacy domain. Hence, a formal systematic study should be developed. As there is not a systematic literature review in the field of security regarding mechanisms for pattern applicability, this study develops it.

Considering what has been mentioned beforehand, the main aim of this final master project is finding how security patterns' applicability has been achieved while designing information systems. To accomplish this purpose, several goals have been proposed:

1. To obtain general background in the theoretical framework needed for carrying out this study. It included, sorted by relevance, the fields of privacy and data protection, design patterns, software engineering and systematic literature reviews.
2. To obtain knowledge about the development of systematic literature reviews and taking it into practice.
3. To specify an affordable procedure to develop a systematic literature review.
4. To know and use techniques and support tools for researching, for example the references manager Mendeley, and search engines like Scopus.
5. To introduce the student into a researcher field, that is completely new for her.
6. To develop a critical reasoning that allows the student to develop her own analysis and conclusions.
7. To take first steps for a future proposal about an enhancement of privacy patterns' applicability.
8. Finally, to know the different mechanisms which help improving the applicability of security patterns that have been reported within the selected papers.

4

Final project planning

This master final project has been planned to be undertaken in a five month period. This chapter enumerates the different activities and tasks carried out in this study. A more detailed vision of the whole planning is depicted in Figure 4.1.

This study was divided into four main activities. The first one was obtaining general background for carrying out this project and it lasted one month. Within this first activity, the student obtained enough knowledge about software engineering, design patterns, privacy and systematic literature reviews, that allowed her to developed the project.

The second activity was carrying out the SLR and it took the student almost three months. This activity was divided into two different tasks: planning and conducting the review. When the student planned the review, she had to identify the reason why it was necessary to develop a SLR, define the research questions, that drove the entire study, and the review protocol. The definition of the review protocol included the definition of the search strategy, search string, the inclusion/exclusion criteria, that helped to identify which papers were included in or excluded from the study, and finally the data that were extracted from the selected papers. When the student conducted the review, she identified the relevant researches for the study, selected the papers that satisfied the inclusion criteria, assessed the quality of the selected papers and extracted the required data according to the data extraction strategy defined in the planing phase.

It is known that data synthesis and analysis is inside the SLR's conducting phase, but we took it as a new one due to its duration. So, data synthesis was the third main activity and it lasted a month and a half. Within this activity, the student discussed the findings of the review. She analyzed the applicability of security patterns and demonstrated as a detailed description the findings in line with the research questions.

The last activity was documenting the followed process as well as the results. This document was written during the five months.

Taking into account an average work rhythm of 80 monthly hours, the effort for the master final project was of 400 hours.

The replication package with all the data and the undertaken procedure can be accessed at https://github.com/julioccaiza/security_patterns_applicability_slr.

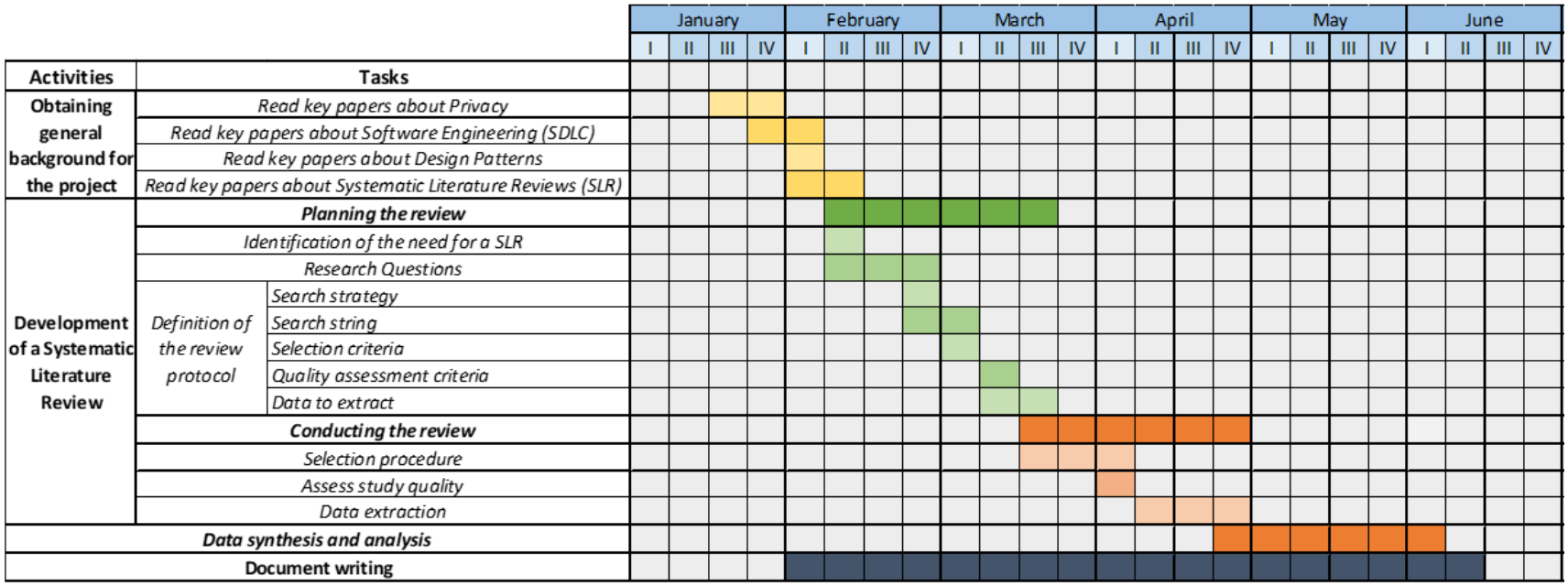

Figure 4.1.- Planning

5

Methodology

This chapter describes the procedure for developing a SLR, that is based on Kitchenham's guidelines [21]. The activities associated to the three main phases of a systematic literature review (planning, conducting and reporting the review), developed in section 2.3, are explained in the remaining part of this chapter.

5.1.- Justification for the SLR

Privacy by Design (PbD) allows building information systems considering privacy from early stages. This has made researchers to think of different approaches to build privacy-friendly systems.

Design patterns are seen as a good tool for designing security and privacy friendly systems. Nevertheless, although there have been improvements in the privacy domain related to privacy patterns, there is still a lack of empirical studies to improve their applicability and complete approaches to engineering privacy into software based on patterns [22].

Considering the closeness of security domain to the privacy one, and avoiding *reinventing the wheel*, the aim of this research is to find which mechanisms have been used in the security domain for facilitating the applicability of security patterns. As there is not a systematic literature review in the field of security regarding mechanisms for pattern applicability, this study develops it.

5.2.- Research Questions (RQs)

Specifying the research questions is the most important part of any SLR [21], so defining the right ones will lead us to obtain more accurate results at the end of the review. The RQs for this study were coined considering some relevant and similar studies [31] and [32].

The main aim of this SLR is to know *how security patterns' applicability has been achieved while designing Information Systems*. This aim leads us to formulate the following RQs:

- ***Research Question 1.*** **Which mechanisms have been used when applying security patterns while designing an Information System?** As the main goal of this research is knowing how to facilitate the applicability of security patterns, there is a need to know which mechanisms have been being used by other authors.

- ***Research Question 1.1.*** **Which of the defined mechanism is the most used?** Once the mechanisms that have been used by engineers have been identified, it would be useful to know if there is a mechanism reported by many researchers and used by many engineers. The most used mechanism could later be translated in a mature and accepted mechanism that can be appropriate to be replicated in other scenarios.
- ***Research Question 1.2.*** **What are the descriptions of the mechanisms?** Knowing mechanisms' description will be valuable to understand and analyze them, even for doing further research such as improvements, replications, etc.

- ***Research Question 2.*** **What is the maturity level of the found mechanisms?** The maturity level could tell us if proposals have achieved a solution level, or if it has been evolved and tested in a laboratory environment, so achieving the *validated* level, or if it has been *evaluated*, so tested in a real environment.
- ***Research Question 3.*** **How have been the results when applying the reported mechanisms?** It is valuable to know how good or bad the results have been after applying the found mechanisms.
- ***Research Question 4.*** **What are the challenges for those identified mechanism?** It is important to know what are the future challenges or improvements for the mechanisms; they can explicitly express the future lines of work.

5.3.- Exploring the domain

The aim of this task is to make a preliminar search to be quickly aware whether or not the subject of this study has been already reported in previous researches. To do so, three different search strings were run into Scopus:

- **First search string**: *TITLE-ABS-KEY("applicability" AND "security patterns")* $\rightarrow$ 11 results. The aim of using this search string was looking exactly for what the study is about. The results here could help to determine whether the context of the project has similar studies in order to analyze them and restructure the initial plan if necessary.
- **Second search string**: *TITLE-ABS-KEY("systematic" AND "literature" AND "review" AND "applicability" AND "security" AND "patterns")* $\rightarrow$ 3 results. The aim of this second search string was knowing whether a systematic literature review about our research topic was previously developed.
- **Third search string**: *TITLE-ABS-KEY("systematic" AND "literature" AND "review" AND "software" AND "security" AND "patterns")* $\rightarrow$ 17 results. Taking into account the previous number of results, the term applicability was removed. The removal of the term applicability was done in order to expand the search domain. Here, it was necessary adding the term software to discard those results related to medicine, physics or chemistry.

Table 5.1 shows the relevant studies obtained from each search string. These papers were selected due to their relevant content. The studies carried out by Ortiz et al. in [25] and Sametinger et al. in [29], are similar to the work that is going to be developed within this thesis, as they are about the applicability of security patterns and their possible problems. However, both of them can be considered old enough (8 and 9 years respectively), as it is well-known that the world of information systems and technology changes very fast, so their content could be outdated.

Maybe the most important fact is that none of the papers found with the three search strings have followed a systematic procedure, so the way of developing the study is not as well guided and defined as in a SLR.

Search String	**Year**	**Title**	**Authors**
First	2010	Applicability of security patterns	Ortiz, R., Vela, B., Moral-García, S., Moral-Rubio, S., Garzás, J., Fernández-Medina, E
	2015	ASE: A comprehensive pattern-driven security methodology for distributed systems	Uzunov, A. V., Fernández, E. B., Falkner, K.
	2016	An Analytical Study of Security Design Patterns	Ponde, P., Shirwaikar, S., Kreiner, C.
Second	2009	A Security Design Pattern Taxonomy Based on Attack Patterns: Findings of a Systematic Literature Review	Sametinger, J., Wiesauer, A
Third	2012	Securing distributed systems using patterns: A survey.	Uzunov, A. V., Fernández, E. B., Falkner, K.
	2015	Software-security patterns	Bunke, M.

Table 5.1.- Relevant studies from the preliminar search

5.4.- Search strategy

There are several search strategies that can be followed, such as catalogs, snowballing or consultations with experts in the field [21].

The strategy followed to develop this study was the use of catalogs, we used Scopus database. Scopus is the largest abstract and citation database of peer-reviewed literature [44], and according to Badia [30], it is better than Web of Science for identifying computer science publications.

Despite the fact that at the beginning of the study it was thought about using at least two search strategies (catalogues searching and snowballing) in order to make a

more complete review, the lack of time during the development of the study, did not let it perform the snowballing technique.

5.5.- Search string

This section is about how the search string was developed. According to Kitchenham in [21], a good practice to develop the search string is breaking down the research questions into individual aspects. Then, writing down a list of synonyms and abbreviations of some key words and finally, construct sophisticated search strings by concatenating the key words with booleans *ANDs* and *ORs*.

5.5.1.- First search string

The search string was divided in four parts. The first one related to the information systems or software systems that will be built when applying security patterns. This first part is composed by the terms *software* and *information system*. Regarding the expression software systems, it is only included the word *software* as it is the most used term and it is understood that it expresses a software-based system. With relation to *information system* term, it cannot be separated because each term by it self becomes too generic and can be applied in several fields of knowledge, for example electric *systems*, people *systems*, etc.

The second part of the search string aims to locate the results in the design phase of the SDLC, so the term *design* was used.

The third component of the search string represents the study's subject, *security patterns*. Here, a dilemma was presented, as the term could be used joint or separated. If the term was separated, the number of results that were obtained was overwhelming (637) and it was complicated to analyze and study all the papers in the fixed period of time. So, it was kept the term *security patterns* together. Further, another synonym (which added relevant papers) appeared in the preliminary search: *security design patterns*. Thus, the third part of the search string consists of the terms *security patterns* and *security design patterns*.

Finally, the last element of the search string is about the applicability of security patterns itself and the type of mechanisms that can be discovered in the studies. To build this part, several synonyms of applicability found in related works were considered, and also some synonyms suggested by the advisor in the study. Terms like applicability, usage, utility, materialization, etc. were used in this part.

Therefore, the first developed search string was the following, and 125 results were given by Scopus:

TITLE-ABS-KEY((“information system” OR “software”) AND “design” AND (“security patterns” OR “security design patterns”) AND (“applicability” OR “utility” OR “adoption” OR “enforcement” OR “operationalization” OR “materialization”))

After validating the previous developed search string, the unfavorable obtained results, made that it was necessary to improve the first search string in order to enhance the results in Scopus.

5.5.2.- Second search string

The decision after validating the first search string was to eliminate the last part of the first search string, that is to say, the one related to the *applicability*.

The number of provided papers by Scopus increased and maybe some results that might not to be related to the applicability of security patterns could also be included. However, those papers that are not related with the applicability of security patterns can be detected while reading their abstracts, and then discarded. The number of results when removing the fourth part of the first search string is consistent (160 results, by March 12th 2018) with the available time for developing the study.

Thus, the final search string that will be employed for the review and with which 160 results are obtained from Scopus is the following:

TITLE-ABS-KEY(("information system" OR "software") AND "design" AND ("security patterns" OR "security design patterns"))

5.5.3.- Validation of the search strings

Once the search string is developed, a good practice is to evaluate the results given by Scopus. To do so, a test-set of six articles (table 5.2) were provided by the advisor of the study.

Using the results of the search string designed in section 5.5.1 and taking the test-set in table 5.2 as a reference to validate this first search string, only the second paper was included in the results provided by Scopus. The reason was that the last part of the search string (the part regarding to applicability) was reducing the scope of the search, as most of the papers in the test-set did not include those words.

The test-set in table 5.2 was used again. This time, the results with the second search string were more favorable than the previous one, as four out of six of the papers in the test-set were included in the Scopus' results. Table 5.3 shows which papers appeared. The fifth paper did not appear because it does not include the term *design* neither in the title nor in the abstract, nor in the keywords. Last paper in the table does not appear in Scopus database.

The last thing it was necessary to verify was that there were no differences when applying the terms *security (design) patterns* in singular or plural. To do so, these terms of the search string were changed into singular, and the number of papers provided by Scopus and the papers themselves were indeed the same.

Year	Title	Authors
2016	Building a security reference architecture for cloud systems	E.B. Fernández, R. Monge, K. Hashizume.
2015	ASE: A comprehensive pattern-driven security methodology for distributed systems	A.V.Uzunov, E.B. Fernández, K. Falkner.
2015	SoSPa: A system of Security design Patterns for systematically engineering secure systems	P.H. Nguyen, K. Yskout, T. Heyman.
2017	Security Patterns and Secure Systems Design	E.B. Fernández.
2009	Enforcing Security in Smart Homes using Security Patterns	P.E. Khoury, P. Busnel, S.Giroux, K.Li, M. Donat, N.H.D.Bohr.
2015	Incorporating Security Features in Service-Oriented Architecture using Security Patterns	A.K. Dwivedi S.K. Rath

Table 5.2.- Test-set for validating the search string

Year	Title	Authors	Appearance
2016	Building a security reference architecture for cloud systems	E.B. Fernández, R. Monge, K. Hashizume.	Yes
2015	ASE: A comprehensive pattern-driven security methodology for distributed systems	A.V.Uzunov, E.B. Fernández, K. Falkner.	Yes
2015	SoSPa: A system of Security design Patterns for systematically engineering secure systems	P.H. Nguyen, K. Yskout, T. Heyman.	Yes
2017	Security Patterns and Secure Systems Design	E.B. Fernández.	Yes
2009	Enforcing Security in Smart Homes using Security Patterns	P.E. Khoury, P. Busnel, S.Giroux, K.Li, M. Donat, N.H.D.Bohr.	No
2015	Incorporating Security Features in Service-Oriented Architecture using Security Patterns	A.K. Dwivedi S.K. Rath	No

Table 5.3.- Success rate for papers test-set

5.6.- Selection criteria

This section explains the criteria used to include or exclude papers in the study. The selection criteria let us identify useful papers for the study and those which were not relevant.

All the result papers from the search string were assessed to see if they have to be included. These papers had to satisfy the inclusion criteria that is exposed in Table 5.4. Automatic inclusion criteria was done inside Scopus, while manual inclusion criteria was applied by the researcher:

Inclusion criteria	**Automatic/Manual critera**
The paper is written in English.	Automatic
The paper is peer-reviewed.	Automatic
The full-text of the paper is available.	Manual
The abstract, title or keywords explicitly mention a mechanism for improving the applicability of security patterns.	Manual
The abstract, title or keywords explicitly state that the article is about security patterns' applicability in the design phase of information systems.	Manual
The paper is not a summary of a workshop/conference.	Manual
The paper is a primary study.	Manual

Table 5.4.- Automatic or manual inclusion criteria

Inclusion criteria from Table 5.4 which required a deeper explanation are following presented:

1. *The paper is peer-reviewed.* Scopus claim that all the paper in its database are peer-reviewed [44]. In academic publishing, the goal of peer review is to assess the quality of articles submitted for publication in a journal. Because of this, peer-reviewed papers can be considered the best research practices in a field.

2. *The abstract, title or keywords explicitly mention a mechanism for improving applicability of security patterns.* The main objective of the review is to know how security patterns can be applied.

3. *The abstract, title or keywords explicitly state that the article is about security patterns' applicability in the design phase of Information Systems.* The study is mainly focused on the design phase within the software development life cycle of Information Systems.

The papers that will be excluded satisfy the exclusion criteria, which are presented down below:

1. *The paper's full-text is not available.* Only those papers with available full-text will be included for further studying, as it is necessary for screening the process and more important, for synthesis stage.

2. *The paper is a summary of a workshop/conference.*

3. *The paper is not a primary study.*

5.7.- Selection procedure

After defining the inclusion/exclusion criteria (I/E criteria) in section 5.6, it was necessary to describe how this criteria will be applied in a selection procedure.

The selection procedure is represented in figure 5.1. The selection procedure was divided into two parts. The first one was an automatic inclusion/exclusion sub-process, that can be done with Scopus tools; and the second one was a manual criteria, which have to be done by the researcher. This procedure was done in two iterations. The first iteration include automatic and manual inclusion/exclusion criteria whilst the second iteration only the manual inclusion/exclusion criteria were performed.

This procedure was developed by two people: the researcher (the student and author of this document) and the advisor. The researcher is who had the charge of reading and marking as included, excluded or unclear each paper. The advisor helped the researcher to classify the unclear papers and also helped her to develop a validation of the selection procedure.

The search string results were marked by the researcher during both iterations as *included* (the paper complies with all inclusion criteria and no exclusion criteria), *excluded* (the paper complies with any exclusion criterion) or *unclear* (the paper is difficult to classify and is left for the next iteration).

5.7.1.- First iteration

The search string developed in section 5.5.3 was typed in Scopus, which provided 160 papers as a result. After this, the first inclusion criterion was applied, and only one out of 160 papers was written in other language but English, so by the end of the automatic criteria, there were 159 papers.

Once the manual criteria were applied, there were 24 papers marked as included, 23 marked as unclear, and the rest (74) marked as excluded.

After the first researcher reading, the unclear papers were sent to the advisor, so he could take also a decision about what to do with those papers, and the researcher read again all the abstracts of those unclear papers, do a further reading and analyze the paper if necessary. According to double checking validation, it was good if the two people (advisor and researcher) had the same result about including or excluding the paper. However, if they did not agree, both of them gave reasons why they should include or exclude the appropriate paper and come to an agreement.

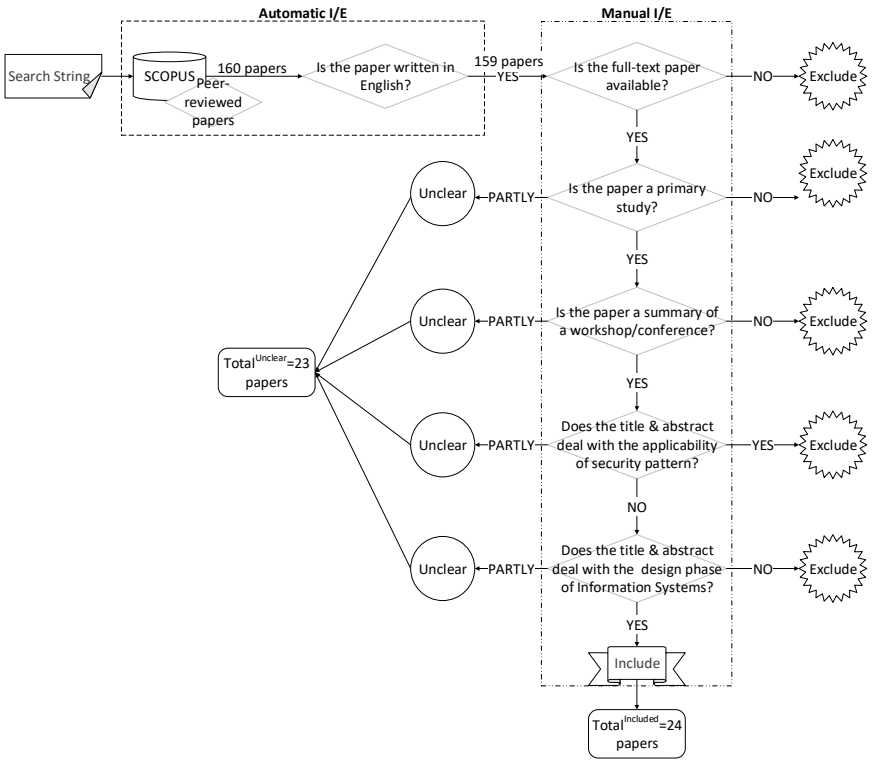

Figure 5.1.- Selection procedure

Table 5.5 shows a sum up of the first iteration of the selection procedure, taking into account the number of papers selected by the researcher and then by the advisor. Finally, out of those 23 unclear papers, six of them were included, two marked again as unclear an the rest of them excluded.

	Researcher	**Advisor**	**First iteration results**
Included papers	24	18	18
Excluded papers	112	139	139
Unclear papers	23	2	2

Table 5.5.- First iteration of the selection procedure

5.7.2.- Second iteration

To classify the two papers that were marked again as unclear, it was necessary not only reading the abstract but the conclusions and in one case, to go in depth in the content of the paper.

For developing the second iteration, only manual I/E criteria were considered. The results are shown in Table 5.6, and both unclear papers were finally excluded.

	Researcher	Advisor	Number of considered papers
Included papers	0	0	0
Excluded papers	2	2	2
Unclear papers	0	0	0

Table 5.6.- Second iteration of the selection procedure

5.7.3.- Validation of the selection procedure

To validate the selection procedure, the included and excluded papers were validated separately, and only the manual I/E criteria were considered. The manual criteria used by the advisor to develop the validation of the included papers, were the same that the researcher applied before.

The number of included papers after the first iteration where short, so all of them were sent to the advisor. The advisor had to read all the abstracts, which were the same sections read by the researcher, in order to reach a conclusion whether include or exclude them. Out of the 24 initially included papers, 12 were finally included and considered for the review. Half of the papers were studies not suitable for the study, and could be discarded thanks to the validation.

Regarding the number of papers initially excluded (159), and by time limitation, a sample of these papers was used to validate them. This sample consisted of the 10 % out of the overall excluded papers. After the advisor had read all the abstracts of that sample, he also marked as excluded all the papers considered in it. Due to this, it was considered that all the papers were correctly excluded out from the study.

At the end of this phase there were 18 included papers to continue with the study which can be seen in the appendix 7.2.

5.8.- Quality assessment

The quality assessment criteria can be useful to reinforce the decisions taken after the selection procedure phases. According to Kitchenham in [21], the quality assessment of the primary studies can provide more detailed inclusion/exclusion criteria, weight the importance of individual studies when results are being synthesized, guide the interpretation of findings and determine the strength of inferences and guide recommendations for further research.

In this review, data quality assessment was used for enhancing the inclusion/exclusion criteria and it was developed during the selection procedure.

Hence, taking as a reference the quality assessment criteria proposed by Kitchenham in [21], Inayat et al in [31] and Achimugu et al in [32], the quality assessment criteria that have been defined for this study, is shown in table 5.7. This table comprises questions that provide a measure of the extent to which a study is satisfactory and will

contribute to the scope of the review. The criteria cover trustworthiness, significance and thoroughness of the result papers.

Criteria	**Response Grading**	**Grade Obtained (Yes and Partially answers)**
Are the aims of the research clearly defined?	{1, 0.5, 0} (Yes, Partially, No)	18 papers, 100 %
Are the proposed mechanisms clearly described?	{1, 0.5, 0} (Yes, Partially, No)	13 papers, 72 %
Is the context of the research well addressed?	{1, 0.5, 0} (Yes, Partially, No)	15 papers, 83 %
Are the conclusions clearly stated?	{1, 0.5, 0} (Yes, Partially, No)	18 papers, 100 %
Based on the whole paper, how valuable is the research?	>80 % (Excellent) → 1; <20 % (Bad) → 0; in-between (Good) → 0.5	7 Excellent papers (39 %), 1 Bad (5 %), 10 Good (56 %)

Table 5.7.- Quality assessment criteria for paper selection

Each study was evaluated according to the quality assessment criteria presented in table 5.7. For a better categorization and rating of the studies, an ordinal scale was used instead of a dichotomous scale.

The first criterion assesses the aims of each study. This question was answered positively (taking into account Yes and Partially answers) in 100 % of the studies. The second criterion assessed whether the proposed mechanism in the study was clearly described and easy to understand. This question was answered positively in 72 % of the studies. In the third criterion, it wanted to be measured whether the context of each study was properly addressed and described. This question was answered positively in 83 % of the studies. The fourth criterion assessed the correctness of the conclusions stated in the selected papers. This question was answered positively in all the selected papers. The heuristic scores for the quality measures of the fifth criterion was based on the quality perception about the study of the researcher developing the review. The scores of the selected studies, which are based on their quality scores are referenced in table 5.7. For obtaining those scores, all the obtained grades in the previous measures were added. Hence, after performing the addition operation, those scores minor to 1.5, were marked considered bad results and they were given a percentage less or equal than 20 %. The scores higher than 3.5 were considered excellent papers, and were given a percentage higher than 80 %. Finally, those papers who obtained a score in between were considered good papers, and a percentage from 20 % to 80 % was given to them.

5.8.1.- Validation of the quality assessment criteria

To avoid subjective decisions about this phase, the quality assessment criteria was validated with a test-set of five papers that was sent to the advisor.

Table 5.8 shows the provided test-set, in which two papers out of five were considered excellent, another two papers out of five were considered good, and the last one was considered bad by the researcher.

Title	**Authors**	**Based on the whole paper, how valuable is the research?**	
		Researcher	**Advisor**
SoSPa: A system of Security design Patterns for systematically engineering secure systems	Nguyen P.H., et al.	85 % (1)	90 % (1)
An analytical study of security patterns	Ponde P., et al.	90 % (1)	90 % (1)
Classifying security patterns	Fernandez E.B., et al.	70 % (0.5)	50 % (0.5)
SCRIStUDIO: A security pattern integration tool	Bouaziz R., et al.	70 % (0.5)	70 % (0.5)
Enterprise security pattern: A new type of security pattern	Castellanos C., et al.	20 % (0)	20 % (0)

Table 5.8.- Validation of the quality assessment criteria

According to table 5.8, altough the researcher and the advisor gave some papers different percentages, at the end, those percentages were considered as a same level of quality. Thus, both researcher and advisor, had the same opinion about the quality of each paper. At the end, the decision was to discard those papers which were considered bad, and finally, only one (paper ID26 in appendix 7.2) out of eighteen was discarded, contemplating 17 papers for the next phase.

5.8.2.- Selected papers

Following, a brief summary of the 17 selected papers for the next phase (data extraction) is depicted. Each paper is referenced with an identifier or ID, which can be checked in the appendix 7.2 [1].

1. **Paper ID1**: this paper proposes a classification based on attacks of the STRIDE (STRIDE stands for Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service and Elevation of privileges) model to help an architect or developer to select appropriate patterns in order to fulfill security requirements given to her. It achieves the validation research level as the authors validate the correctness and usefulness of the classification. Furthermore, they introduce a taxonomy based on attack patterns that enables architects and developers to select appropriate patterns according to possible attacks.

2. **Paper ID10**: the paper presents a MDS (Model-Driven Security) framework based on a System of Security design Patterns (SoSPa) that allows practitioners to

[1] Consider that paper ID56 have been discarded from the final selected papers.

systematically address multiple security concerns in secure systems development. It achieves the evaluation research level as there is a dedicated section explaining the procedures they followed to develop a controlled experiment with practitioners. The security patterns are collected and specified as reusable aspect models (RAM) to form a coherent system of them. SoSPa also provides a refinement process supported by RAM to derive the detailed security design patterns closer to implementation. SoSPa aims at systematically addressing the globally accepted security concerns such as confidentiality, integrity, availability, accountability. Thus, SoSPa is composed of an extensible set of security solution blocks which consists of interrelated security design patterns.

3. **Paper ID14**: solution proposal in which researchers propose SCRIStUDIO (SeCurity patteRn IntegratIon Studio), an integrated tool which give developers the possibility to integrate security patterns in their application and especially in component based applications based on UML language. In a previous work they proposed SCRIP (Paper ID 26), a process that enables designers to use solutions based on security. In this work, they propose the automation of this process via a plug-In called SCRIStUDIO. The main objective of this plug-In is to allow non-expert security developers to integrate different security properties throughout the development cycle of a component based application. It is based on four "free software" tools that all run on one unique IDE (Integrated Development Environment): Eclipse.

4. **Paper ID15**: solution proposal research that presents a template which considers sections such as the context of the problem, the description of the problem to be solved together with the threats, possible solutions, technological considerations and examples of known incidents. This template extends the sections of security patterns. The specific new concepts that can be applied may include:

   a) The type of information assets to protect (data, applications, and code and configuration).

   b) The security realms where the assets are stored.

   c) The security policies associated with the information assets.

   d) The general features of who (customers, employees, or technical users) or what (systems) will access them.

5. **Paper ID26**: solution proposal paper that introduces a process, called SCRIP (SeCurity patterRn Integration Process) and an associated tool for automatically integrating security patterns into component-based models, and producing an executable secure code. SCRIP is an iterative structured process that helps designers selecting the right security pattern by defining different tasks and roles.

6. **Paper ID28**: the researchers use a specific proposal of 4 annotation types that has been developed at KU Leuven for teaching purposes and has been applied to a catalog of 35 security patterns [45]. The annotations in the catalog cover four dimensions:

   a) The security objective(s) for which the pattern provides a solution.

b) The applicability of the pattern to either the high-level architecture of a system or its detailed design.

c) The trade-off labels that indicate the positive/negative impact of the pattern.

d) The relationships among patterns.

The paper achieves the validation research level as the study divided 45 teams of master students in two treatment groups. Each student had to perform four design tasks involving the hardening of a software architecture via security patterns using the four proposed annotations.

7. **Paper ID30**: solution proposal research in which the authors describe the different sections of the patterns. They also state that it is necessary to be more concise in the definition of those aspects and claim that an extension of this aspects is needed.

8. **Paper ID32**: the paper presents a framework called SCRI-PRO (SeCurity Pattern Integration apPROach) based on pattern navigable map called Security Pattern Cartography. SCRI-PRO is the evolution of the process SCRIP described in paper ID26. The paper achieves a validation research level because the implementation and the experimentation of the framework were done in a prototype as a partial validation of the authors' approach.

9. **Paper ID38**: solution proposal that proposes an extension in the definition of the essential sections of the security patterns. The sections of the pattern template the authors provide are following explained:

   - Context: they model the context with an aggregation of *Domain Property* (a fact about the world), via *and/or* operators.
   - Problem: they use one or several *Goals* or *Softgoals* to capture stakeholders' needs concerning a problem.
   - Force: they use *Domain Assumption* to model such forces.
   - Solution: they model them as Tasks, which the system-to-be performs to satisfy its requirements.

   To facilitate this extension in patterns' definiton, they provide three complementary advices:

   a) Some patterns not only present forces in the Force section, but also in the Consequence section.

   b) Apart from the general context specified in the Context section, the problems, forces, and solutions may involve particular contexts.

   c) Some solutions are described in a very abstract manner.

10. **Paper ID40**: the paper proposes a new pattern list classification based on interrogatives, purposes, system types, security objectives or location among others. It also proposes a template for classifying those patterns and three functions (pseudo-code) to choose the correct security pattern. The paper achieves a validation research level because they develop a case study in a laboratory.

11. **Paper ID41**: solution proposal paper which presents a methodology based on merging techniques and verifications described in OCL (Object Constraint Language) language. The process follows four phases:

    a) Preparation: extracts from the pattern the solution description and its related constraints into a UML model. Pattern properties and constraints are formalized using OCL.

    b) Elicitation: builds a bridge between the application and the pattern.

    c) Merge: does the merging between them and generates an integration trace.

    d) Adaptation: offers the possibility to make changes by letting the user refining the new application.

12. **Paper ID46**: solution proposal where the authors define a new methodology, which consists on a set of security pattern application rules (SPARs) to automate the integration of security patterns into software components. These rules are deduced through the relationships between security concepts of the selected pattern and the corresponding UML profile.

13. **Paper ID53**: the goal of this paper is to construct a common way to implement secure applications using security patterns at design level for several domains proposing a re-definition of the common sections (context, problem, solution...) of the security patterns. The authors propose three layers:

    a) *Component Metamodel*: describes the fundamental concepts that are already used with success in CBSE (Component-Based Software Engineering).

    b) *Component Security Pattern*: is a chosen security pattern according to the security requirements for the particular application. The pattern template includes name, intent, context, problem, structure and participants.

    c) *Component Security UML Profile*: it embeds security expertise provided by the security pattern presented. This profile is resulting from the *Component Metamodel* that present the domain concepts, and the *Component Security Pattern* that chooses the security pattern that presents security solution according to the problem faced.

    The paper achieves the validation research level as it presents a case of study of GPS system.

14. **Paper ID58**: this solution paper proposes a re-definition of the intern structure of security patterns. Authors claim that a pattern definition should contain: Context, Problem, Solution, Structure, Dynamics, Implementation, Consequences, Known uses and Related Patterns.

15. **Paper ID60**: this solution paper presents a possible classifications based on architectural concerns, architectural layers, and some relationships between patterns. With their classification, they can define patterns at all levels. This allows a designer to make sure that all levels are secured, and also makes easier propagating down the high-level constraints. There are two differentiated levels:

highest level and operating system level. At the highest level there are patterns that describe the use of security models to define access control to the application objects. At the operating system level there are only security patterns. They also propose *automatic relationships* between patterns and a pattern diagram uses these classifications to help the designer navigate in the design space.

16. **Paper ID72**: the paper presents a new definition of the sections that may have a security pattern. The re-definition of the patterns should be decomposed into new components, roles, requirements, expectations, and residual goals. Furthermore, this re-definition follows two connected formal models:

    a) The refined model contains the internals of the pattern and is defined by a security expert who undergoes the task of interpreting the pattern documentation, translating it into a sound formal model, and verifying its security properties.

    b) The abstract model is directly used by the architect, who can more easily integrate it in larger models (the whole design) to reason about compositional security properties.

17. **Paper ID155**: validation research which proposes a comprehensive pattern-driven security methodology named ASE: conceptual security framework aspect. ASE presents in an structured way how security patterns can be applied following that methodology, which is divided in the following phases:

    a) *Security requirements determintation* phase: its intent is to specify both prescriptive and resultant security requirements for the target system.

        1) *Adversary modeling* stage: it models attackers and their potential attacks (e.g. in the form of threats) to a system.
        2) *Security modeling* stage: it incorporate security countermeasures (e.g. via the application of security patterns, security aspects) as appropriate representations into a system's models.

    b) *Countermeasure introduction* phase: introduces security countermeasures by converting problem-space requirements into solution-space goals and policies, and incorporating these into relevant models of the target system.

        1) *Countermeasure identification* stage: maps problem-space security requirements to solution-space abstractions.
        2) *Security modeling* stage: incorporates security countermeasures (e.g. via the application of security patterns, security aspects) as appropriate representations into a system's models.
        3) *Security verification* stage: verify that the introduced constructs (design, code, etc.) provide the necessary level of security and adequately protect the system against relevant threats.

    c) Re-iteration of the security requirements determination phase.

5.9.- Data extraction strategy

Based on the guidelines provided by Kitchenham in [21], a data extraction strategy was defined to identify and extract relevant information from the 17 included primary papers. Data extraction strategy includes recording ideas, concepts, contributions and findings of each of the 17 studies, a template was set up. This template assures having information in a common repository and in the same format, so future analysis is facilitated.

The aim of this phase was to synchronize selected studies in order to enhance their clarity and a possible comparison between them. This would also help in the identification of precise answers to the research questions. To get this aim, a two-iteration process was developed. The first iteration, was based on research questions in order to identify data that will be extracted. After data extraction, the main outcome of this analysis was the redefinition of applicability scope for this study, now considering different facets: selection, structure and holistic (deeper explanation in subsection 6.1). Then, both researcher and advisor defined a comparison criteria that would be used in the next stage of the study. These criteria were used to update the data to extract. Since this point, the comparison criteria for synthesis and the data to extract were strictly related. Finally, in the second iteration the final data for the next stage in the study were gotten.

Figure 5.2 depicts the followed procedure. First iteration included the definition of the data to extract, the data extraction, the analysis of the previous data and finally, the definition of the comparison criteria. Second iteration included the redefinition of the data to extract, the data extraction and finally, the total conformation of data.

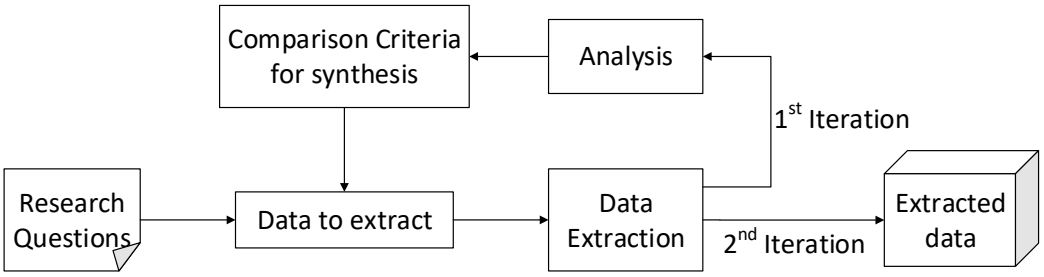

Figure 5.2.- Data extraction strategy

5.9.1.- Data to extract

In this subsection, it is presented the different data extracted from each study. This data to extract can be classified in four different types: common data (extracted from every paper, no matter to which facet they belong to.), data for intern structure facet mechanisms, data for selection facet mechanisms and data for holistic facet mechanisms.

1. Common data:

- **Bibliographic data**: including authors, title, year of publication, reference, URL (Uniform Resource Locator) and source title.
- **Type of study**: journal, conference papers, articles and book chapters.
- **Methodology elements:** they show the different kind of methodology elements according to Software Engineering Metamodel for Development Methodologies (SEMDM) in [36] as well as other elements that were reported in the papers. In fact, the mechanism we are looking for can be:
  - *Classification:* a pattern classification helps designers to know what patterns can be used without the need of reading, analyzing and understanding every pattern.
  - *Taxonomy:* a scheme that partitions a body of knowledge and defines the relationship among the pieces. The main difference between taxonomies and classifications is that taxonomies try to establish a more exhaustive relationships between the security patterns and how can they interact between them when applying them.
  - *Process/Method/Methodology:* cohesive collection of endeavor-specific process components that models the enactment of an endeavor.
  - *Tool:* a software product providing automatic support for security patterns' application.
  - *Work flow:* a work unit that models a cohesive collection of tasks that either produces a new version of a single work product or provide a single service.
  - *Model:* it represents an abstraction of something that captures its essential characteristics (for some specific purpose) while ignoring unimportant or diversionary details.
- **Mechanism's description**: it is a description of how the mechanism works.
- **Exclusive security feature**: some specific characteristic only for use while applying security patterns.
- **Mechanism's application context**: it is the domain in which the mechanism is supposed to work on (Internet of Things, Cloud Computing, or if it can be applied in any information system). This also makes reference whether the mechanism can be applied over all the SDLC or only during the design phase and any other relevant context feature.
- **Maturity level of the mechanisms used for the applicability of security patterns in the design phase of an Information System**: research types' definition according to Wohlin et al. in [33] and Petersen in [34]. Applicability's concept is related to something that is suitable to be applied. Considering this as a true statement, the initial beliefs were that most of the papers could report empirical results rather than non-empirical ones. Empirical results are positive and typically involve systematic collection and analysis of data like observation and evidence [46]. Hence, for this review, validation and evaluation research can be considered empirical methods. On its behalf, non-empirical results in this review are very close to the empirical ones, except for they only propose a solution for a known problem and prove their proposal by examples. This example proofs can be considered close enough to an empiric validation. There are

more research types presented in [33] and [34] that can be considered non-empirical methods, but for the aim of this study, philosophical, opinion and experience papers are not considered, as they do not fit in the empirical goal for the study.

- **Research method**: only defined when the papers' research type is a validation or evaluation research. Research methods defined by Petersen et al. in [35] were used. The research methods that can be found in the researches are divided depending on the type of research type they are:
  - Evaluation research: the possible research methods that can be considered in an evaluation research are an industrial case study, a controlled experiment with practitioners, a practitioner targeted survey, an action research or an ethnography, among others.
  - Validation research: the possible research methods that can be considered in an validation research are a simulation as an empirical method, a laboratory experiment, a prototype, a mathematical analysis and proof of properties or an academic case study among others.
- **Design activity:** it shows to which design activity the outcome of security patterns' applicability was aimed to. We used those activities proposed in [43] and [14]:
  - Architectural design: it identifies the global system structure, main modules or components with their proper relations and how these components are distributed.
  - Interface design: it presents a clear definition of interfaces among the components of the system.
  - Component design: it defines the operation of each component of the system.
  - Database design: it represents the data system structure and how the data will be represented in a database.
- **Results**: how good or bad are the results when applying those mechanisms or how far they went in the research.
- **Improvements on design**: in the case of success on applicability, it shows how the patterns' applicability was improved (e.g. more accurate outcomes, less design time, and so on).
- **Mechanism's possible limitations**: restraints documented by the authors when applying the mechanism.
- **Challenges for the mechanisms**: ways of improvement of the mechanisms or future challenges they are trying to solve.
- **General notes**: researcher's own conclusions after reading the paper.
- **List of references of the paper**: list of references provided by Scopus that can be useful for snowballing.

2. Data for structure facet mechanisms:

   - **Patterns' structure change**: the researchers think that the definition of security pattern is not complete or might need of modifications. Hence, they

propose either a *re-definition* of the sections of a pattern, trying to add more detailed information or an *extension* of the sections, in which they propose a new way of definition adding more sections to it. Maybe there are *other* proposed changes that are not reflected in the values, and should also be considered.

3. Data for selection facet mechanisms:

   - **Process description**: the steps followed for selecting a pattern or patterns for improving applicability.
   - **Pattern selection number**: it is important to know the final result of the selection procedure, single or multiple patterns.

4. Data for holistic facet mechanisms:

   - **Process description**: description of the steps needed to apply a security pattern.
   - **Description' formality**: how well and formal are described the mechanisms (e.g. the methodology is detailed or is described in a lax way).
   - **Graphical representations**: any diagram or scheme that helps to understand the mechanism.

6

Results and discussion

This chapter presents and discusses the findings of the review. It starts by presenting an analysis about applicability, its definition and facets. Then, an overview of the selected studies is presented. After that, the main contribution of this study is demonstrated as a detailed description of the findings in line with the research questions.

6.1.- Applicability analysis

At the beginning of the development of the systematic literature review, we understood applicability as taking an already defined pattern for applying and using it while designing information systems. After the first iteration in the data extraction stage, it means after reading 17 papers once, our applicability's definition changed. Authors referenced applicability not only as the mere use of patterns, but they took different perspectives. Some of them focused on selecting patterns, others on the intern structure of the patterns and the last ones, considered a holistic approach, which included selecting a pattern, understanding its intern structure and then, knowing how to use it for designing information systems. We are referring to these different approaches as facets, and are respectively described below:

1. **Structure facet:** it refers to how security patterns are defined taking into account the different sections (context, problem, solution, etc.). Papers ID15, ID30, ID38, ID53, ID58, ID60 and ID72 (appendix 7.2) consider this approach. All the authors that took into account this facet, claim that the way security patterns are defined is not the most adequate for helping designers. Hence, there was a need to develop a template to define them with the help of the taxonomies or classifications that have been reported in the papers related to this facet. Authors also took into account the different levels of abstractions with which patterns could be defined. A level of abstraction, should be a low level, so designers could know which pattern should be applied in any situation. Another important characteristic to acknowledge is the semantics and syntax used when defining security patterns. This could be solved by the definition of a new vocabulary or taxonomy that simplifies and normalizes the terms that are used while writing security patterns. Finally, it would be appreciated if the definition of patterns include some kind of graphics, such as use diagrams, models, etc. thus, improving the applicability of security patterns.

2. **Selection facet:** this facet is about selecting the right pattern or set of patterns while designing information systems. It is important to consider that when designing a system, maybe not only one security pattern would be selected, but a set of them. The papers where the selection facet is referenced are known as

selection papers and they are papers ID1, ID10, ID26, ID28, ID32 and ID40 (appendix 7.2). Some authors like Sametinger et al. in [29] and Nguyen et al. in [39], among others, propose in their studies taxonomies, classifications, processes o methods that would help designers to select the most appropriate security patterns for each scenario.

3. **Holistic facet:** this facet is about knowing which pattern or set of patterns could be selected understanding how they are structured and then knowing how to apply them in order to design an information system. The papers which report the holistic facet are ID14, ID41, ID41 and ID155 (appendix 7.2). To fulfill this whole process, it is essential to have a methodology, process, work flow or even a tool that helps designers to really apply security patterns effectively. These kinds of mechanisms are presented by Bouaziz in [41] and Uzunov et al. in [26], among others.

6.2.- Overview of selected studies

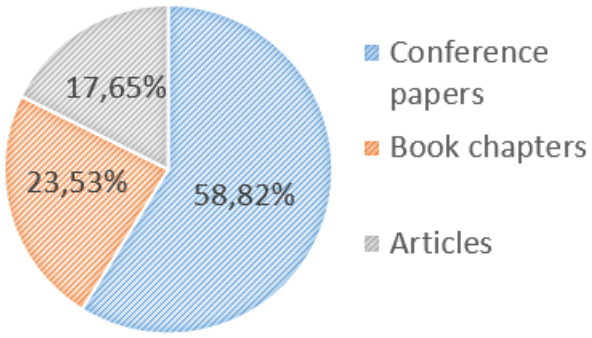

Figure 6.1.- Overview of collated studies

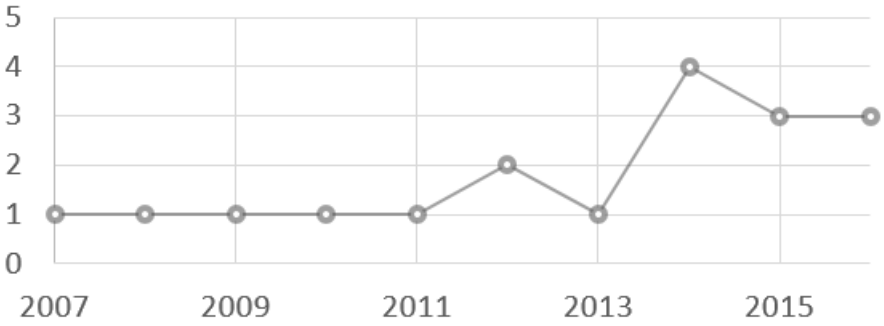

Figure 6.2.- Paper publication distribution per year (from 2007 to 2016)

17 papers were selected for this review. Among them, 10 papers appeared in conference proceedings, 4 papers were from book chapters and 3 papers were articles. The respective numbers and percentages of the selected papers are represented in Figure 6.1; while the number of papers by year of publication is depicted in Figure 6.2.

Figure 6.2 depicts that there is a tendency of studying the applicability of security patterns since 2007. Now, the number of papers per year about this topic has increased as people are still investigating on it.

6.3.- Research questions' analysis

6.3.1.- *RQ 1.* Which mechanisms have been used when applying security patterns while designing an Information System?

There are several mechanisms that have been used by authors when applying security patterns while designing information systems. Figure 6.3 shows the different found mechanisms: taxonomies, classifications, methodologies, tools and models. Some mechanisms were only seen in some types of facets, for example taxonomies and classifications were not seen in the holistic facet. It will be further explained in RQ 1.1 in subsection 6.3.1.1.

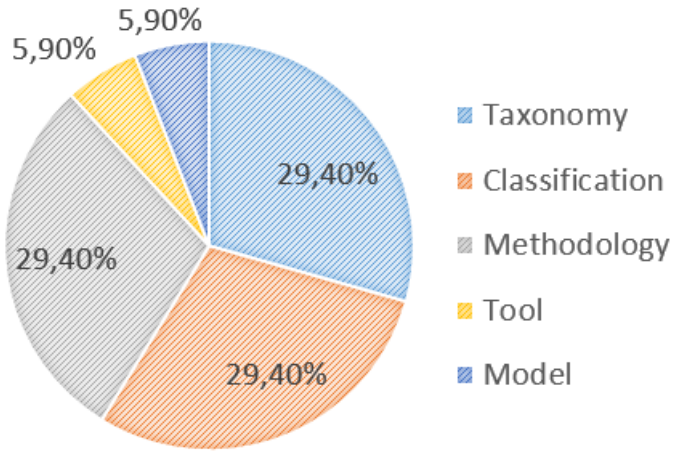

Figure 6.3.- Mechanisms used by authors when applying security patterns

6.3.1.1.- *RQ 1.1.* Which of the defined mechanism is the most used?

Methodology Element	Total papers (17)	Selection papers (6)	Structure papers (7)	Holistic papers (4)
Taxonomy	5 (29.4 %)	2 (33.33 %)	3 (42.85 %)	0 (0.00 %)
Classification	5 (29.4 %)	2 (33.33 %)	3 (42.85 %)	0 (0.00 %)
Methodology	5 (29.4 %)	2 (33.33 %)	0 (0.00 %)	3 (75.00 %)
Tool	1 (5.9 %)	0 (0.00 %)	0 (0.00 %)	1 (25.00 %)
Model	1 (5.9 %)	0 (0.00 %)	1 (14.3 %)	0 (0.00 %)

Table 6.1.- Methodology elements analysis

According to Table 6.1, the most common methodology elements, taking into account the whole set of selected papers, are taxonomies (29.40 %), classifications (29.40 %) and methodologies (29.40 %). Focusing on each facet of papers, most of the methodology elements reported in selection and intern structure papers were also taxonomies (42.85 % in structure facet and 33.33 % in selection facet) and classifications (42.85 % in structure facet and 33.33 % in selection facet), while 75 % of holistic papers reported a methodology to help designers in security patterns' applicability.

There is a correspondence between the most used types of the methodology elements and the different facets. When talking about selection facet, the most used methodology elements are classifications, which means that authors are mainly focused on the importance of how to organize security patterns.

The authors of the structure facet want to normalize how the patterns are internally defined, so taxonomies or classifications fit to accomplish this task.

The holistic facet reports the highest number of methodologies (75 %) as mechanisms for applying security patterns, which corresponds to a 29.40 % considering all the facets. Considering the holistic view (as described in section 6.1), and the previous percentage, the most suitable mechanisms is a methodology. This is because a methodology explains in a guided way how to apply patterns in the different contexts and gives designers a set of systematized steps and an ensemble of elements that will used those steps. Furthermore, there is still a lack of tools that could help designers to materialize the applicability of security patterns, as only paper ID14 reported one based on Eclipse.

6.3.1.2.- *RQ 1.2.* What are the descriptions of the mechanisms?

Knowing mechanisms' description will be valuable to understand and analyze them, even for doing further research such as improvements, replications, etc. In this section, the description of the mechanisms of each facet will be presented.

When talking about the structure facet, researchers think that the definition of security patterns is not complete. Hence, they propose either a re-definition of the sections of a pattern, trying to add more detailed information, or an extension of the sections, in which they propose adding more sections to it.

Table 6.2 shows that 57.14 % authors propose an extension of the current definition of security patterns. This could help designers to better understand the way a pattern is defined. The rest of the authors (42.86 %) maintain that it is better to re-define the structure of the patterns, adding more detailed information such as an example of how the patterns can be applied.

	Extension or re-definition?
ID38	Extension
ID53	Re-definition
ID72	Re-definition
ID58	Re-definition
ID15	Extension
ID60	Extension
ID30	Extension

Table 6.2.- Structure papers' criteria and analysis

Table 6.3 refers to the selection facet. It displays a summary of the description of the steps provided in each selection paper. All the papers belonging to the selection facet report taxonomies, classifications and methodologies with structured steps that help designers to select the right set of patterns. A deeper explanation of the steps mentioned in Table 6.3 is described below:

	Steps & Description	**Single/Multiple pattern election**
ID1	No	Multiple
ID10	1. Building security solutions. 2. Defining mappings. 3. Weaving the security solutions into the base system.	Multiple
ID26 **ID32**	1. Elicitation. 2. Modeling. 3. Implementation.	Multiple
ID40	*ChooseCluster()* *SecuritySolution(SR,C)* *PatternSelect(SA,C)*	Multiple
ID28	Four annotations applied to a catalog of 35 security patterns.	Multiple

Table 6.3.- Selection papers' summarized description

- **Paper ID10**: this paper proposes a classification based on attacks of the STRIDE model to help an architect or developer to select appropriate patterns in order to fulfill security requirements given to her. The given steps for the classification are the following:

  1. Constructing security solutions from the security patterns in SoSPa (System of Security design PAtterns): for each security concern, the interrelations specified in the feature model (Figure 6.4) are used to select the most appropriate security design patterns. This step derives a detailed RAM (Reusable Aspect Models) design of a customized security solution for the concern, including its customization interface and usage interface. This woven RAM model of the authentication solution later can be integrated into a base system model via its customization interface.
  2. Defining mappings to integrate the newly built security solutions to a base system model: for each selected security pattern, use the customization interface of the generated design to map the generic design elements to the application-specific context. This step generates the mappings of the parametrized elements in the security design pattern with the target elements in the target system design. Any constraints/conflicts between mappings of all the selected security design patterns need to be resolved.
  3. Weaving the security solutions into the base system model: all the security solutions are automatically woven into the target system design. The mappings from previous step are the input for this weaving process.

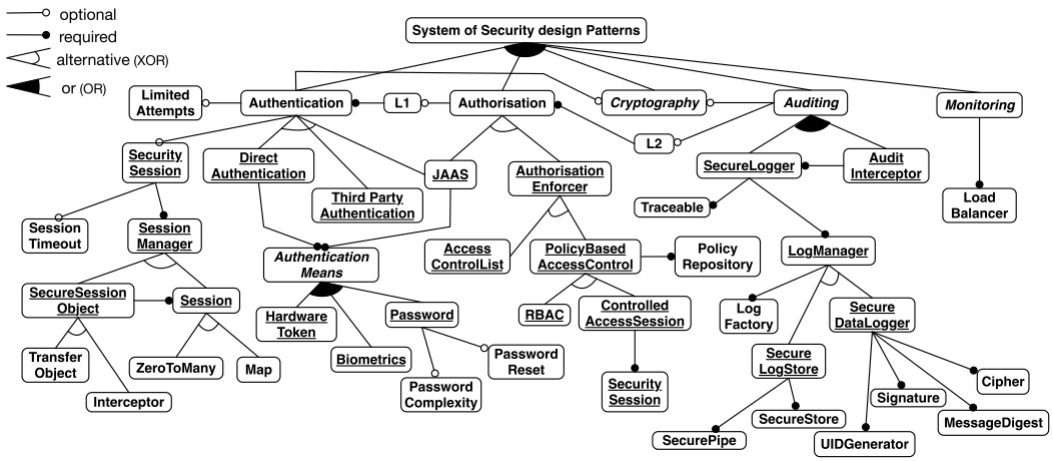

Figure 6.4.- Partial feature model of SoSPa [39]

- **Papers ID26 and ID32**: paper ID26 introduces a process called SCRIP while paper ID32 presents the evolution of SCRIP, a framework named SCRI-PRO based on a pattern navigable map called Security Pattern Cartography which gives the following phases:

  1. Elicitation phase: it comprises two TaskUses [1]. The first one is *TaskUse Define Security Profile* which is performed by the security and component specialist. It consists on mapping the concepts of the chosen security patterns with concepts of the component meta-model). The second TaskUse is *Define Security Pattern application rules* that is performed by the same specialist and it defines a set of rules to automate the integration of security patterns into software components. These rules are deduced through the relationships between security concepts of the selected patterns and the corresponding UML profile.
  2. Modeling phase: it produces a secure component model after having applied one or several security patterns to an initial application component model. It is carried out by the TaskUse *Application Design*, which model the functional application design. The *Software Designer* RoleUse carries out this TaskUse and apply the security pattern.
  3. Implementation phase: it is dedicated to produce a secure application code through three TaskUse. The first one is *Generate the functional code* which aims to generate the Functional code of the component based application. The second TaskUse is *Generate the functional code* that takes as input the secure application model to define aspects. During this TaskUse, the *Security specialist* and the *Software designer* RoleUses collaborate to generate «aspect code»). The last TaskUse in this fase is *Generate secure application code*, which takes as input the application functional code and the aspects code» to produce a code for secure application.

[1] According to SPEM [7]: a *TaskUse* describes a piece of work performed by one *RoleUse*, which may consist of atomic elements called Steps. A *RoleUse* defines responsibilities over specific artifacts (anything produced, consumed, or modified by a process), which are consumed or produced in specific activities.

- **Paper ID40**: the paper introduces a new pattern list classification based on system types, security objectives or interrogatives among others, as well as it proposes a template for classifying those patterns and three functions to choose the correct security pattern. The proposed pattern search starts from the high level clusters and continues to the lower level ones in the cluster hierarchy. The algorithm proceeds iteratively for each Requirement (Ri) in the set of high level Security Requirements (SR). For each requirement Ri, the first step is to extract Security Attributes (SA) based on the general schema (lifecycle stage, quality attribute, threat, trust boundary and countermeasure). This step may be performed iteratively to convert the high level requirements to specific security attributes based on the chosen categories. The *PatternSelect()* algorithm returns the patterns which satisfy the attributes. *PatternSelect()* is a recursive algorithm which works at each level of the cluster hierarchy. The final solution is the union of all patterns returned by *PatternSelect()* for each requirement. The attributes are matched with the dominant attributes of each cluster at the same level by the *ChooseCluster()* algorithm. *ChooseCluster()* returns a set of clusters *Cmatch* having the attributes as its dominant attributes. Patterns are selected from the cluster *Cmatch* if it is a leaf cluster using algorithm *GetPatterns()*. If not, the same algorithm is applied to the child clusters of *Cmatch* recursively.
- **Paper ID28**: in this paper the authors propose an specific proposal of 4 annotation types that has been developed at KU Leuven for teaching purposes and has been applied to a catalog of 35 security patterns. The annotations in the catalogue cover four dimensions:

    1. The security objective(s) for which the pattern provides a solution, for example confidential data transmission, data storage integrity, accountability and so on.
    2. The applicability of the pattern to either the high-level architecture of a system or its detailed design. This annotation also indicates whether the pattern is going to be applied in the core of the system or rather in its deployment environment.
    3. The trade-off labels, which indicate the positive or negative impact of the pattern on other software qualities, such as performance or maintainability.
    4. The relationships among patterns, e.g., functional dependency, mutually-exclusive conflict, alternative, and so on.

Table 6.4 refers to the holistic facet. It displays a summary of the description of the steps provided in each selection paper, the considered grade of fullness that expresses how well and formal are described the mechanisms, and finally if there are any diagram or scheme that helps to understand the mechanism. A further explanation about the description of the steps for each paper is followed presented:

	Steps & Description	Fullness of the description	Diagram
ID14	1. Ensure a dynamic interaction between the user and the GUI through an XML parser. 2. The parser runs through the diagram and extract the list of artifacts. 3. The artifacts are passed as parameters of the GUI. 4. ATL models transformation modules that are processed by text analyzers to inject the previously stored user choices.	0.90	Yes
ID46	1. Ensuring the correspondence between the main pattern concepts and the correspondence model elements 2. Ensures the automatically mapping the other security pattern concepts to the corresponding model elements by applying the respective stereotypes defined in an UML profile	0.70	Yes
ID41	1. Preparation. 2. Elicitation. 3. Merge. 4. Adaptation	0.50	No
ID155	1. Security Requirements Determination phase: 1.1.Adversary Modeling stage. 1.2. Security Modeling stage. 2. Countermeasure introduction phase: 2.1. Countermeasure Identification stage. 2.2. Security Modeling stage. 2.3. Security Verification stage 3. Re-iteration of the Security Requirements Determination phase.	0.90	Yes

Table 6.4.- Holistic papers' summarized description

- **Paper ID14**: the paper proposes an integrated tool (SCRIsTUDIO) which gives developers the possibility to integrate security patterns in their application following the next steps:

  1. To ensure a dynamic interaction between the user and the GUI, SCRIsTUDIO manages a diagram through an XML parser.
  2. This parser runs through the diagram and extract the list of artifacts.
  3. These artifacts are passed as parameters of the GUI. Through these interfaces, the user can modify and/or add attributes dynamically. The proposed tool provides to the user through several interfaces to set the security configuration file according to its application.

4. An XML file is generated and recorded in the security configuration defined by the designer. Furthermore, these interfaces offer to the user the choice of selecting which security patterns he would like to apply to its component diagram and the artifacts on which he prefers to apply security stereotypes of the chosen pattern.
5. ATL (ATLAS Transformation Language) models transformation modules that are processed by text analyzers to inject the previously stored user choices. After the transformation was configured, the ATL modules of security pattern application on a component-based model can be performed. As result of this transformation, ATL produces the second output of the proposed tool that is a component diagram annotated by security stereotypes. After producing the secure component diagram and the security configuration file in the modeling phase of the process, the proposed tool reuses these two results as input of the code generation phase.

- **Paper ID46**: the paper defines a new methodology to automate the integration of security patterns into software components. To do this, the authors proposed the following steps:

  1. Ensuring the correspondence between the main pattern concepts and the correspondence model elements (specified as a component, a connection or a port). For each security pattern, it is selected the main concept that should be applied by the designer.
  2. The designer ensures the automatically mapping which is performed by applying the respective stereotypes defined in the corresponding UML profile.

- Paper ID41: it presents a methodology based on merging techniques and verifications described in OCL (Object Constraint Language) language. The process follows four phases:

  1. Preparation: extracts from the pattern the solution description and its related constraints (preconditions and post conditions) into a UML model. Pattern properties and constraints are formalized using OCL.
  2. Elicitation: builds a bridge between the application and the pattern.
  3. Merge: does the merging between the application and the pattern and generates an integration trace.
  4. Adaptation: offers the possibility to make changes by letting the user refining the new application.

- **Paper ID155**: it proposes a comprehensive pattern-driven security methodology named ASE, which presents in an structured way how security patterns can be applied following the previous mentioned methodology. This methodology is divided in the following phases:

  1. *Security requirements determination* phase: its intent is to specify both prescriptive and resultant security requirements for the target system.

   a) *Adversary modeling* stage: it models attackers and their potential attacks (e.g. in the form of threats) to a system.
   b) *Security modeling* stage: it incorporates security countermeasures (e.g. via the application of security patterns, security aspects) as appropriate representations into a system's models.
2. *Countermeasure introduction* phase: it introduces security countermeasures by converting problem-space requirements into solution-space goals and policies, and incorporating these into relevant models of the target system.
   a) *Countermeasure identification* stage: it maps problem-space security requirements to solution-space abstractions.
   b) *Security modeling* stage: it incorporates security countermeasures (e.g. via the application of security patterns, security aspects) as appropriate representations into a system's models.
   c) *Security verification* stage: it verifies that the introduced constructs (design, code, etc.) provide the appropriate level of security and adequately protect the system against relevant threats.
3. Re-iteration of the security requirements determination phase.

The 75 % of papers belonging to the holistic facet reports methodologies. According to the column fullness of description, the reported methodologies (papers ID46, ID41 and ID155) follow a formal description of the mechanisms that could be replicated. Furthermore, the only tool that helps the designer to apply security patterns through an Eclipse environment is quite detailed and easy to understand and follow, as it presents screenshots of the environment explaining the steps that should be followed.

Finally, it is worth mentioning that the main differences between the descriptions in selection facet and the holistic one is that selection facet reports more static mechanisms, like classifications and taxonomies. The main aim in the selection facet is helping the designer to select the right set of patterns for an specific information system. This selection does not involve a dynamism, which is required in the methodologies reported in the holistic facet. The holistic facet explain how to use those patterns that were previously selected, and then that sense of dynamism is needed.

6.3.2.- *RQ 2.* What is the maturity level of the found mechanisms?

Total papers	Validation Research	Evaluation Research	Solution Proposal
17	7 (41.2 %)	1 (5.9 %)	9 (52.9 %)
Selection facet (6 papers)	4 (66.66 %)	1 (16.67 %)	1 (16.67 %)
Structure facet (7 papers)	2 (28.57 %)	0 (0.00 %)	5 (71.43 %)
Holistic facet (4 papers)	1 (25.00 %)	0 (0.00 %)	3 (75.00 %)

Table 6.5.- Maturity level analysis

Table 6.6 shows that out of the total selected papers, only 8 were considered empirical papers (7 validation researches and 1 evaluation research) and the remaining 9

papers were considered solution proposals. So, the initial belief about that there should be more empirical results, as the concept of applicability were more kindly related to it, was not right. However it is worth mentioning that almost half of the results were empirical, which is a reasonable number.

Specifically, when referring to the selection papers, there are more empirical papers than non empirical ones. Out of 6 selection papers, there are 5 empirical papers (4 validation research and 1 evaluation research) and only 1 non empirical papers. There is a tendency in the selection facet for achieving more empirical proofs. These results lead us to think that in the selection facet, there are more mature proposals than in the remaining facets, as they have been at least validated and they could be nearer of their replication in other environments of study.

In the remaining facets (structure and holistic), there are more solution proposals (5 and 3 respectively) than validation researches (2 and 1 respectively), and there are no evaluation researches in neither both facets. This proves that there is still a lack of empirical evaluations to prove the possible effectiveness of restructuring the intern structure of security patterns and also the effectiveness of the methodologies reported in the holistic papers.

Only validation and evaluation researches have research methods associated to them. Table 6.6 shows the different research methods that have been reported in the studies. Considering the 8 out of 17 selected papers that are either a validation or evaluation researches, the most reported research method is the academic case study (62.5 %). This can be because case studies give researchers a chance to study one aspect of a real-world problem, like the applicability of security patterns in detail from many different viewpoints, and it is the most controlled research method. It is important to mention that there is still a lack of research methods related to evaluation researchs, as only a 12.5 % out of the total of selected papers were found.

Total papers (Validation & Evaluation Researches)	**Prototype**	**Controlled Experiment with practitioners**	**Academic Case Study**
8	2 (25 %)	1 (12.5 %)	5 (62.5 %)
Selection facet (5 papers)	2 (40 %)	1 (20 %)	2 (40 %)
Structure facet (2 papers)	0 (0.0 %)	0 (0.0 %)	2 (100 %)
Holistic facet (1 paper)	0 (0.0 %)	0 (0.0 %)	1 (100 %)

Table 6.6.- Research method analysis

6.3.3.- *RQ 3.* How have been the results when applying the reported mechanisms?

Regarding success on the applicability, the 52.95 % of the papers did not report their success when applying the mechanisms, and only the remaining 47.05 % reported were successful.

Further in the analysis of these results, we wanted to explore which were the improvements in security patterns' application process. First thing that comes into our minds as an improvement when applying the reported mechanisms is a reduction on design time. The reported mechanisms should reduce the amount of time of selecting and applying the right pattern to build an IS. But, according to Yskout et al in paper ID28 (selection facet), time is not improved, but efficiency when selecting the right set of patterns. This efficiency is measured according the formula *E=Nselect/Nview*, where *E* is Efficiency, *Nview* is the total number of patterns that the team assessing the research looked, and *Nselect* is the number of patterns that the team finally selected for instantiation in the architecture for that task. Yskout's team also claim that every patterns must be viewed more times than be selected, and thus, the efficiency must be between the values 0 and 1, which means that if the efficiency is equal to 1 is that every viewed pattern was selected, maximizing the efficiency. Yskout's efficiency metric could be reused to evaluate other proposals for selecting patterns. Paper ID28 is the only paper that report this kind of success, as the rest of the selected papers (independently to which facet they belong to) do not talk about reducing the time in design phase, neither enhancing the efficiency when choosing the right set of patterns.

Authors in paper ID40 (selection facet), looked for reducing the total number of security patterns, as the clusters can give additional insight into the similarity between patterns. This could lead to form a core set of patterns, which facilitates designers to select the most adequate pattern for each situation.

Affirmative answers of the selected papers about the sucess on applicability were answered by the same researchers that carried out the study, so it could be not reliable enough as they evaluate their own tools. Possible means to tackle this problem could be looking for third-party validations in other papers. This could help us proving whether the applicability of those mechanism was indeed successful or not in a more objective and theoretical way.

6.3.4.- *RQ 4.* What are the challenges for those identified mechanism?

Not all the selected papers have the same challenges and some of them did not report any. From those which include challenges, some are trying to deal with grade of knowledge the designer need to have to apply the reported mechanisms while others are trying to reach a validation or evaluation research level. Different challenges considered by the selected papers are described in this subsection.

The main challenge of the structure facet is trying to validate or evaluate their solution proposals as they want them to become empirical solutions.

Reaching an empirical solution was also a matter of concern in the selection facet, as paper ID26 which consists on a study about a process called SCRIP, first proposed it as a proposal solution and then they validate it. Furthermore, this SCRIP process presented another challenge which is providing a complete development environment based on the process itself, and they get it, by developing SCRIsTUDIO (paper ID14). SCRIsTUDIO is a tool based on Eclipse environment in order to apply SCRIP process automatically to ease the applicability of security patterns.

The selection facet (papers ID1 and ID40) is also trying to overcome the amount of knowledge needed when using the reported mechanisms. It is known that security patterns are defined by experts, but not all the designers are experts in the field of security. This means that perhaps these designers cannot easily understand what to do when selecting a pattern, and a series of advices or tips should be handed to them in order to ease their decision.

Additionally to future challenges, it is worth mentioning that right now some proposals have limitations (table 6.7). They are important to know before trying to take them to other domains. These limitations show that there is still work to do in the scope of security patterns' applicability, as some important issues are pending to be solved. Paper ID15 reports that the proposed taxonomy cannot deal with unknown threats. This means that designers should consider alternative proposes to face those unknown threats.

Limitations when applying the mechanism		
Selection facet	ID1	Valid if the application of security pattern prevents an specific attack.
	ID40	Cluster interpretability requires domain knowledge.
Structure facet	ID15	Cannot deal with unknown threats.
	ID38	Security patterns are sometimes documented in different ways.
Holistic facet	ID14	1. The step to the integration of patterns requires manual contribution. 2. SCRIsTUDIO needs to be completed and validated. 3. Process based on direct engineering methods. 4. Only confidentiality is guaranteed.
	ID41	The methodology is part of global MDE process for securing systems
	ID46	Not all security patterns are considered.

Table 6.7.- Mechanisms' limitations

6.4.- Further analysis

There were some extracted data that did not help to answer the defined research questions. Nevertheless, these criteria provide two relevant results that are explained in this section.

6.4.1.- Exclusive security feature

The main objective of this criterion was to identify if there was an exclusive feature for security domain. What we were looking for was any characteristic that is considered

by a mechanism, but only applicable for the security domain, and not able to be generalized by other bigger ones. After analyzing all facet papers, we have seen that three of them (ID10, ID40 and ID155) reported patterns with features applicable only to security domain. In general, these three papers try to guarantee Confidentiality, Integrity and Availability (CIA) triad.

In the selection facet, paper ID10 presents five main blocks that should be considered in SoSPa: authentication, authorization, cryptography, auditing and monitoring; paper ID40 classifies patterns according the security goals they are more according to resolve.

In the holistic facet only paper ID155 incorporated specific security attributes via security solution frames. There were eight security solution frames proposed in this paper:

1. Authorization: encapsulates security patterns allowing for custom conceptual authorization models to be built and realized using different enforcement architectures.

2. Identity management: encapsulates patterns concerned with managing (assigning, establishing, validating) identities of users and/or processes in a distributed system.

3. Secure communication: encapsulates security patterns for enabling two or more parties to communicate securely over a message channel.

4. Filtering: encapsulates patterns for network- and application-level filtering of information, including firewalls and custom data filter.

5. Storage security: encapsulates patterns for secure storage of information, including patterns for storing passwords, information dispersal, database security and others.

6. Logging and monitoring: encapsulates patterns for logging and monitoring events in a distributed system, ranging from creating simple log files to the deployment of intrusion-detection systems (network monitoring).

7. Execution control: encapsulates patterns for controlling and managing the execution of processes or, more generally, execution abstractions in a distributed system, including for process serialization/concurrency management, safe handling of mobile code, process isolation, improved resource availability and others.

8. Security information management: encapsulates patterns concerned with the management of security information such as policies, credentials, cryptographic keys etc.

6.4.2.- Design activity

Table 6.8 only depicts architectural and component design, as no author reported that his mechanism was focused neither on interface nor on database design. It shows that considering the total papers, most studies are more focused on the architectural design rather than component design. This means that authors are focusing on the global structure of the system, their units and relationships and how they are distributed [43], as the 70.59 % out of the total of papers were related to architectural design. Only the 23.53 % of the papers refer to component design, that is focused on the operation of each individual component. The remaining 5.88 % out of the total of papers, did not mention to which design activity they were referring to.

The results considering selection and structure facets follow the same tendency, as the studies are more focused on the architectural design. Nevertheless, in holistic facet papers, there are the same number of studies focused on the architectural design as on the component design.

	Architectural Design	**Component Design**	**Not mentioned**
Total papers (17)	12 (70.59 %)	4 (23.53 %)	1 (5.88 %)
Selection (6)	5 (83.33 %)	1 (16.67 %)	0 (0.00 %)
Structure (7)	5 (71.44 %)	1 (14.28 %)	1 (14.28 %)
Holistic (4)	2 (50.00 %)	2 (50.00 %)	0 (0.00 %)

Table 6.8.- Design activity analysis

6.5.- Comparison with previous studies

Most relevant previous related studies included Ortiz et al. in [25] and Sametinger et al. in [29]. Their studies were about the applicability of security patterns, but they were developed 8 and 9 years ago, so our thoughts were that their content could be outdated. Further, none of them were a systematic study.

Ortiz reported a classification and Sametinger a taxonomy. Both methodology elements can be considered as mechanisms that help designers to select security patterns while designing information systems. Considering the selection facet for making a fair comparison, it can be said that the number of mechanisms used has increased. In the proposal of Ortiz, a classification was proposed, while Sametinger proposed a new taxonomy; nowadays, authors have added methodologies to their contributions and also some of the reported mechanisms included the description of the steps that should be followed in order to select the right set of patterns in a guided way. Ortiz's study was a solution proposal whilst Sametinger's reached the validation research level. However, related approaches to the Ortiz's one have improved to a validation level. Currently, some classifications, such as the ones reported in papers ID32 and ID40, have been validated or even evaluated, as paper ID10. Hence, the selection facet is nowadays the one which has the highest number of empirical papers, leading us to think that this facet is the more mature one. Further, Ortiz and Sametinger claimed that the

taxonomies and classifications already proposed were not adequate for non-experts in order to select security patterns in specific situations. This problem still remains in the present, as after analyzing all selection-facet papers, we realized that papers ID1 and ID40 are also trying to overcome the amount of knowledge needed for non-experts when applying the reported mechanisms in those papers.

Ortiz and Sametinger also asserted that security patterns were not enough documented to let non-experts understand and apply them. This assertion can be related to our structure facet, but neither Ortirz nor Sametinger proposed any solution to that problem. Nowadays, structure facet reports taxonomies and classifications that help to normalize the way security patterns are defined. These mechanisms that belong to the structure facet in appendix 7.2, reported extensions or re-definitions in the sections (context, problem, solution, forces, etc.) of security patterns.

Finally, maybe our approach of considering the three facets to apply security patterns while designing information systems could help to overcome the problems that have been appearing for years. Considering the selection facet, designers could know which set of security patterns is the most suitable for designing a specific system. Further, taking into account the structure facet, it will help to normalize the way security patterns are defined and their structure. And finally, considering the holistic facet, designers know how to apply and use each pattern depending on the system they want to design. It is important to mention, that the number of facets could be increased depending on the approaches taken by authors regarding applicability.

7

Conclusions

This section details the conclusions that have been reached after the development of this study. Furthermore, some lines of future work are proposed.

7.1.- Conclusions

The main aim of this study was analyzing the mechanisms proposed to facilitate the applicability of security patterns. Hence, a systematic literature review was undertaken. The papers that were considered for developing the review were those reporting security patterns' applicability mechanisms found in Scopus.

It was known that the concept of applicability implies something that was susceptible to be applied. After having read the 17 selected papers, we have found that applicability of security patterns is not a simple process which do not only refer how to use patterns, but it may also embrace the selection of them and a good structure definition.

Due to those different approaches, we defined three different facets: selection, structure and holistic. Selection facet refers to the process of selecting the right pattern set that could be applied to design an information system and the methodology elements that best fit this requirements are methodologies and taxonomies. Structure facet was defined due to all authors who wrote papers related to this facet claimed that there should be a normalized way of defining security patterns including extensions and updates of patterns' structure. The most use mechanisms within this facet were taxonomies and classifications. Finally, the holistic facet represents the whole process of selecting and applying security patterns and the authors within this facet mostly reported guided methodologies that helped designer applying the right set of security patterns. Although a tool supports the application of a methodology, only one author (paper ID14) reported such a methodology element.

Our initial beliefs were that most papers could report empirical results, such as validation or evaluation researches, rather than non-empirical ones due to their intrinsic relationship with applicability. Despite this, 52.9 % of the selected papers were non empirical studies. So it can be concluded that there is still much work to do to implement the reported mechanisms into practice, so they can become an empirical study. This could be done by developing the reported solutions and implement them in a laboratory context (so validation research level could be reached) or in an industry context (in order to get evaluation research level), where the consequences, benefits and drawbacks about the applicability of security patterns can be materialized in the real world.

The mechanisms that we were looking for at the beginning of the study, were mapped to methodology elements defined in software engineering method metamodels such as SEMDM in [36]: classifications, taxonomies, processes, tools or work flows. These methodology elements could be seen as static or dynamic oriented. Selection and structure facets reported the majority of static elements, such as their description do not imply a *sense of movement*. Instead of that, those elements aim to define structure for normalizing syntax, vocabulary, sections or categories to which a pattern could belong. Holistic facet reported most of dynamic method elements, such as processes or tools for helping in carrying out them. These methodologies involve the *action* of selecting and applying the right security patterns when designing an information system.

An interesting discovery was that some papers reported an exclusive security feature that only security patterns considered. Papers from the selection (ID10 and ID40) and holistic (ID155) facets tried to guarantee the Confidentiality, Integrity and Availability (CIA) triad, as they are considered the three most crucial components of security. Moreover, paper ID10 also deals with other security concepts such as cryptography, auditing and monitoring; and paper ID155 implements frames to control the logging or filtering. The way those exclusive features were addressed by the different mechanisms could be replied for other domain patterns' application.

Contrary to what could be thought, the main benefit when using the reported mechanism in paper ID28 is not an improvement of the employed time while designing information systems, but an improvement of the efficiency when selecting the right set of patterns. This efficiency is also defined in paper ID28.

It is important to mention that there are some limitations (papers ID1, ID14, ID15, ID38, ID40, ID41 and ID46) of the mechanisms which have to be taken into account for implementing them. One of the main limitations for papers ID1 and ID40 is the grade of knowledge that designers must have to apply the reported mechanism, as security patterns are defined by experts. However, not all the designers are experts in the field of security, so they can not easily understand how the experts define those patterns. So, it is important that any designer, expert or not, should be able to apply security patterns, no matter their knowledge in security patterns. Usually, when it is wanted to hide complexity away from developers, it is a good practice to encapsulate it as a framework or library, so that the designers do not know the details but just apply the pattern.

Further, limitations could be seen as future work together with challenges reported in different papers. Main challenges are to reach a validation or evaluation research level of those papers that are solution proposals.

7.2.- Future work

Lack of time prevent us from developing a snowballing search in order to complement this systematic study. This could add significant value to the study.

In general terms, affirmative answers to the question regarding the success when applying the reported mechanism were answered by the same authors of the papers.

To really prove the success of those mechanisms in a more objective way there are two different options. The first one, is developing a theoretical search to look for papers where those mechanisms were cited, and prove whether the application of them is successful or not. The second option, is that the authors of the paper where the mechanisms were reported, should have developed more cases of study, in order to prove empirical success.

It is also necessary to make further research for taking some of these mechanisms, or considerations inside them, to the privacy engineering field.

References

[1] Ann Cavoukian (2011). Privacy by Design. The 7 Foundational Principles. Implementation and Mapping of Fair Information Practices. https://www.iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf

[2] Colesky, M., Caiza, J.C., Alamo, J.M. Del, Hoepman, J.-H. and Martin, Y.-S. (2018). A System of Privacy Patterns for User Control http://repository.ubn.ru.nl/handle/2066/191709

[3] Ito, Y., Washizaki, H., Yoshizawa, M., Fukazawa, Y., Okubo, T., Kaiya, H., Hazeyama, A., Yoshioka, N. and Fernandez, E.B. (2015). Systematic Mapping of Security Patterns Research. Proceeding PLoP '15 Proceedings of the 22nd Conference on Pattern Languages of Programs (2015), 3–4.

[4] Schumacher, M. and Roedig, U. (2001). Security Engineering with Patterns.

[5] Uzunov, A. V., Fernandez, E.B. and Falkner, K. (2012). Securing distributed systems using patterns: A survey. Computers and Security. 31, 5 (Jul. 2012), 681–703. https://doi.org/10.1016/J.COSE.2012.04.005

[6] Fischer-Hübner, S., Köffel, C., Pettersson, J.-S., Wolkerstorfer, P., Graf, C., Holtz, L.E., König, U., Hedbom, H. and Kellermann, B. (2010). HCI Pattern Collection – Version 2. (2010).

[7] SPEM 2.0. http://www.omg.org/spec/SPEM/2.0/.

[8] Drozd, O. (2016). Privacy pattern catalogue: A tool for integrating privacy principles of ISO/IEC 29100 into the software development process. IFIP Advances in Information and Communication Technology. 476, (2016), 129–140. https://doi.org/10.1007/978-3-319-41763-9_9

[9] Privacy Patterns: (2011). https://privacypatterns.org/. Accessed: 2018-01-18.

[10] privacypatterns.eu - Collecting patterns for better privacy: 2015. https://privacypatterns.eu/. Accessed: 2018-01-18.

[11] Romanosky, S., Acquisti, A., Hong, J., Cranor, L.F. and Friedman, B. (2006). Privacy patterns for online interactions. Proceedings of the 2006 conference on Pattern languages of programs - PLoP '06 (New York, New York, USA, 2006), 1.

[12] Caiza, J.C., Martín, Y.-S., Del Alamo, J.M. and Guamán, D.S. (2017). Organizing Design Patterns for Privacy. Proceedings of the 22nd European Conference on Pattern Languages of Programs - EuroPLoP '17 (New York, New York, USA, 2017), 1–11.

[13] IEEE Xplore. 610-1990 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. https://ieeexplore.ieee.org/document/182763/

[14] Tutorials point. SDLC Overview. http://www.tutorialspoint.com/sdlc/sdlc_overview.htm. Accessed: 2018-01-23.

[15] C. Alexander, S. Ishikawa, and M. Silverstein. A Pattern Language: Towns, Buildings, Constructions. Oxford University Press, 1977. http://books.google.com/books?id=hwAHmktpk5IC

[16] E. Gamma, J. Vlissides, R. Johnson and R. Helm (1994) Design Patterns. Elements of Reusable Object-Oriented Software.

[17] The Hillside Group. Pattern Language of Programs. http://www.hillside.net/plop/2018/. Accessed: 2018-06-08

[18] EuroPLoP. European Pattern Language of Programs. http://europlop.net. Accessed: 2018-06-08

[19] EuroPLoP. European Pattern Language of Programs. http://www.europlop.net/content/introduction

[20] F. Bushmann, R. Meunier, and H. Rohnert. (1996). Pattern-oriented software architecture: A system of patterns.

[21] Kitchenham, Barbara and Charters, Stuart (2007) Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. https://doi.org/10.1145/1134285.1134500

[22] Jörg Lenhard, Lothar Fritsch, and Sebastian Herold (2017) A Literature Study on Privacy Patterns Research.

[23] Petticrew, Mark and Helen Roberts (2005) Systematic Reviews in the Social Sciences: A Practical Guide Blackwell Publishing, 2005.

[24] Ponde, P., Shirwaikar, S., and Kreiner, C. (2016) An Analytical Study of Security Design Patterns. Proceedings of the 21st European Conference on Pattern Languages of Programs, 1–22 https://doi.org/10.1145/3011784.3011821

[25] Ortiz, R., Moral-García, S., Moral-Rubio, S., Vela, B., Garzás, J., and Fernández-Medina, E. (2010) Applicability of security patterns. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6426 LNCS) pages 672-684. https://doi.org/10.1007/978-3-642-16934-2_49

[26] Uzunov, A. V., Fernandez, E. B., and Falkner, K. (2015). ASE: A comprehensive pattern-driven security methodology for distributed systems. Computer Standards and Interfaces, 41, 112–137. https://doi.org/10.1016/j.csi.2015.02.011.

[27] Uzunov, A. V., Fernandez, E. B., and Falkner, K. (2012) Securing distributed systems using patterns: A survey. Computers and Security, 31(5), 681–703. https://doi.org/10.1016/j.cose.2012.04.005

[28] Bunke, M. (2015) Software-security patterns. Proceedings of the 20th European Conference on Pattern Languages of Programs - EuroPLoP '15, 1–17. https://doi.org/10.1145/2855321.2855364

[29] Sametinger, J., Wiesauer, A., and Sametinger, J. (2009). A Security Design Pattern Taxonomy Based on Attack Patterns : Findings of a Systematic Literature Review. https://doi.org/10.13140/RG.2.1.4615.7202.

[30] Badia, G. (2015) Multiple databases are needed to search the journal literature on computer science. Evidence Based Library and Information Practice, 10(4), 241–243.. https://doi.org/10.1007/s11192-014-1506-1

[31] Inayat, I., Salim, S. S., Marczak, S., Daneva, M., and Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. Computers in Human Behavior, 51, 915–929. https://doi.org/10.1016/j.chb.2014.10.046

[32] Achimugu, P., Selamat, A., Ibrahim, R., and Mahrin, M. N. R. (2014). A systematic literature review of software requirements prioritization research. Information and Software Technology, 56(6), 568–585. https://doi.org/10.1016/j.infsof.2014.02.001

[33] Wohlin, C., Runeson, P., Anselmo, P., Mota, D., Neto, S., Engström, E., ... Santana De Almeida, E. (2013). On the reliability of mapping studies in software engineering. The Journal of Systems and Software, 86, 2594–2610. https://doi.org/10.1016/j.jss.2013.04.076

[34] Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (n.d.). Systematic Mapping Studies in Software Engineering. http://www.robertfeldt.net/publications/petersen_ease08_sysmap_studies_in_se.pdf

[35] Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. https://doi.org/10.1016/j.infsof.2015.03.007

[36] International Organization for Standardization. ISO/IEC 24744:2007 Software Engineering - Metamodel for Development Methodologies

[37] Li, T., and Mylopoulos, J. (2014). Modeling and applying security patterns using contextual goal models. CEUR Workshop Proceedings, 1157. https://doi.org/10.13140/2.1.2756.7361

[38] Dingsoyr, T., Hanssen, G. K., Dyba, T., Anker, G., and Nygaard, J. O. (2006). Towards a better integratioin of patterns in secure component-based systems design 607-621. https://doi.org/10.1007/978-3-642-29066-4{_}11.

[39] Nguyen, P. H., Yskout, K., Heyman, T., Klein, J., Scandariato, R., and Le Traon, Y. (2015). SoSPa: A system of Security design Patterns for systematically engineering secure systems. https://doi.org/10.1109/MODELS.2015.7338255.

[40] Bouaziz R., Kammoun S. (2015) A decision support map for security patterns application, 750-759. https://link.springer.com/chapter/10.1007/978-3-319-21410-8_57

[41] Bouaziz, R. (2016). SCRISTUDIO : A Security Pattern Integration Tool. http://ieeexplore.ieee.org/document/7479264/.

[42] Fernandez, E. B., Washizaki, H., Yoshioka, N., Kubo, A., and Fukazawa, Y. (2008). Classifying security patterns. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 4976 LNCS). https://doi.org/10.1007/978-3-540-78849-2_35

[43] Sommerville, I. (2011). Software Engineering. Software Engineering. https://doi.org/10.1111/j.1365-2362.2005.01463.x.

[44] Elsevier Scopus. https://www.elsevier.com/solutions/scopus.

[45] R. Scandariato, K. Yskout, T. Heyman, and W. Joosen (2008). Architecting software with security patterns

[46] Viorela Dan (2017) Empirical and Non-Empirical Methods.

Appendix

A. Selected Papers' Identifiers

- **ID1.** Sametinger, J., Wiesauer (2009). A Security Design Pattern Taxonomy Based on Attack Patterns : Findings of a Systematic Literature Review. https://doi.org/10.13140/RG.2.1.4615.7202
- **ID10.** Nguyen, P. H., Yskout, K., Heyman, T., Klein, J., Scandariato, R., and Le Traon, Y. (2015). SoSPa: A system of Security design Patterns for systematically engineering secure systems. 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, MODELS 2015 - Proceedings, 246–255. https://doi.org/10.1109/MODELS.2015.7338255
- **ID14.** Bouaziz, R. (2016). SCRISTUDIO : A Security Pattern Integration Tool. http://ieeexplore.ieee.org/document/7479264/
- **ID15.** Moral-García, S., Moral-Rubio, S., Rosado, D.G., Fernández, E.B., Fernández-Medina, E. (2014) Enterprise security pattern: A new type of security pattern pp. 1670-1690 https://doi.org/10.1002/sec
- **ID26.** Bouaziz, R., Kallel, S., and Coulette, B. (2013). An engineering process for security patterns application in component based models. 2013 IEEE 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2013, 231–236. https://doi.org/10.1109/WETICE.2013.27
- **ID28.** Yskout, K., Scandariato, R., and Joosen, W. (2012). Does organizing security patterns focus architectural choices? Proceedings - International Conference on Software Engineering, 617–627. https://doi.org/10.1109/ICSE.2012.6227155
- **ID30.** Moral-García, S., Moral-Rubio, S., Fernández, E. B., and Fernández-Medina, E. (2014). Enterprise security pattern: A model-driven architecture instance. Computer Standards and Interfaces, 36(4), 748–758. https://doi.org/10.1016/j.csi.2013.12.009
- **ID32.** Bouaziz R., Kammoun S. (2015) A decision support map for security patterns application, 750-759. https://link.springer.com/chapter/10.1007/978-3-319-21410-8_57

- **ID38.** Li, T., and Mylopoulos, J. (2014). Modeling and applying security patterns using contextual goal models. CEUR Workshop Proceedings, 1157. https://doi.org/10.13140/2.1.2756.7361
- **ID40.** Ponde, P., Shirwaikar, S., and Kreiner, C. (2016). An Analytical Study of Security Design Patterns. Proceedings of the 21st European Conference on Pattern Languages of Programs, 1–22. https://doi.org/10.1145/3011784.3011821
- **ID41.** Motii, A., Hamid, B., Lanusse, A., and Bruel, J. M. (2016). Towards the integration of security patterns in UML Component-based Applications. CEUR Workshop Proceedings, 1693, 2–6. http://ceur-ws.org/Vol-1693/PamePaper1.pdf
- **ID46.** Bouaziz R., Kallel S., Coulette B. (2014). An approach for security patterns application in component based models https://link.springer.com/chapter/10.1007/978-3-319-09156-3_21
- **ID53.** Bouaziz R., Hamid B., Desnos N. Towards a better integratioin of patterns in secure component-based systems design 607-621.
https://link.springer.com/chapter/10.1007/978-3-642-21934-4_49
- **ID56.** Castellanos C., Vergnaud T., Borde E., Derive T., Pautet L. Formalization of design patterns for security anddependability https://www.scopus.com/inward/record.uri?eid=2-s2.0-84880056635&doi=10.1145%2f2465470.2465476&partnerID=40&md5=9256e82fb3660bf21eeaf3dd2d896623
- **ID58.** Fernández, E. B. (2006). Security Patterns and Secure Systems Design, (June), 1–12. http://www.laccei.org/LACCEI2006-PuertoRico/Papers%20-pdf/IT048_Fernandez.pdf
- **ID60.** Fernandez, E. B., Washizaki, H., Yoshioka, N., Kubo, A., and Fukazawa, Y. (2008). Classifying security patterns. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 4976 LNCS). https://doi.org/10.1007/978-3-540-78849-2_35
- **ID72.** Heyman, T., Scandariato, R., and Joosen, W. (2012). Reusable formal models for secure software architectures. Proceedings of the 2012 Joint Working Conference on Software Architecture and 6th European Conference on Software

Architecture, WICSA/ECSA 2012, 41–50. https://doi.org/10.1109/WICSA-ECSA.212.12

- **ID155.** Uzunov, A. V., Fernandez, E. B., and Falkner, K. (2015). ASE: A comprehensive pattern-driven security methodology for distributed systems. Computer Standards and Interfaces, 41, 112–137. https://doi.org/10.1016/j.csi.2015.02.011

B. Selected papers' facets

- Selection papers.

  - **ID1.** Sametinger, J., Wiesauer (2009). A Security Design Pattern Taxonomy Based on Attack Patterns : Findings of a Systematic Literature Review. https://doi.org/10.13140/RG.2.1.4615.7202
  - **ID10.** Nguyen, P. H., Yskout, K., Heyman, T., Klein, J., Scandariato, R., and Le Traon, Y. (2015). SoSPa: A system of Security design Patterns for systematically engineering secure systems. 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, MODELS 2015 - Proceedings, 246–255. https://doi.org/10.1109/MODELS.2015.7338255
  - **ID26.** Bouaziz, R., Kallel, S., and Coulette, B. (2013). An engineering process for security patterns application in component based models. 2013 IEEE 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2013, 231–236. https://doi.org/10.1109/WETICE.2013.27
  - **ID28.** Yskout, K., Scandariato, R., and Joosen, W. (2012). Does organizing security patterns focus architectural choices? Proceedings - International Conference on Software Engineering, 617–627. https://doi.org/10.1109/ICSE.2012.6227155
  - **ID32.** Bouaziz R., Kammoun S. (2015) A decision support map for security patterns application, 750-759. https://link.springer.com/chapter/10.1007/978-3-319-21410-8_57
  - **ID40.** Ponde, P., Shirwaikar, S., and Kreiner, C. (2016). An Analytical Study of Security Design Patterns. Proceedings of the 21st European Conference on Pattern Languages of Programs, 1–22. https://doi.org/10.1145/3011784.3011821

- Structure papers:

  - **ID15.** Moral-García, S., Moral-Rubio, S., Rosado, D.G., Fernández, E.B., Fernández-Medina, E. (2014) Enterprise

security pattern: A new type of security pattern pp. 1670-1690 https://doi.org/10.1002/sec

- **ID30.** Moral-García, S., Moral-Rubio, S., Fernández, E. B., and Fernández-Medina, E. (2014). Enterprise security pattern: A model-driven architecture instance. Computer Standards and Interfaces, 36(4), 748–758. https://doi.org/10.1016/j.csi.2013.12.009
- **ID38.** Li, T., and Mylopoulos, J. (2014). Modeling and applying security patterns using contextual goal models. CEUR Workshop Proceedings, 1157. https://doi.org/10.13140/2.1.2756.7361
- **ID53.** Bouaziz R., Hamid B., Desnos N. Towards a better integratioin of patterns in secure component-based systems design 607-621. https://link.springer.com/chapter/10.1007/978-3-642-21934-4_49
- **ID58.** Fernández, E. B. (2006). Security Patterns and Secure Systems Design, (June), 1–12. http://www.laccei.org/LACCEI2006-PuertoRico/Papers%20-pdf/IT048_Fernandez.pdf
- **ID60.** Fernandez, E. B., Washizaki, H., Yoshioka, N., Kubo, A., and Fukazawa, Y. (2008). Classifying security patterns. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 4976 LNCS). https://doi.org/10.1007/978-3-540-78849-2_35
- **ID72.** Heyman, T., Scandariato, R., and Joosen, W. (2012). Reusable formal models for secure software architectures. Proceedings of the 2012 Joint Working Conference on Software Architecture and 6th European Conference on Software Architecture, WICSA/ECSA 2012, 41–50. https://doi.org/10.1109/WICSA-ECSA.212.12

- Holistic papers:

  - **ID14.** Bouaziz, R. (2016). SCRISTUDIO : A Security Pattern Integration Tool. http://ieeexplore.ieee.org/document/7479264/
  - **ID41.** Motii, A., Hamid, B., Lanusse, A., and Bruel, J. M. (2016). Towards the integration of security patterns in UML

Component-based Applications. CEUR Workshop Proceedings, 1693, 2–6. http://ceur-ws.org/Vol-1693/PamePaper1.pdf

- **ID46.** Bouaziz R., Kallel S., Coulette B. (2014). An approach for security patterns application in component based models https://link.springer.com/chapter/10.1007/978-3-319-09156-3_21
- **ID155.** Uzunov, A. V., Fernandez, E. B., and Falkner, K. (2015). ASE: A comprehensive pattern-driven security methodology for distributed systems. Computer Standards and Interfaces, 41, 112–137. https://doi.org/10.1016/j.csi.2015.02.011