Security and Privacy under a Unified Framework: A Review

Argyri Pattakou and Christos Kalloniatis

Privacy Engineering and Social
Informatics Laboratory,
Department of Cultural
Technology and Communication,
University of the Aegean
University Hill
GR 81100 Mytilene, Greece
Email: a.pattakou@aegean.gr, chkallon@aegean.gr

Stefanos Gritzalis

Information and Communication Systems
Security Laboratory,
Department of Information
and Communications Systems Engineering,
University of the Aegean
GR 83200 Samos
Greece
Email: sgritz@aegean.gr

***Abstract*—As the software industry experiences a rapid growth in developing information systems, many methodologies, technologies and tools are continuously developing in order to support the system implementation process. However, as security and privacy have been considered important aspects of an information system, many researchers presented methods that, through a number of specific steps, enable system designers to integrate security and privacy requirements at the early stage of system design. Different security and privacy engineering methods have been presented in order to be applied in traditional or cloud architectures. This paper reviews a number of security and privacy requirements engineering methods in both areas and presents a comparative study between these methods. Additionally, as at the recent years, security and privacy tend to be considered as two different but interdependent concepts, we present a conceptual model that considers both security and privacy under the same unified framework.**

Keywords–security; privacy; requirements engineering methods; traditional architecture; cloud computing; unified framework.

I. Introduction

For many decades, as the software industry has been constantly growing, the main interest of software engineers was to deliver new software releases rapidly, with no bugs and with the appropriate functionality. Under these circumstances, new tools, methodologies and technologies have been introduced in order to support system analysis and design, as well as software implementation. However, in the last decade, the software engineers community has realized that security and privacy are very important aspects in software engineering and, as a result, all the development software systems have to ensure security and privacy of the stored data [1-7].

As the interest of software engineers was mainly in developing new software, security and privacy was considered during implementation stage more as an ad-hoc process rather than an integrated process at the system design level. However, each late detection of possible security or privacy vulnerabilities has been proven to be extremely costly and time-consuming. Indeed, many researchers argue that security and privacy requirements have to be considered at the system analysis and design stage as security and privacy constraints might affect software functional requirements [8]. In this direction, we need mechanisms in order to elicit and analyse security and privacy requirements through a number of well-defined steps.

However, as the software industry was faced with a lack of integrated security and privacy requirements engineering methods, many researchers focused on introducing methods that support the elicitation of security and privacy requirements during the system design process. A requirement engineering method in the area of security and privacy can support engineers to define critical assets and the threats against them, to identify with accuracy security or/and privacy goals and to examine any kind of conflicts between them in order to come up with a clear and resistant set of security or/and privacy requirements.

Security and privacy requirements engineering methods have been built based on different approaches because, for each method, security and privacy requirements can be derived from different processes. For instance, some methods were introduced as goal-oriented methodologies as security and privacy goals might affect functional goals while other methods put as central issue potential risks and threats in order for security and privacy requirements to be derived. Different approaches can cover possible limitations or gaps among methods, as well as provide a variety of options to system analysts and designers in order to select the method that best fits the system into consideration.

During the last decade, literature has presented a number of security and privacy requirements engineering methods that support system designers and developers to implement secure and privacy-aware information systems hosted in traditional architectures. Some methods consider security or privacy requirements separately, but some other methods consider privacy as a subset of security. Recent literature efforts [7],[9-10] emphasize the need for parallel examination of security and privacy requirements under the same unified framework, as a possible security breach might affect users privacy and vice versa. However, few steps have been taken in this direction [11].

On the other hand, as cloud computing architecture introduces special characteristics, security and privacy requirements methods have to be developed in order to cover these special needs [12-14]. However, as the cloud computing area still suffers from a lack of integrated requirements engineering methods, methods that were initially introduced for traditional

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

architecture systems were extended in order to be applied in cloud systems as well [15]. But, at the moment, as far as we know, a method for cloud architecture that supports the parallel examination of security and privacy concepts has not been introduced.

Generally, literature presents a large number of papers that review security or privacy requirements engineering methodologies in traditional or cloud architectures. However, non of these reviews conduct a comparative study among them in order to support designers to select the most appropriate method for their system into consideration. Additionally, none of these studies justifies and analyses the need for a unified approach between security and privacy in the requirements engineering area.

In this paper, we present a number of security and privacy requirements methods that have been introduced in the last decades in order to support system design and analysis in traditional or cloud architectures. Also, we present a comparative study among methods that demonstrates the need for designing a framework that will consider security and privacy together under a holistic unified approach. At the end, a conceptual model for cloud-based systems that considers in parallel security and privacy requirements is presented.

Section II presents a set of security and privacy requirements engineering methods for traditional architectures and a comparative study among them. Section III is referring to security and privacy requirements engineering methods that can be applied in a cloud environment. Section IV presents the proposed conceptual model applied for cloud-based systems and Section V concludes the paper.

II. Security and Privacy Requirements Engineering Methods in Traditional Architectures

In this section, we present a number of security and privacy requirements engineering methods for traditional architecture that was introduced in the literature, in order to map the area and to conduct a comparative study between them.

A. Security and Privacy Requirements Engineering methods

1) Security Quality Requirements Engineering (SQUARE) Methodology: SQUARE methodology [16] was introduced because the software industry was missing an integrated model for eliciting and analyzing security requirements. The proposed methodology is a risk-driven method that supports the elicitation, categorization, prioritization and inspection of the security requirements through a number of specific steps. SQUARE also supports the performance of risk assessment in order to verify the tolerance of the system against possible threats. The final output of this method is a document that includes all the necessary security requirements that are essential in order for the security goals of the system to be satisfied. The methodology introduces the terms of security goal, threat and risk but does not take into consideration the assets and the vulnerabilities of the system. The application of SQUARE methodology requires the participation and the cooperation between stakeholders and the requirements engineering team in order to identify with accuracy all the necessary security requirements at the early stage of the development process. SQUARE does not refer to the elicitation of privacy requirements.

2) Model Oriented Security Requirements Engineering (MOSRE): As many research efforts conclude that considering non-functional requirements after system design can be proved very costly, Salini and Kanmani introduced a security requirements engineering framework (MOSRE) [17] for Web applications that considers security requirements at the early stages of the development process. The framework covers all phases of requirements engineering and suggests the specification of the security requirements alongside with the specification of system requirements. The authors suggest the identification of the objectives, stakeholders and assets of the Web application during the inception phase. The elicitation phase includes the identification of non-security goals and requirements in parallel with security goals, the identification-categorization-prioritization of threats and system vulnerabilities and a risk assessment process in order to elicit the final security requirements. Next phases include the analysis and modeling, the categorization-prioritization and the validation of the final security requirements. The framework does not support the elicitation of privacy requirements.

3) Security Requirements Engineering Framework (SREF): Haley et al. [18] introduced a problem based approach in order to elicit and analyze security requirements. The authors describe an iterative process of four steps. During these steps, security goals can be identified after the identification of functional (business) requirements. The identification of security goals includes the identification of system assets and a threat analysis. Risk assessment is also supported during the identification of security goals. However, in order to elicit security requirements from these security goals, the authors of Security Requirements Engineering Framework (SREF) take security requirements as constraints for functional requirements of the system under consideration and these constraints satisfy one or more security goals. The authors also encourage the use of problem diagrams to capture functional requirements with such constraints. The framework includes the notion of trust assumptions and the construction of satisfaction arguments by system analysts in order to validate security requirements. Privacy requirements are not considered by this framework.

4) Eliciting Security Requirements from the Business Process Models : Ahmed and Matulevicius introduced an asset based approach in order to elicit security goals from business process models and translate them into security requirements [19]. The method consists of two stages. At the first stage, an early analysis is performed in order to determine business assets that must be protected against security risks and security goals. At the second stage, the elicitation of security requirements is performed during examination of the security risk of business assets in five contextual areas: access control, communication channel, input interfaces, business services and data store. The final result is the elicitation of security requirements and the generation of business rules that satisfy security goals of the system under consideration. This framework does not support categorization, prioritization and validation of security requirements.

5) Security Requirements Engineering process (SREP): Mellado et al. presented SREP method [20] in order to provide a unified framework that considers concepts from requirements engineering and security engineering as well. Security Requirements Engineering Process (SREP) is an iterative and incremental security requirements engineering process and is aiming to integrate security requirements at the early stages of software development life cycle [21]. SREP is an asset-based method, as well as a threat and risk driven method and it is based on the integration of Common Criteria [22] into the software life cycle in order to specify security requirements and validate that products meet security goals. The main idea of the proposed framework is that the unified process is divided into four phases: Inception, Elaboration, Construction and Transition. Each phase might include many iterations of nine activities (definitions, identification of assets, security objectives and threats, risk assessment, elicitation of security requirements, categorization-prioritization, inspection and repository improvement) but with different emphasis depending on what phase of the lifecycle the iteration is in [20]. Also, the authors propose the use of Security Resources Repositories to store sets of requirements that can be reused in different domains. Privacy requirements have not been considered by the authors.

6) Secure Tropos: Tropos methodology [23] was introduced by Castro et al. in order to cover system requirements during the whole software development process. However, Tropos methodology gives a strong focus on the early stage of system analysis. The framework includes five development phases: early requirements, late requirements, architectural design, detailed design and implementation. However, security concepts have not been considered in any of theses phases. Thus, Mouratidis et al. extended Tropos methodology in order to accommodate security concepts during the requirements analysis. The extension is called Secure Tropos [24] and utilizes only the early and late requirements phases of Tropos framework. Secure Tropos introduces the concept of security constraints. According to the authors, security constraints are a set of conditions, rules and restrictions that are imposed on a system and the system must operate in such way that none of them will be violated [24]. In the early requirements phase, a security diagram is constructed in order to represent the connection between security features, threats and mechanisms that help the satisfaction of security goals. The security diagram is taken into consideration at the late requirements phase in order for the designers to impose security constraints to the system-to-be. The enforcement of security constraints in different parts of the system can facilitate the disclosure of possible conflicts between requirements.

7) KAOS: In 2000, KAOS [25] was first introduced as a goal-oriented requirements engineering method in order to elaborate requirements from high level goals. According to the authors, the fulfillment of goals might be blocked by some exceptional agent behaviors that are called obstacles. In KAOS method, these obstacles have to be identified and resolved, through the elaboration of scenarios between software and agents, in order to produce a reliable system [26]. However, due to the fact that KAOS methodology considers only functional requirements, authors extended KAOS [27] in order to elaborate security requirements as well. The main idea of the extended framework is to build two models. A model of the system-to-be, that will describe the software and the relations between goals, agents, objects, operations and requirements and an anti-model that will capture possible attackers, their goals and system vulnerabilities in order to elicit all possible threats and security requirements to prevent such treats. Security requirements that derived by the anti-model as countermeasures have to be integrated in the original model.

8) PresSure: In 2014, Fabender et al. introduced a problem-based methodology, which is called presSure [28-29] in order to identify security needs during requirements analysis of software systems. The identification of security requirements is based on functional requirements of a system-to-be and on the early identification of possible threats. The methodology supports the modeling of functional requirements through problem diagrams. At next stage and after identifying the critical assets of the system and the rights of the authorized entities, possible attackers and their abilities have to be determined. Based on that information, a set of graphs is generated in order to visualize flows of possible threats related to the attackers access to critical assets. Security requirements derived from the analysis of these graphs. For each identified asset, every functional requirement is related with possible threats and security requirements.

9) LINDDUN: LINDDUN [30] was first introduced in 2010 by Deng et al. as a privacy threat analysis framework in order to support the elicitation and fulfillment of privacy requirements in software-based systems. According to the LINDDUN methodology, after designing a data flow diagram (DFD) of the system, privacy threats are related to the listed elements of the DFD. Threats in LINDDUN are categorized in seven types: Linkability, Identifiability, Non-repudiation, Detectability, Information Disclosure, Content Unawareness, Policy and consent Non-compliance. The method uses privacy threats trees and misuse cases in order to collect the threat scenarios of the system. Trough these misuse cases, privacy requirements can be extracted. Also, LINDDUN supports the prioritization and validation of privacy threat through the process of risk-assessment, before eliciting the final privacy requirements and before selecting the appropriate privacy-enhancing technologies. The authors of LINDDUN also map privacy-enhancing technologies to each privacy requirement in order to support system designers to select the appropriate techniques that satisfy privacy requirements.

10) SQUARE for privacy: As privacy plays an important role in software engineering, the authors of SQUARE methodology [16] adapted their approach in order to support the elicitation of privacy requirements at the early stages of software development process [31]. The extended framework includes the same steps as the original SQUARE method in conjunction with the Privacy Requirements Elicitation Technique (PRET) [32], a technique that supports the elicitation and prioritization of privacy requirements. This technique uses a database of privacy requirements based on privacy laws and regulations. However, the authors note that the database needs to be updated as the laws change and conclude that a new integrated tool is needed in order to support the elicitation of security and privacy requirements in parallel.

11) PriS: PriS [33] has been introduced by Kalloniatis et al. as a goal-oriented approach in order to integrate privacy requirements into the system design process. The main idea of this methodology is that privacy requirements are considered as organizational goals and adopts the use of privacy-process patterns in order to describe the impact of privacy goals to the affected organizational processes, to model the privacy-related organizational processes and to support the selection of the system architecture that best satisfies these processes. Thus, the authors of PriS cover the gap between system design and implementation phase. According to PriS, the identification of privacy goals is based on eight privacy concepts namely authentication, authorization, identification, data protection, anonymity, pseudonymity, unlinkability and unobservability.

12) Secure Tropos and PriS metamodel: According to the above methodologies, most of the approaches in requirements engineering tend to consider security and privacy separately or consider privacy as a subset of security. However, a number of research efforts [7], [9] support that security and privacy are two different concepts that have to be examined separately but under the same unified framework. Under these circumstances, Islam et al. [11] introduced a model-based process that considers security and privacy concepts in parallel at the early stage of system analysis. This process integrates two different engineering methods. Secure Tropos is used as the main method in order to identify and analyse security requirements of the system under consideration. However, as privacy concepts are not considered through this method, Secure Tropos is extended integrating the PriS solution. Thus, security and privacy requirements can be identified and analysed in order to meet the goals but also the appropriate architecture and implementation technique can be selected in order for privacy goals to be satisfied.

13) Goal-based requirements analysis method (GBRAM): Anton and Earp introduced the Goal-based requirements method (GBRAM) [38] in order to support system designers to design secure e-commerce systems via identifying system and enterprise goals and requirements. The ultimate goal of this approach is to ensure security and privacy requirements coverage during the early stage of system design level by supporting the specification of security and privacy requirements and policies and checking the compliance among them. Risk assessment is considered critical in order to ensure that security and privacy policies reflect the actual security and privacy system requirements. In this direction, GBRAM describes four activities in order to support the identification of security and privacy goals and their conversion into security requirements and security/privacy policies. These activities include: Goals identification, Goals elaboration, Goals refinement and goals operationalization. In the GBRAM, each goal is assessed for risks and potential impacts. During risk assessment a new goal or a sub-goal might be added or the existing goal can be adjusted in order to mitigate the risk. In the GBRAM, goals are, also, categorized in five classes: user, system, communication, knowledge and quality goals. Finally, an assess compliance activity is introduced in order to be ensured that system requirements that have been elicited are aligned with the enterprise's security and privacy policy.

14)Abuse frames: L. Lin et al. presented the Abuse Frames approach [39] in order to analyse and represent security threats and to derive security requirements at the early phases of system requirements level. The authors support that abuse frames can delimit security problems so that system analysts and designers can focus on the characteristics of problem domains in order to uncover more easily security vulnerabilities and threats and to select the most appropriate security measures. According to the authors, abuse frames provide an abstract model of threats imposed by a potential malicious user within a defined system boundary. The approach uses Jackson's Problem Frames [40] in order to define boundaries in the problem areas and to focus on early security threat analysis. Also, the authors introduce the meaning of anti-requirements that define a set of undesirable requirements imposed by malicious users in order to subvert the existing system requirements. Privacy requirements are not considered by this approach.

15) Misuse Cases: Use Cases [41] have been proven very helpful for the elicitation and documentation of functional requirements in system analysis phase. Use cases methodology uses UML diagrams and various templates for textual description, in order to capture the appropriate functional requirements. UML diagrams contain actors, relations and processes in order to capture functions related to user's needs. However, Use Cases focus mostly on the representation of what the system should do rather than on the representation of what the system should not do. Thus, as security requirements was not possible to be derived by this method, Use Cases methodology extended in order to capture the behavior that should be avoided by the system. This extended version of Use Cases is named ”Misuse Cases”.

Misuse Cases [42] are an inversion of Use Cases that uses mis-actors instead of actors in order to represent possible threats by misuse behaviors. However, due to the fact that analysts need to indicate functions that prevent or detect misuse, the authors of Misuse Cases suggest the representation of a Use Case and the relevant Misuse Case in the same diagram. As in Use Cases, the textual representation of Misuse Cases is also important. The authors of Misuse Cases encourage the use of templates for textual description of use cases, but with adaptions in some fields in order to fulfill the representation of misuse behaviors. Security requirements derive from the analysis of threats that come up from Misuse Cases. Privacy requirements have not been considered in Misuse Cases.

16) The RBAC method: He and Anton [43] presented an agent-oriented framework for modeling privacy requirements and user privacy preferences in the role engineering process. The RBAC framework maps with a systematic way roles and permissions, while considers privacy requirements as constraints on permissions and roles in order to define access control policies. The framework consists of a data model and a goal-driven role engineering process.

As the authors of the framework refer, a typical access control rule in an RBAC policy is expressed as: <u, r, p >. That means that a user u can only access an object (o), if he/she is assigned a role r, and if the role is assigned certain permission p, which is allowed to access the object (o). Thus, the authors of the RBAC method consider Roles (r), Permissions(p) and Objects (o) as the basic elements of an RBAC system. Roles, permissions and objects are called contexts. On the other hand, purposes, conditions and obligations are identified as privacy elements in an RBAC system. The authors consider privacy elements as attributes of contexts. The data model that is included in the proposed framework represents the way that these three privacy elements can be modeled in the RBAC system.

Additionally, the framework includes a goal-driven role engineering process in order to support the elicitation and modeling of the three privacy elements. This process is divided in two phases: Role-Permission Analysis (RPA) and Role-Permission Refinement (RPR). During RPA phase, business processes and business task are analyzing via goal and scenario-oriented requirements analysis techniques in order to identify the corresponding permissions, permission constraints and roles for each task. However, as the set of roles and permissions generated in this stage are probably ambiguous, they can be refined in the RPR phase according to organization

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

structure, policy statement e.t.c.

With this framework, the authors aim to bridge the gap between high-level privacy requirements and low-level-access control policies in the early stages of system analysis and design.

17) The M-N (Moffett - Nuseibeh) framework: As the authors noted a non satisfactory integration of security requirements into requirements engineering, they presented a framework for analyzing and eliciting security requirements from the early stage of the system analysis and design process [44]. The proposed framework combines concepts both from software requirements engineering (functional goals, functional requirements and constraints) and from security engineering (assets and threats) as well.

According to the proposed framework, the elicitation of security requirements takes place in two steps. Firstly, the authors propose the application of risk analysis and management techniques in order to identify the threats against the valuable assets and to decide the appropriate security measures. Next step includes the definition of high-level security goals. Security goals arise from the inversion of the threats identified in the previous step. Generally, security goals aim to protect the valuable assets by possible threats and are operationalized into security requirements. Security requirements are considered as a set of constraints in functional requirements.

18)The STRAP method: The STRAP [45] method is a goal-oriented approach that promotes the design of privacy-aware systems as, following this approach, system designers are able to analyze and elicit privacy requirements form the early stage of system design level. The method is based on a structured analysis of privacy vulnerabilities in design and on an iterative process for the adaption of preferences. Thus, in STRAP, the derived vulnerabilities are considered as privacy requirements and these vulnerabilities are presented as obstacles in the satisfaction of a system 's functional requirements.

The STRAP method includes four (4) steps:

1)Analysis: The system 's analysis step includes a goal-oriented analysis. During this analysis, different actors and their privacy expectations, privacy goals and sub-goals and all the major system components and their relevant limitations are identified. In parallel, STRAP uses a number of questions for each goal and sub-goal that has been identified earlier. The result of the questions leads to the identification of a privacy vulnerabilities set. Thereafter, the derived vulnerabilities are evaluated for possible duplicates and are categorized in order to proceed to the Refinement step.

2)Refinement: During this step, system designers have to check the existence set of vulnerabilities and to identify those that can be eliminated or mitigated in order to eliminate the provided set of vulnerabilities.

3)Evaluation: As several design scenarios are generated by different designers, the purpose of this step is the selection of this design that decreases mostly the risk by the relevant privacy vulnerabilities.

4)Iteration: Finally, in the iteration phase, all the previous steps are repeated in order the new detected vulnerabilities to be integrated in the system design. Thus, the goal-model is re-designed in order the relevant alterations to be included. When no alterations are needed, the iteration phase ends.

19)The NFR (Non-Functional Requirements) framework: The authors proposed a process-oriented approach in order to support system designers to design secure Information Systems via a systematic, integrated and automated process. Although the NFR framework [46] can be applied in every phase of a system development life cycle, the authors proposed the use of NFR framework at the early stage of system design level. The NFR framework considers the non-functional requirements, such as security, accuracy, performance and cost, as softgoals that have to be achieved by the development system. Softgoals are considered special types of goals that need to be clarified, disambiguated, prioritized, elaborated upon, etc.

The main idea of this framework is to identify security goals, to represent them in a goal graph structure, to examine any possible interactions between security goals and finally to assess the degree of a goal achievement. More specific, the NFR approach includes the incremental and interactive construction, elaboration and revision of a softgoal interdependency graph (SIG). The graph consists of nodes that represent the softgoals (security goals) as well as the interdependencies among softgoals. Also, the interdependencies represent the relation between general softgoals with more specific low level soft-goals. In parallel, the NFR approach includes an evaluation procedure that considers interdependencies in order to verify that the identified softgoals have been achieved. Finally, the derived set of softgoals is linked to the system functional requirements. Softgoals operate as constraints in the implementation of the system's functional requirements.

Specific NFR catalogues have been constructed for security requirements also considering privacy (as part of the confidentiality security requirement), which makes NFR a useful tool for defining privacy requirement and identifying possible design alternatives. Also, the authors of NFR approach provide an automated tool, namely RE-Tools toolkit, in order to assist designers to build their NFR models.

20)The i method:* The i* method [47] was introduced as an agent-oriented method, as it focuses on systems agents and their social interdependencies. The main interest is to map the organization's logic and context at the early stages of system design. In this direction, the i* method was first designed as a tool for modeling, analyzing and redesigning organization processes. However, recently, the method is used in order to model security and privacy requirements of a system into consideration.As with the NFR method, the i* method is based on the notion of softgoals. However, the i* method focuses on the individuals goals of the systems'actors and not on the overall system goals. The actors are considered interdependent as the achievement of their goals depends on other actors and their tasks.

Security analysis takes place by using several analysis techniques and is aiming at the construction of a domain model that will capture the involved actors and their dependencies. In particular, attacker analysis helps identify potential system abusers and their malicious intents (threats). Dependency vulnerability analysis helps detect vulnerabilities in terms of organizational relationships among stakeholders. Countermeasure analysis supports the dynamic decision-making process of addressing vulnerabilities and threats. The results of this accurate analysis can be used for further refinement of actor softgoals. The i* method includes an evaluation procedure in order system designers to decide whether the impact of threats and vulnerabilities has been eliminated to an acceptable level. Finally, depending on actor s tasks, a role-based access control analysis can be performed in order the appropriate actor's roles to be defined.

B. A Comparison of Security and Privacy Requirements Engineering Methods

Many different approaches in the area of security and privacy requirements engineering have been presented in previous section. In this section, we present a comparative study between these methodologies according to specific criteria. A first criterion that is used in our study is the ”Requirements” criterion and it is referred to the requirements that each method expects to meet. Some methodologies deal exclusively with the elicitation of security or privacy requirements but some others with both of them in parallel. Additionally, as each method bases the elicitation of security/privacy requirements on different concepts (i.e. goals, risks, threats etc) and processes, the approach that follows each method in order security/privacy requirements to be derived is used as a criterion in this comparative study. Another critical issue is the system development life cycle level where each method can be applied. Apart from these criteria, another set of criteria is examined. More specific, it is examined if:

- the assets of the system that have to be secured are considered by the method,

- any possible threats are considered by the method,

- a risk assessment is performed during the execution of the method,

- a categorization/prioritization of the derived security/privacy requirements is performed during the methodology,

- a requirements inspection step is included in the method,

- the method identifies and resolves any possible conflicts between the derived security/privacy requirements.

Table I summarizes and compares the aforementioned methodologies. A table entry that is labeled with Y or N means that the relevant criterion is considered or not by the relevant method.

A first remark is that most methods consider explicitly security or privacy requirements in order to design secure systems. On the other hand, the extension of KAOS, NFR and GBRAM method consider privacy as a subset of security. However, as privacy has separate aspects than security and a security incident might have a serious impact in user's privacy and vice versa, security and privacy requirements have to be examined in parallel under the same framework in order to design secure systems [7],[9-10]. The meta-model presented by Islam et al. [11] is able to support security and privacy requirements as it combines concepts from Secure Tropos and PriS methodologies that deal with security and privacy issues separately. Also, the i* method can support the elicitation of security and privacy requirements as well.

It is worth noting that all the aforementioned methodologies can be applied at the early stage of system analysis and design as a late reconsideration of security and privacy requirements can be extremely costly and time-consuming. LINDDUN, PriS methodology and therefore Secure Tropos and PriS metamodel include steps in order to fill the gap between system design and implementation stage and to support developers to select the most appropriate implementation technique. On the other hand, NFR method can be applied at all system development stages.

Each methodology has been build by using a different approach. MOSRE, Secure Tropos, KAOS, PriS, the Secure Tropos and PriS meta-model, GBRAM, M-N framework, STRAP and NFR method have been introduced as goal-oriented methodologies as security and privacy requirements are considered as organizational goals that have to be satisfied by the system into consideration. On the other hand, SQUARE methodology and SQUARE extension for integrating privacy requirements have been based on risk analysis results. It is worth noting that even if SQUARE method supports the identification of system threats and the corresponding vulnerabilities, the assets of the system that have to be secured are not considered by the method. On the contrary, the proposed methods by Ahmed et al. [19], MOSRE, SREF, SREP, M-N framework and i* method support risk analysis on business assets in order to elicit security/privacy requirements. Additionally, as many methodologies have integrated steps in order to support threat identification, SREP, LINDDUN, Misuse Cases and i* method put threat analysis in the center of their attention in order to elicit security or privacy requirements. SREF and presSure have been introduced as problem-based methods as the analysis and the elicitation of security requirements comes from the analysis of problem diagrams. Additionally, the RBAC and i* method have been characterized as agent-oriented methods. Both of these methods does not examine the overall security/privacy goals of the organization but the agent goals according to the roles that have been assigned to each agent.

Regarding the categorization/prioritization criterion, it could be noticed that for many methods this step is a logical extension of a risk analysis process. A categorization and prioritization of security or privacy requirements is an important aspect of many approaches, as, during this process, system designers have to decide if the implementation cost of a requirement is comparable with the value of the secured asset. SQUARE, MOSRE, SREP, LINDDUN, PriS, GBRAM, Misuse Cases and STRAP method support categorization/prioritization of requirements. Additionally, most of the approaches, SQUARE, MOSRE, SREF, SREP, PriS, the Secure Tropos with PriS metamodel, GBRAM,RBAC, M-N framework, SRAP and NFR method include steps for requirements inspection. Finally, MOSRE, Secure Tropos, PriS, the Secure Tropos with PriS meta-model, GBRAM and the NFR method examine the existence of any conflicts between requirements and security or privacy goals.

Table II presents the security and privacy requirements that each method aspires to cover. Where $\sim$ is labeled, that means that the author of the method does not specify the requirements that takes into consideration.

III. Security and Privacy Requirements Engineering Methods in Cloud Computing Environment

As presented in previous section, all the above methodologies were designed to contribute to system analysis and design in traditional architecture environments. As it is shown in Table II, the concepts of confidentiality, integrity and availability are the most frequent requirements that a method designed for traditional architecture systems examines. However, as a cloud computing structure is a more demanding environment, a methodology that is aiming to help system analysts and

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. COMPARISON OF SECURITY AND PRIVACY ENGINEERING METHODS

Method	Requirements	Approach	Stage	Assets	Risk Assessment	Categorization/Prioritization	Threats	Req. Inspection	Conflicts Identification
SQUARE	Security	Risk driven	Early Design	N	Y	Y	Y	Y	N
MOSRE	Security	Goal oriented	Early Design	Y	Y	Y	Y	Y	Y
SREF	Security	Problem based	Early Design	Y	Y	N	Y	Y	N
N. Ahmed et al.	Security	Asset based	Early Design	Y	Y	N	N	N	N
SREP	Security	Threat based	Early Design	Y	Y	Y	Y	Y	N
Secure Tropos	Security	Goal oriented	Early/Late Design	Y	N	N	Y	N	Y
KAOS	Security	Goal oriented	Early Design	N	Y	N	Y	N	N
PresSure	Security	Problem based	Early Design	Y	N	N	Y	N	N
LINDDUN	Privacy	Threat driven	Early/Late Design	N	Y	Y	Y	N	N
SQUARE for privacy	Privacy	Risk driven	Early Design	N	Y	Y	Y	Y	N
PriS	Privacy	Goal oriented	Early/Late Design - Implementation	N	N	Y	N	Y	Y
Secure Tropos with PriS	Security/Privacy	Goal oriented	Early/Late Design - Implementation	Y	N	N	Y	Y	Y
GBRAM	Security (Privacy is a subset)	Goal-oriented	Early Design	N	Y	Y	N	Y	Y
Abuse Frames	Security	Problem based	Early Design	Y	N	N	Y	N	N
Misuse Cases	Security	Threat Driven - UML Based	Early Design	N	N	Y	Y	N	N
RBAC	Privacy	Role based/Agent oriented	Early Design	N	N	N	N	Y	N
M-N framework	Security	Goal oriented	Early Design	Y	Y	N	Y	Y	N
STRAP	Privacy	Goal oriented	Early Design	N	Y	Y	Y	Y	N
NFR	Security (Privacy is a subset)	Goal oriented	All system development stages	N	N	N	N	Y	Y
i*	Security/Privacy	Agent oriented	Early Design	Y	Y	N	Y	N	N

*Y=Yes, N=No

designers to design a secure and privacy oriented information system in a cloud structure has to examine more specific requirements like user 's isolation, data portability, Cloud Service Providers transparency etc. Nevertheless, most of the methods presented in Section II, with the exception of Secure Tropos methodology, do not consider cloud security and privacy requirements and therefore could not be used during designing cloud systems. On the other hand, Secure Tropos methodology was extended in order to model special cloud security requirements [15]. A brief description of security and privacy requirements that are unique in a cloud structure is presented in Section IV.

In the recent years, as cloud computing has rapidly grown, many research efforts have been presented that consider security and privacy into the development process. Almorsy et al. [12] introduced a Model-Driven Security Engineering at Runtime (MDSE@R) approach for multi-tenant cloud-based applications. MDSE@R supports different tenants and service providers security requirements at runtime instead of design time by externalizing security from the application. More specific, service providers may impose some security controls as mandatory but multi tenants can also add extra security requirements at runtime at their own instance of the application. Fernandez et al. [13] presented a method on how to build a cloud Security Reference Architecture (SRE). An SRE is an abstract architecture that describes functionality without implementation details and includes security mechanisms to the appropriate places in order to provide a degree of security. This approach includes threat identification and uses misuse patterns in order to describe how an attack can be performed. Through this process, it can be verified that security patterns have been selected correctly and have been placed properly in the cloud architecture. In 2015, Perez et al. [14] presented a data-centric authorization solution, namely SecRBAC, in order to secure data in the cloud. SecRBC is a rule-based approach that provides data managing authorization to CSP through roles and object hierarchies. The authorization model uses advanced cryptographic techniques in order to protect data from CSP misbehavior also. In 2016, Mouratidis et al. [15] extended Secure Tropos requirements engineering approach for traditional software systems in order to enable modeling of security requirements that are unique in cloud computing environment and to support the selection of the appropriate cloud deployment model as well as the cloud service provider that best satisfies security requirements of the system under consideration. In 2013, Tancock et al. [34] presented the architecture of a Privacy Impact Assessment (PIA) tool in order to identify and evaluate possible future security and privacy risks on data stored in a cloud infrastructure. The risk summary that derives from PIA tool takes into consideration aspects like who the cloud provider is, what is the trust rating and what security and privacy mechanisms are used. As threat modeling is an important aspect for developing secure systems, Cloud Privacy Threat Modeling (CPTM) methodology [35] was proposed in order to support the identification of possible attacks and to propose the corresponding countermeasures for a cloud system through a number of specific steps. However, CPTM was designed in order to support only EU data protection directives and as a result the methodology presented a number of weaknesses in threat identification. Thus, A. Gholami and E. Laure [36] extended CPTM methodology in order to be complied with various legal frameworks. As it is hard for an organization to choose the appropriate cloud deployment

TABLE II. SECURITY AND PRIVACY REQUIREMENTS PER METHOD

Method	*Requirements*
SQUARE	CIA
MOSRE	CIA, Authentication, Authorization, Auditing
SREF	CIA, Accountability
N. Ahmed et al.	CIA, Authentication, Authorization
SREP	∼
Secure Tropos	CIA, Access control, Non-repudiation, Authentication, Accountability
KAOS	CIA, Privacy, Authentication, Non-repudiation
PresSure	CIA
LINDDUN	Unlinkability, Anonymity, Pseudonimity, Plausible deniability, Undetectability, Unobservability, Confidentiality, Content awareness, Policy & consent compliance
SQUARE for privacy	∼
PriS	Identification, Authentication, Authorization, Data protection, Anonymity, Pseudonimity, Unlinkability, Unobservability
Secure Tropos with PriS	All SecureTropos and PriS requirements
GBRAM	∼
Abuse frames	∼
Misuse Cases	∼
RBAC	∼
M-N framework	CIA
STRAP	∼
NFR framework	CIA
i*	∼

**CIA=Confidentiality, Integrity, Availability

type (public, private, hybrid or community), K. Beckers et al. presented a method that can support requirements engineers to decide which cloud deployment model best fits the privacy requirements of the system under consideration [37]. This approach is based on a threat analysis in parallel with the privacy requirements that the system shall satisfy and some other facts and assumptions about the environment like the number of stakeholders on each deployment scenario and the domains that have to be outsourced into a cloud.

Despite the fact that all these contributions develop different kind of mechanisms or processes that consider security and privacy issues in the context of cloud computing, most of them present a number of limitations. Some of them are related to specific cloud service models. MDSE@R is referred to a Software as a Service service (SaaS) model while the method for building a Security Reference Architecture is referred to an Infrastructure as a Service (IaaS) service model. On the other hand, most of the proposed frameworks, methods or processes in the context of cloud computing deal exclusively with security or privacy issues or in some cases privacy is considered as a subset of security. For instance, MDSE@R, secRBAC and SecureTropos consider only security issues while the Privacy Assessment Impact Tool (PIA), CPMT and the method for selecting the appropriate cloud deployment model focus explicitly on privacy issues. In our previous work [10], we presented the reasons why security and privacy have to be considered as two different concepts but have to be examined under the same unified framework. Nevertheless, one of the most important issues is that most of the proposed frameworks that are based on the idea of cloud computing integrate security and privacy controls during implementation phase and not earlier in requirements phase. But, such practices might create late corrections in security and privacy requirements, which means additional cost and severe delays in project delivery.

As cloud computing is a new and continuously developing environment, many research efforts have been presented over the last decade that highlight the need of adopting security and privacy mechanisms from the early stage of development life cycle. Nevertheless, until today security and privacy in the context of cloud computing is still performed as an ad-hoc process rather than an integrated process in the development life cycle. As it is mentioned above, Mouratidis et al. [15] presented a requirements engineering method in order to model cloud security requirements at the design level but no privacy requirements have been considered. Under these circumstances, literature presents a lack of integrated methods that through a number of specific steps could be able to support the parallel elicitation and analysis of cloud security and privacy requirements from the early stage of system design. It is worth noting that a security and privacy requirements engineering method at the design level should include steps in order to fill the gap between analysis and implementation phase in order to support system developers to select the appropriate technologies that best satisfy security and privacy requirements.

IV. CLOUD SECURITY AND PRIVACY UNDER THE SAME UNIFIED FRAMEWORK

The specific review aims on identifying the main security and privacy concepts that are proposed from the respective literature in the area of security and privacy requirements engineering from the relevant methodologies. Since most of the requirements engineering methods proposed in the literature were applied for the design and modelling of information systems in traditional environments, the paper aims to identify the existence of requirements engineering methods proposed explicitly for the modelling of Information Systems in cloud environments and the changes that these methods bring in terms of the set of concepts that need to be considered when designing cloud-based services. Thus, the proposed conceptual model is a beginning towards the direction of proposing a framework that will be used by software engineers for the design of Information Systems in traditional and cloud-based systems.

As it is mentioned above, a cloud infrastructure is a continuously developing and a very demanding architecture as many additional parameters have to be considered during designing and developing a cloud infrastructure. As a result, system designers and developers have to take into consideration all the special characteristics of a cloud infrastructure (on-demanding services, resources sharing, remote access etc) in order to provide a secure and privacy-aware environment. An accurate determination of security and privacy goals of the system into consideration can prove to be crucial for achieving this goal. In this section, we present a set of security and privacy requirements that have to be provided by a trustworthy cloud infrastructure.

A. Security Requirements:

- Integrity: The integrity of data refers to the preservation of data from a possible malicious, intentional or unintentional modification during storage or transmission.
- Confidentiality: The confidentiality concept is referred to the assurance that user 's data and information will not be disclosed to unauthorized persons. Data should remain confidential not only to other users but to Cloud Service providers and system administrators as well.
- Availability: As the idea of cloud computing is based on the idea of on-demand services, data availability is referred to the ability of a Cloud Service Provider to provide continuous service delivery. Users have to be able to access their stored data any time by any device.
- Non-repudiation: This property is aiming to ensure that user 's actions will not be repudiated later.
- Authentication: Authentication requirements is referred to the implementation of authentication mechanisms in order to prevent access to data from non-legitimate users.
- Authorization: Authorization follows authentication and is aiming to the accurate determination of the resources and services that an authenticated user can access.

B. Privacy Requirements:

- Data portability: A cloud infrastructure have to ensure that user 's data could be transferred anytime to another cloud service provider. This requires that data will follow a standard format during their storage in the cloud infrastructure.
- Interoperability: Interoperability is referred to the ability of a cloud service provider to cooperate and interoperate with different cloud systems.
- Anonymity: anonymity is defined as the ability of a customer to use cloud resources and services without being obliged to reveal his/her identity and without being tracked [48].
- Pseudonimity: The concept of pseudonymity is very closed to the concept of anonymity. The difference lies in the fact that with pseudonimity a user can access cloud resources and services without being obliged to reveal his identity but by acting under one or more pseudonyms.
- Unlinkability: In order a cloud service provider to provide privacy to customers must prevent linkage between data and the customer that processes the specific data. In parallel, the provider must protect the privacy of a communication between a sender and a recipient. Than means that a possible attacker, another user or cloud administrators should not be able to identify two entities that communicate.
- Undetectability: Cloud users should be able to access cloud resources and services without being detectable by potential attackers.
- Unobservability: The concept of unobservability in the cloud is aiming to keep cloud users not only undetectable but anonymous as well while interaction with cloud resources or other cloud users.
- Provenancability: The requirement of provenancability is referred to the need for a mechanism that collects data in a structure way in order to record the history of every piece of data that exists in a cloud infrastructure. However, as provenance data might reveal sensitive data, the cloud service provider should be able to keep them secure inside the cloud infrastructure.
- Transparency: In order users to trust a cloud vendor, they should be aware for the procedures and policies that the cloud vendor follows. As Gartner [49] supports, cloud providers have the obligation to provide customers with clear details about architectures, risk controls policies, data location, recovery mechanisms etc.
- Isolation: As a cloud infrastructure allows the sharing of resources between multi tenants, the cloud provider should guarantee a certain level of isolation in order to achieve the complete seal of user's data [48].
- Accountability: An accountable cloud service provider must provide to cloud users a full control on their data and to function with transparency about how their data are used. That includes the clear identification of data policies, the compliance with the identified policies, the ability of data recovery in case of violation and the monitoring of data. Auditing user 's data and maintaining log records are common practices in this direction.
- Intervenability: Any cloud user should have the right to intervene in data processing where he considers that the cloud provider violates the policies. The meaning of intervenability includes the rights to data access without limitations, rectification and erasure of data, objection to data processing when processing does not comply with rules as well as the right to withdraw consent [50].
- Traceability: Traceability is referred to the ability of a cloud vendor to register in log files every human activity during processing data in the cloud infrastructure.

Below, we present a conceptual model that considers cloud security and privacy concepts, in parallel, during the system design process. The proposed conceptual model has been based on PriS method that was first introduced as a privacy requirements engineering method [33] in systems with traditional architecture only. In our previous work [10], the PriS framework was extended in order to consider security and privacy concepts, in parallel, in a cloud environment too. Indeed, the conceptual model represents a modeling language of security and privacy organization goals and requirements.

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Also, the proposed conceptual model could be the base for developing a new requirements engineering method in a cloud environment that will consider system 's security and privacy requirements under the same unified framework. Such a method could contribute to the effective identification of security and privacy goals and requirements as well as to the effective evaluation of cloud providers.

In Figure 1, the central concept of the extended conceptual model is ”goal”. Goals are referred to any intentional objectives that an organization needs to achieve. Goals in a cloud environment are generated by the issues raised by stakeholders. Thus, goals can be derived by anyone involved in the cloud infrastructure (Cloud Service Provider, cloud users, system designers and administrators, external CSP, etc). For instance, a CSP must operate and provide services within a specific legal framework and must protect user 's privacy and data from any malicious attack. All these restrictions generate issues that in turn can generate new goals. Also, a SWOT (strength, weakness, opportunity, threats) analysis in a cloud based system might generate new issues and goals as well. Thus, an accurate identification of these issues at the early stage of system design level can contribute to the accurate determination of system 's objectives.

Processes can realise goals. However, the achievement of a goal might presuppose the achievement of one or more goals. Thus the origin goal has to be broken down to simpler goals by system designers in order each process to be applied in the relevant sub-goal and not directly to the origin goal. Also, a sub-goal might be related to the achievement of more than one goal, thus forming a structure of goals/sub-goals and their relationships. During this process, new goals can be identified and some others can be rejected or replaced in the hierarchy of goals. In Figure 1, the satisfaction relationships between goals and sub-goals is illustrated with the AND/OR decomposition entity.

The proposed conceptual model includes the examination of possible relations between two or more different goals. In this direction, two influence types are introduced in order system analysts to identify the relation between them and to examine whether two different goals are conflicting or not. The first influence type is referred as a Support relationship where the achievement of one goal assists in the achievement of another. The second one is illustrated as a Conflict relationship where the achievement of one goal prevents the achievement of another. In case of a conflict relationship, the involved stakeholders have to negotiate in order to resolve these conflicts.

As it is shown in Figure 1, goals are classified into three types: Organizational goals, Privacy goals and Security goals. Organizational goals are referred in the main objectives that an organization needs to achieve through the system into consideration. On the other side, privacy and security goals are introduced due to the special privacy and security concepts of a cloud based system. Anonymity, pseudonimity, undetectability, unlinkability, portability, interoperability and data protection have been identified as privacy-related concepts. Data protection includes the concepts of isolation, provenancability, traceability, intervenability, accountability and transparency as these concepts aim at protecting system or user's data in a cloud infrastructure. Unobservability is referred to the coexistence of undetectability of assets and anonymity of users. On the other side, integrity, confidentiality, availability, non-repudiation and access control have been indicated as security concepts. Additionally, authentication and authorization have been included in access controls concept as both aim at defining user's access level to the cloud infrastructure. However, privacy and security goals may have an impact on organizational goals as the identification of privacy and security requirements during system design might trigger new organization goals or reject others. Kalloniatis et al. in [48] described in details all the aforementioned security and privacy concepts.

As goals are realized by processes, it is proposed that system designers and developers use patterns in order to build processes with specific properties. Process patterns are general process models that deal with a specific issue through specific steps. In the security and privacy area, process patterns can help system designers to map the effect of security/privacy requirements on system processes and facilitate developers to select the technology (IDS, Digital Signature, PET's, etc) that best supports security and privacy goals. Thus, a system designer/developer should be able to select from a repository of patterns those that best fit in the process into consideration. Depending on the goal that a process is aiming to implement, the related pattern has to be selected. A privacy process pattern can be selected in case the relevant process aims to realize a privacy goal or a security process pattern can be used in order a security goal to be achieved. In general, security and privacy process patterns can standardize security and privacy procedures in order to support system designers to generate the appropriate processes that best satisfy security and privacy goals. In parallel, when processes have been generated based on specific process patterns, developers can select easily the appropriate security or privacy technologies that satisfy the specific process and therefore security or privacy goals.

V. Conclusion and Future work

In this paper, we presented a set of security and privacy requirements engineering methods that have been introduced by several researchers. Our research has focused on two areas: on those methods that aim to support software engineers to design and develop information systems hosted in traditional architectures and on those methods that can be applied in cloud systems. Thus, a narrow analysis for the security and privacy requirements engineering methods was performed, in order to map the area of security and privacy requirements methodologies as well as to record the gaps in this area and to justify the need for a holistic approach in the field of security and privacy. However, we will be able to provide extended information in this area in our future work.

As already mentioned, different security and privacy requirements engineering methods have been introduced in the past as software engineers community agree that security and privacy is still an integral part of the information systems design process. Referring to traditional architectures, there are different approaches that each method has been based on. For instance, security or privacy requirements can be derived from the determination of security or privacy goals, from the results of a risk analysis or from problem diagrams. Additionally, as it is clear from the above analysis, most researchers deal with security or privacy issues separately, a fact that can cause possible conflicts and late reconsiderations in functional requirements.

On the other hand, cloud computing is a more demanding structure as it introduces special characteristics like multi-

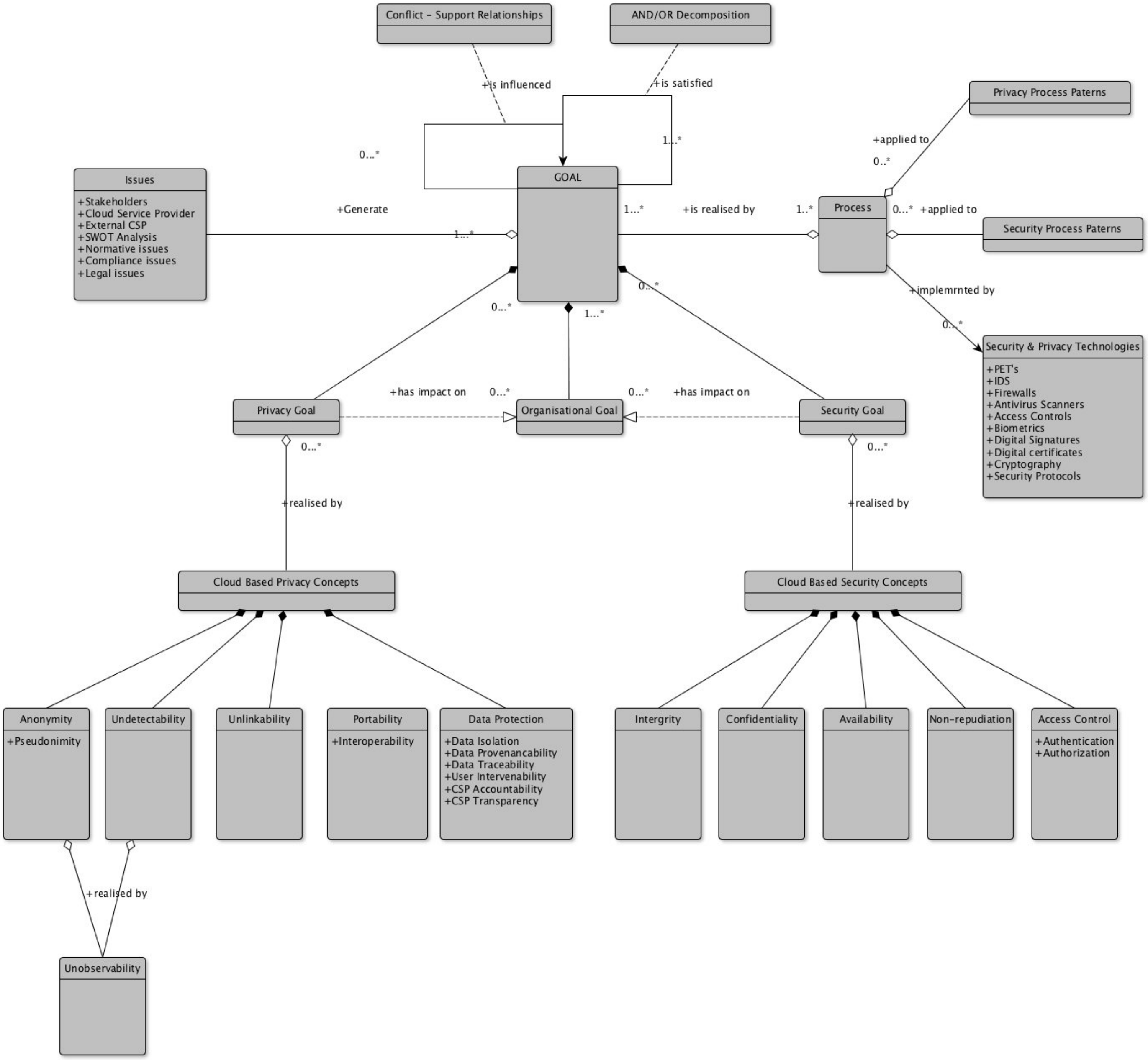

Figure 1. Conceptual model

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

tenancy and on-demand services. Special characteristics introduce new security and privacy concepts that software engineers have to take into account during system designing and developing. However, even though cloud computing presents a rapid growth last decade, all methods that have been presented by researchers present limitations while it is noting the lack of integrated methods that support the elicitation and analysis of security and privacy requirements in parallel.

The purpose of this research is to demonstrate that in cloud computing area there is a lack of integrated requirements engineering methods that consider security and privacy as two different concepts that have to be examined in parallel under the same unified framework. Thus, the aim of our analysis is to map and compare the existing methodologies in order to produce a unified framework that considers security and privacy concepts in parallel. This study constitutes the base for developing a new methodology in the cloud computing area that will consider security and privacy under the same unified framework.

References

[1] A. Pattakou, C. Kalloniatis, and S. Gritzalis, ”Security and Privacy Requirements Engineering Methods for Traditional and Cloud-Based Systems: A Review”, CLOUD COMPUTING 2017 8th International Conference on Cloud Computing, GRIDs, and Virtualization, P. Dini, (ed), February 2017, Athens, Greece, IARIA

[2] I. M. Alharbi, S. Zyngier, and C. Hodkinson, ”Privacy by design and customers perceived privacy and security concerns in the success of e-commerce,” Journal of Enterprise Information Management, vol. 26, no. 6, 2013, pp. 702-718

[3] R. Cullen, ”Culture, identity and information privacy in the age of digital government”, Online Information Review, vol. 33, no. 3, 2009, pp. 405-421

[4] Z. Karake Shalhoub, ”Trust, privacy, and security in electronic business: the case of the GCC countries”, Information Management Computer Security, vol. 14, no. 3, 2006, pp. 270-283

[5] M. Meingast, T. Roosta, and S. Sastry, ”Security and privacy issues with health care information technology”, Engineering in Medicine and Biology Society, 2006. EMBS’06, 28th Annual International Conference of the IEEE, 2006, pp. 5453-5458

[6] S. E. Sarma, S. A. Weis, and D. W. Engels, ”RFID systems and security and privacy implications”, International Workshop on Cryptographic Hardware and Embedded Systems, Springer Berlin Heidelberg, 2002, pp. 454-469

[7] S. Gritzalis, ”Enhancing Web privacy and anonymity in the digital era”, Information Management and Computer Security, vol. 12, no. 3, 2004, Emerald Group Publishing Limited, pp. 255-288

[8] H. Mouratidis, P. Giorgini, G. Manson, and I. Philp, ” A Natural Extension of Tropos Methodology for Modelling Security”, Proceedings Agent Oriented Methodologies Workshop, Annual ACM Conference on Object Oriented Programming, Systems, Languages (OOPSLA), Seattle, USA, 2002

[9] C. Kalloniatis, E. Kavakli, and S. Gritzalis, ”Addressing privacy requirements in system design: The PriS method”, Requirements Engineering Journal, vol. 13, no. 3, 2008, pp. 241- 255

[10] A. Pattakou, C. Kalloniatis, and S. Gritzalis, ”Reasoning About Security and Privacy in Cloud Computing under a Unified Meta-Model”, In Proceedings of the Tenth International Symposium on Human Aspects of Information Security Assurance, HAISA 2016, pp. 56

[11] S. Islam, H. Mouratidis, C. Kalloniatis, A. Hudic, and L. Zechner, ”Model based process to support security and privacy requirements engineering”, International Journal of Secure Software Engineering (IJSSE), 2012, vol. 3, no. 3, pp. 1-22

[12] M. Almorsy, J. Grundy, and A. S. Ibrahim, ”Adaptable, model-driven security engineering for SaaS cloud-based applications”, Automated software engineering, vol. 21, no. 2, 2014, pp. 187-224

[13] E. B. Fernandez, R. Monge, and K. Hashizume, ”Building a security reference architecture for cloud systems”, Requirements Engineering, 2015, pp. 1-25

[14] J.M.M. Perez, G. M. Perez, and A. F. Gomez-Skarmeta, ”SecRBAC: Secure data in the Clouds”, IEEE Transactions on Services Computing, 2016

[15] H. Mouratidis, N. Argyropoulos, and S. Shei, ”Security Requirements Engineering for Cloud Computing: The Secure Tropos Approach”, Domain-Specific Conceptual Modeling, Springer International Publishing, 2016, pp. 357-380

[16] N. R. Mead and T. Stehney, ”Security quality requirements engineering (SQUARE) methodology”, ACM, 2005, vol. 30, no. 4, pp. 1-7

[17] P. Salini and S. Kanmani, ”Model oriented security requirements engineering (MOSRE) framework for Web applications”, Advances in Computing and Information Technology, Springer Berlin Heidelberg, 2013, pp. 341-353

[18] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, ”Security requirements engineering: A framework for representation and analysis”, IEEE Transactions on Software Engineering, 2008, vol. 34, no. 1, pp. 133-153

[19] N. Ahmed and R. Matulevicius, ”A Method for Eliciting Security Requirements from the Business Process Models”, In CAiSE (Forum/Doctoral Consortium), 2014, pp. 57-64

[20] D. Mellado, E. Fernandez-Medina, and M. Piattini, ”A common criteria based security requirements engineering process for the development of secure information systems”, Computer standards interfaces, vol. 29, no. 2, 2007, pp. 244-253

[21] B. Fabian, S. Grses, M. Heisel, T. Santen, and H. Schmidt, ”A comparison of security requirements engineering methods”, Requirements engineering, 2010, vol. 15, no. 1, pp. 7-40

[22] Infrastructure, Public Key, and Token Protection Profile, ”Common criteria for information technology security evaluation.” National Security Agency, 2002

[23] J. Castro, M. Kolp, and J. Mylopoulos, ”Towards requirements-driven information systems engineering: the Tropos project”, Information systems, vol. 27, no. 6, 2002, pp. 365-389

[24] H. Mouratidis, P. Giorgini, G. Manson, and I. Philp, ”A Natural Extension of Tropos Methodology for Modelling Security”, In Workshop on Agent-oriented methodologies, Annual ACM Conference on Object Oriented Programming, Systems, Languages (OOPSLA), 2002

[25] A. V. Lamsweerde and E. Letier, ”Handling obstacles in goal-oriented requirements engineering”, IEEE Transactions on Software Engineering, vol. 26, no. 10, 2000, pp. 978-1005

[26] S. Pachidi, ”Goal-Oriented Requirements Engineering with KAOS”, 2009, Method Description for ”Method Engineering” course Master in Business Informatics, Utrecht University

[27] A. V. Lamsweerde, ”Elaborating security requirements by construction of intentional anti-models”, Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society, 2004

[28] S. Fassbender, M. Heisel, and R. Meis, ”Functional requirements under security pressure”, Software Paradigm Trends (ICSOFT-PT), 9th International Conference on IEEE, 2014

[29] S. Fabender, M. Heisel, and R. Meis, ”Problem-Based Security Requirements Elicitation and Refinement with PresSuRE”, International Conference on Software Technologies, Springer International Publishing, 2014, pp. 311-330

[30] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, ”A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements”, Requirements Engineering, 2011, vol. 16, no. 1, pp. 3-32

[31] A. Bijwe and N. R. Mead, ”Adapting the square process for privacy requirements engineering”, 2010

[32] S. Miyazaki, N. Mead, and J. Zhan, ”Computer-aided privacy requirements elicitation technique”, Asia-Pacific Services Computing Conference, APSCC’08, IEEE, 2008

[33] C. Kalloniatis, E. Kavakli, and S. Gritzalis, ”Addressing privacy requirements in system design: the PriS method”, Requirements Engineering, vol. 13, no. 3, 2008, pp. 241-255

[34] D. Tancock, S. Pearson, and A. Charlesworth, ”A privacy impact

assessment tool for cloud computing”, Privacy and security for Cloud computing, Springer London, 2013, pp. 73-123

[35] A. Gholami, A. S. Lind, J. Reichel, J.E. Litton, A. Edlund, and E. Laure, ”Privacy threat modeling for emerging biobankclouds”, Procedia Computer Science, 2014, vol. 37, pp. 489-496

[36] A. Gholami and E. Laure, ”Advanced cloud privacy threat modeling”, arXiv preprint arXiv:1601.01500, 2016

[37] K. Beckers, S. Fabender, S. Gritzalis, M. Heisel, C. Kalloniatis, and R. Meis, ”Privacy-Aware Cloud Deployment Scenario Selection”, In International Conference on Trust, Privacy and Security in Digital Business, 2014, September, pp. 94-105, Springer International Publishing

[38] A. I. Anton and J.B. Earp, ”Strategies for developing policies and requirements for secure electronic commerce systems”, E-commerce security and privacy, vol. 2, 2000, pp. 29-46

[39] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett, ”Introducing abuse frames for analysing security requirements”, In Requirements Engineering Conference Proceedings, 11th IEEE International, 2003, pp. 371-372

[40] M. Jackson, ”Problem Frames: Analyzing and structuring software development problems”, Addison-Wesley, 2001

[41] I. Jacobson, ”Object-oriented software engineering: a use case driven approach”, Pearson Education India, 1993

[42] G. Sindre and A. L.Opdahl, ”Eliciting security requirements with misuse cases”, Requirements engineering 10.1, 2005, pp. 34-44

[43] Q. He and A.I. Anton, ”A framework for modeling privacy requirements in role engineering”, In Proceedings of REFSQ, 2003, vol. 3, pp. 137-146

[44] J. D. Moffett and B. A. Nuseibeh, ”A framework for security requirements engineering”, REPORT-UNIVERSITY OF YORK, DEPARTMENT OF COMPUTER SCIENCE YCS, 2003

[45] C. Jensen, J. Tullio, C. Potts, and E.D. Mynatt, ”STRAP: a structured analysis framework for privacy”, Georgia Institute of Technology, 2005.

[46] L. Chung, B. Nixon, E. Yu., and J. Mylopoulos, ” Non-Functional requirements in software engineering”, Kluwer Academic Publishers, Massachusetts, USA, 2000

[47] L. Liu, E. Yu, and J. Mylopoulos, ” Security and privacy requirements analysis within a social setting”, In Proceedings of Requirements Engineering Conference, 2003, 11th IEEE International, IEEE, pp. 151-161

[48] C. Kalloniatis, H. Mouratidis, M. Vassilis, S. Islam, S. Gritzalis, and E. Kavakli, ” Towards the design of secure and privacy-oriented information systems in the cloud: Identifying the major concepts”, Computer Standards Interfaces, 2014, vol. 36, no. 4, pp. 759-775

[49] J. Brodkin, ”Gartner: Seven cloud-computing security risks”, Network World, 2008, pp. 1-3: https://www.networkworld.com/article/2281535/data-center/gartner–seven-cloud-computing-security-risks.html

[50] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data

2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org