Security and Privacy Requirements Engineering Methods for Traditional and Cloud-Based Systems: 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.**

Keywords–security; privacy; requirements; engineering methods; traditional architectures; cloud computing.

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]-[6].

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. 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 [6]-[8] 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 [9].

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 [10]-[12]. However, as the cloud computing area still suffers from a lack of integrated requirements engineering methods, methods that were initially introduced for traditional architecture systems were extended in order to be applied in cloud systems as well [13]. But, at the moment, a method

 ISBN: 978-1-61208-529-6

for cloud architecture that supports the parallel examination of security and privacy concepts has not been introduced.

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. Section 2 presents a set of security and privacy requirements engineering methods for traditional architectures and a comparative study among them. Section 3 is referring to security and privacy requirements engineering methods that can be applied in a cloud environment and Section 4 concludes the paper.

II. Security and Privacy Requirements Engineering Methods in Traditional Architectures

A. Security and Privacy Requirements Engineering methods

1) Security Quality Requirements Engineering (SQUARE) Methodology: SQUARE methodology [14] 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, P. Salini and S. Kanmani introduced a security requirements engineering framework (MOSRE) [15] 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. [16] 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) [16] 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 : N. Ahmed and R. Matulevicius introduced an asset based approach in order to elicit security goals from business process models and translate them into security requirements [17]. 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 [18] 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 [19]. 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 [20] 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 [18]. 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 [21] was introduced by Castro et al. in order to cover system requirements

 ISBN: 978-1-61208-529-6

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 [22] 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 [22]. 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 [23] 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 [24]. However, due to the fact that KAOS methodology considers only functional requirements, authors extended KAOS [25] 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 [26]-[27] 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 [28] 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: After noting that, apart from security, privacy needs more attention during developing software systems, the authors of SQUARE methodology [14] adapted their approach in order to support the elicitation of privacy requirements at the early stages of software development process [29]. The extended framework includes the same steps as the original SQUARE method in conjunction with the Privacy Requirements Elicitation Technique (PRET) [30], 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 [31] 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 [6]-[7] 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. [9] 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

 ISBN: 978-1-61208-529-6

to meet the goals but also the appropriate architecture and implementation technique can be selected in order for privacy goals to be satisfied.

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 the previous section. Table 1 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 method considers 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[6]-[8]. The meta-model presented by Islam et al. [9] 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.

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.

Each methodology has been build by using a different approach. MOSRE, Secure Tropos, KAOS, PriS and the Secure Tropos and PriS meta-model 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. [17], MOSRE, SREF and SREP support risk analysis on business assets in order to elicit security requirements. Additionally, as many methodologies have integrated steps in order to support threat identification, SREP and LINDDUN 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.

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 and PriS support categorization/prioritization of requirements. Additionally, most of the approaches, SQUARE, MOSRE, SREF, SREP, PriS and the Secure Tropos with PriS metamodel include steps for requirements inspection. Finally, MOSRE, Secure Tropos, PriS and Secure Tropos with PriS meta-model examine the existence of any conflicts between requirements and security or privacy goals.

Table 2 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.

III. Security and Privacy Requirements in Cloud Computing Environment

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. [10] 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. [11] 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 [12] 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. [13] 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. [32] 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 [33] 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

 ISBN: 978-1-61208-529-6

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

*Y=Yes, N=No

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	$\sim$
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	$\sim$
PriS	Identification, Authentication, Authorization, Data protection, Anonymity, Pseudonimity, Unlinkability, Unobservability
Secure Tropos with PriS	All SecureTropos and PriS requirements

**CIA=Confidentiality, Integrity, Availability

and as a result the methodology presented a number of weaknesses in threat identification. Thus, A. Gholami and E. Laure [34] 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 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 [35]. 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 [8], 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. This framework has also been presented in our work. 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

 ISBN: 978-1-61208-529-6

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. [13] 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. 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.

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-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. This study along with our previously proposed conceptual framework [8] will be 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] 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

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

[3] 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

[4] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

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

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

[13] 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

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

[15] 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

[16] 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

[17] 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

[18] 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

[19] 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

 ISBN: 978-1-61208-529-6

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

[21] 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

[22] H. Mouratidis, "A natural extension of tropos methodology for modelling security", 2002

[23] 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

[24] S. Pachidi, "Goal-Oriented Requirements Engineering with KAOS", 2009

[25] 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

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

[27] 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

[28] 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

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

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

[31] 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

[32] 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

[33] 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

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

[35] 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

 ISBN: 978-1-61208-529-6