Contextualizing Security Goals: A Method for Multilateral Security Requirements Elicitation

Seda F. Gürses
Humboldt University, Department of Information Systems, Berlin
seda@hu-berlin.de
Thomas Santen
Berlin University of Technology, Department of Computer Science, Berlin
santen@acm.org

Abstract: In networked environments, the importance of eliciting security requirements as part of the process of requirements elicitation is increasing. Yet, it is difficult to articulate what security requirements are and how they can be elicited and implemented in the system. Few security requirements elicitation methods deal with the security needs of the end-users in specific contexts towards other end-users of the system – who may or may not be malicious. Often the affects of the users' social as well as functional roles on their security interests are disregarded. The goal of this paper is to attend to these problems by introducing a method that integrates the process of eliciting security requirements of the end-users into the requirements elicitation process of a multilaterally secure system. Throughout the method emphasis is put on contextualizing security goals by analyzing the different viewpoints: whose security goal is it? against whom? for which functionality? which other users have a mutual interest in or conflict with the given security goal?

1 Introduction

The importance of considering security requirements from the very beginning of the software engineering process is reflected in the research papers of the last years [LYM03, MN03, AE01]. Although the importance of the social contexts and the security interests of the end users are mentioned, often security requirements analyses only insinuate the security interests of the end-users and focus on threat and vulnerability analysis.

This paper introduces a method for eliciting security requirements for a multilaterally secure system. In a multilaterally secure system [RPM99] the security interests of all the stakeholders of the system are considered. Each person has the chance to individually decide on his/her security interests and on the mechanisms to implement these interests within the system. The security interests of the stakeholders of a multilaterally secure system are expressed in terms of *security goals*. With our method we propose a way to elicit these security goals and later make suggestions on how these can be composed to finalize the security requirements of a multilaterally secure system.

The security interests we consider are not limited to security goals against malicious at-

tackers but also towards all users of the system. We assume that users have common as well as conflicting security interests towards the system and towards other users, depending on their social context.

The method requires breaking down what we mean by users and what we mean by a system. In order to distinguish the diverging functional and security interests of the different users we work with the concepts of *roles* and *subgroups*. Further, the method proposes distinguishing the interests that users have towards different parts of the system. To realize this we make use of abstractions of system functionalities called *episodes*. Moreover, we *contextualize* each security goal. For each security goal the method identifies whose security goal it is, against whom, for which functionality of the system. Last but not least, suggestions are made on how to compose the different security goals into valid and consistent security requirements.

Throughout the paper, we assume that a requirements team is managing the steps defined in our method. Ideally, such a requirements team is made up of users, developers as well as further stakeholders of the system.

In Section 2 we shortly introduce the taxonomy of security goals and the analysis of security goals that we utilize in our method. Section 3 shortly introduces an example which we use to illustrate our method. Section 4 describes our method step by step, shortly outlining the essentials of each step. In Section 5, we review research in security requirements engineering and sketch the differences in approaches and methods. Section 6 concludes and lists prospects for future research.

2 Security goals

The security interests of the users are articulated in terms of security goals. *Security goals* articulate system behaviors that are wanted or not wanted by the end users. We distinguish between confidentiality, integrity and availability goals.

More specifically, we distinguish between the confidentiality goals confidentiality, pseudonymity and unobservability and the integrity goals integrity and accountability. In a communication setting the following are the definitions of the following security goals. *Confidentiality:* no one other than the communication partners can recognize the content of the communication. *Pseudonymity:* a user does not identify herself but is still accountable for her actions. *Unobservability:* no one other than the communication partners are aware that a communication is taking place. *Integrity:* any modifications made to the content of communication can be recognized by the communication partners. *Accountability:* communication partners can prove the existence of a communication to a third party.

In our analysis of the security goals of the users, we also make use of the characteristics of security goals as listed by Wolf [Wol01]. Characteristics of security goals exhibit how security goals impact each other. Security goals can enforce or contradict each other. This information is especially useful for a requirements team when they are negotiating between conflicting security goals to obtain a set of consistent requirements. For example, Wolf states that anonymity and accountability weaken each other, whereas anonymity and

confidentiality strengthen each other. Unobservability implies anonymity and therefore strengthens confidentiality. Accountability implies integrity. We will make use of these analyses.

3 The Intranet

In order to give a better presentation of our method we make use of an Intranet example. The objective of the Intranet is defined as follows:

An institute with a fellowship program has decided to introduce an Intranet in order to improve the curricular and administrative assistance of its scholars. All stakeholders of the Intranet, which include the institution, staff, scholars, alumni and developers of the system, are interested in having a computer system which securely mediates all the activities and communication within the Intranet.

Our goal is to elicit the security requirements, as well as the functional requirements of the Intranet. For the sake of compactness, we focus on the functional and security requirements of a subset of the users, namely scholars who are PhD students. The PhD students want to have the possibility to exchange ideas on their theses in a confidential manner. Especially they would like their communication to remain confidential to the staff of the fellowship program. This communication requires trust within a closed circle of users. We will now go through our method and analyze the functional and security requirements for the meeting of the PhD students of the Intranet.

4 The Security Requirements Elicitation Method

We give a short overview of the most important terminology before we discuss the tasks that make up our method, also illustrated in Figure 1. Since we focus on multilateral security, we are interested in distinguishing between the heterogeneous security and functional requirements of the various stakeholders. Further, we are interested in distinguishing the different functionalities of the system for which the security requirements vary. The terminology that we introduce helps in breaking down the categories stakeholders and system in order to elicit multilateral security requirements.

The *stakeholders* of the system are all persons involved in the conception, production, use, and maintenance of the system. All stakeholders have security interests in the system. *Users* are the stakeholders who will potentially interact with the system. Here, we concentrate on the users of the Intranet.

Each user has relationships with the system and with the other users of the system. These determine the security interests of the user in the system. In order to differentiate between users with their variable relationships we define the term roles. A *role* contains information either about the relationships of the user to the system or the relationships of the user to other users of the system. Examples of types of roles are social roles, functional roles and security roles. A *social role* describes the set of behaviors and interests a person may have

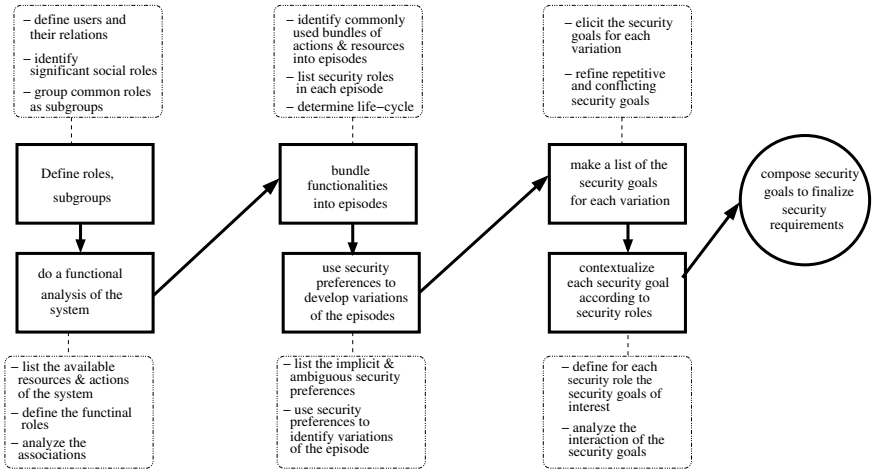

Figure 1: The method consists of seven steps in which the security goals of the users for the variations of episodes are elicited and refined into security requirements.

as a consequence of belonging to a social group. A *functional role* describes the set of resources a person may use and the actions a person may take within a system. A *security role* defines the security goals a person has or the security goals attributed to a person for a specific functionality of the system. A role may be attributed to many users and each user may have many roles. A set of users with common roles are defined as a *subgroup* of the system.

Users attribute security interests to the system and/or to a set of users of the system. Yet, there are sets of functionalities or resources of the system for which the security and functional interests of the users vary. A useful analysis of the users' varying security interests requires the definition of meaningful instances that occur in the system. These instances consist of actions and resources within a time frame for which common security interests for a set of users can be defined. Each of these instances are called *episodes* of the system.

We shall define further terminology as we move along. In the next sections we go through the method step by step in which we show how we use roles and episodes to elicit security goals.

4.1 Roles, subgroups and functionalities

The roles and subgroups: The first step of the method calls for a requirements team to classify the users with their corresponding roles and subgroups. This process includes defining any restrictions users may have in having roles or belonging to subgroups. Combinations of roles or subgroup membership which are allowed or restricted are documented. Further, the relationships of the users with the system as well as the relationships that the users have amongst each other and their affect on the user's interests are taken into

account.

In the Intranet, the users originally have one of three social roles: staff, scholars and alumni of the fellowship program. These social roles determine the users' interests towards the Intranet:

The members of the staff see the Intranet as an organizational tool supporting the administrative and communicative processes of the scholars in the fellowship program. The scholars share these interests. In addition, the scholars see the Intranet as a system for supporting their self-organization and networking. The alumni are less interested in the administrative processes but more in the networking possibilities. The members of the staff make decisions concerning the scholars. Therefore, the scholars are interested in being able to use the Intranet without the intervention of staff.

Because of the conflicting interests in the Intranet, it is necessary to insert restrictions on multiple roles. For example, a user may not be a scholar and a staff member at the same time. Or, the requirements team may define a new scholar-staff role for those scholars who are employed at the institute to emphasize the differences in their security interests.

Here we must emphasize that by roles, we are not talking about the roles used to implement "role based access control" (RBAC). Sophisticated variations of RBAC allow developers to establish relationships between roles, between permissions and roles, and between users and roles. In contrast, we use roles to emphasize the importance of the different viewpoints on the system among the users depending on their social context, functional capabilities, and security interests.

The Functional Analysis: In order to talk about the security goals of the users, we need to define the functionalities of the system that these security goals are attributed to. The functionalities of the system consist of actions that can be taken on the resources of the system. Associations among these resources and actions are part of the definition of the functionalities. Each subgroup has its functional interests in the system. These interests are used to elicit an initial set of functional requirements. In order to elicit the functional requirements we define functional roles. Functional roles reflect the relationships users can have to the functionalities of the system. Functional roles bundle a set of actions and resources.

Let us observe the PhD students. They are interested in having a hidden communication with a selected group of users. This communication will have a beginning and an end and a group of attendees. These characteristics are similar to that of a meeting. Here is the definition of a meeting of the PhD students:

A PhD meeting is organized by an organizer. The organizer initiates the meeting, invites selected users, and, when the meeting is over, ends the meeting. Users may decide to attend the meeting if they are invited to the meeting. Until they attend the meeting they are invitees of the meeting. If they decide to attend the meeting, they are attendees of the meeting. Attendees may leave the meeting.

Here we observe three functional roles: the organizer, the invitee and the attendee of the PhD meeting. Further, we define the Intranet as such, and that users of the Intranet all use pseudonyms for communication. Any user may have multiple pseudonyms with varying degrees of pseudonymity for different functional roles.

The initial functional requirements document of the system consists of a list of the re-

sources, actions, functional roles and their associations. The elicitation of security goals and consequently security requirements is based on this document. Next, we define episodes by bundling functionalities of common interest to subgroups.

4.2 Episodes and their variations

Identifying meaningful episodes: Defining the security goals each user has for every single action in the system would lead to a complex document. Such a document is difficult to handle, has many repetitions, and delivers little overview for analysis. In addition, attributing security goals to actions of the system makes it difficult to analyze global security goals attributed to the whole system or to major components of the system. In order to avoid listing the security goals for each role, we talk about the security goals of the subgroups. Moreover, we bundle together functionalities that provide us with meaningful instances of the system towards which subgroups of users may have common security goals. These instances of the system are called *episodes*.

The granularity of each episode is negotiated by the requirements team. If subgroups have the same security interests towards two or more episodes, an overarching episode may be considered. If the same subgroup has varying security interests attributed to parts of an episode, it may be necessary to define sub-episodes. In the case of the PhD students' meeting, the episode "meeting" is not sufficient:

The PhD students have an interest in keeping the communication around their meeting hidden. This brings along a set of security goals directed at the organizer, invitees and attendees of the meeting. Once an invited user decides to attend a "meeting", the attendees attribute a different set of security goals to that user concerning the contributions made at the PhD students' meeting. These are about the integrity and confidentiality of the contributions, rather than the invitations to and the conditions of attending the meeting. Hence, two episodes are defined for each PhD meeting. The first episode covers the communication around the organization of a meeting, called the "meta-level communication of a meeting". The second episode covers all the communication that takes place within a meeting, called the "content-level communication of a meeting".

In the rest of the paper, we will only analyze the communication at the meta-level of a meeting.

Once the episodes with the right granularity are identified, subgroups associated with the episode can be assigned security roles according to their security interests. Security roles allow the requirements team to classify all the users who have security interests in the episode and all the users these security interests are directed at. The security roles assigned at the meta-level communication of a meeting organized by PhD students are the following: the *organizer* of the meeting; the *attendees* of the meeting; the *invitees* of the meeting; *other users* who have not been invited to the meeting.

That the functional and security roles have the same name in this case is a matter of coincidence and not a rule. The security role "users who have not been invited to the meeting" is an *inverse role*. An inverse role is assigned to a user or stakeholder because s/he has not been assigned a set of other roles. Users will not literally be assigned inverse roles. Yet,

inverse roles are necessary to talk about security goals attributed at such users.

The temporality of security roles are also essential to defining security requirements. By temporal we mean chronological order and not absolute time. In order to better understand the relationship between security roles and episodes we define life-cycles of episodes. Within the life-cycles the states of the episode are defined and the security roles involved in each state of the episode. We describe the first three states of the life-cycle of the meta-level communication of the PhD students in the Intranet as follows:

1. a PhD student using a pseudonym initiates a meeting for the PhD students and becomes the organizer of the meeting.
2. once the meeting has been initiated and characterized, the organizer invites other PhD students using their pseudonyms.
3. if the invited PhD student is interested in the meeting, s/he then attends the meeting.

In the life-cycle of the meeting of the PhD students the security role of an invitee changes as soon as she becomes an attendee. The security role of the organizer remains unchanged when an invitee decides to attend the meeting. The security interests of the organizer towards the invitee are changed to those towards an attendee.

Variations of Episodes: Often in requirements analysis, the process of defining the security characteristics is detached from the process of defining the functionalities of the system. Actually, every episode or each of its functionalities, by definition, has security characteristics implicit to it. As soon as we talk about who can use functionalities in an episode, for what purposes, how and when, we start making assumptions about the security characteristics of the functionalities.

In order to accommodate the security interests of the users, we need to make explicit the security characteristics of the functionalities. These we call the *security preferences* of episodes. Once the episodes are well defined, the requirements team shall elaborate the security preferences implicit to the episodes. This requires identifying security characteristics which are implicit and ambiguous, studying permutations of preferences, and negotiating plausible alternative episodes, called *variations*. Some of the security characteristics implicit to the hidden meeting of the PhD students and their possible ambiguities are as follows:

- only PhD scholars and alumni may be invited to the meeting
- ambiguous: in order to be able to invite only PhDs to the meeting, it shall be possible for the organizer to identify PhD scholars and alumni despite their pseudonyms. It is ambiguous which pseudonyms are appropriate for a meeting of the PhD students.
- ambiguous: the communication around the organization of a PhD meeting shall be hidden from those not invited to the meeting. It is ambiguous if the invitations shall be hidden from other invitees or attendees of the PhD meeting.

The ambiguities can be solved by considering the different permutations of the ambiguous security preferences. For example, in one variation all attendees maybe informed of all the invitations sent out to the invitees. In another variation, only the organizer may be informed of all the invitees. The final variations of the episodes need to be evaluated and negotiated by the members of the requirements team. The variations of the episodes also have an effect on the functionalities included in the system. This shows that making the security characteristics explicit and considering the diverging security interests of the users has an impact on the functionalities of the system.

We identify "hidden meeting" as a variation of meetings in the Intranet. According to the analysis of the security interests of the subgroup PhD students we can revise the first three steps of the life-cycle of the hidden PhD meeting as follows:

1. a PhD student using a non-anonymous pseudonym initiates a hidden meeting for the PhD students and becomes the organizer of the meeting. Anonymity is not allowed in a hidden meeting since it weakens accountability, which strengthens the confidentiality of the meeting.

2. once the meeting has been initiated and characterized the organizer invites other PhD students using their non-anonymous pseudonyms. Anonymous pseudonyms would make it impossible to identify PhD students. The invitees are informed of the existence of the hidden meeting.

3. Attendees of the hidden meeting may be informed of the invitees of the hidden meeting but not the other way around. The attendees are informed of the other attendees of the hidden meeting.

Once the variations and their security preferences are finalized the requirements team may elicit the security goals for each variation.

4.3 Eliciting and contextualizing security goals

Eliciting security goals: The requirements team elicits the security goals from the security preferences. This requires rephrasing security preferences as goals. Once the security goals are abstracted from the security preferences, the interactions among the security goals can be analyzed. Moreover, security preferences may contain repetitive or conflicting goals. It may also be the case that additional security goals need to be defined in order to support the security goals elicited from the security preferences. The requirements team needs to then negotiate the conflicts, refine the goals, and add missing goals in order to derive the final list of security goals of a variation of an episode.

Accountability is the only integrity goal considered in the preferences of the PhD meeting. In the Intranet example, we state that availability and integrity are prerequisites to all confidentiality goals. The following are a subset of the security goals of the hidden PhD meeting elicited from the security preferences of the variation:

PhD.1 only PhD scholars may be invited to the hidden PhD meeting.

PhD.2 all communication around the organization of the hidden PhD meeting shall remain confidential and unobservable to all users not attending the hidden PhD meeting.

PhD.3 users shall not be invited to the hidden PhD meeting using an anonymous pseudonym.

Conflicts in security goals may lead to further variations. In some cases, it may not be practical to conceive yet another variation of the episode. In such cases, the conflict in security interests among the users can be resolved by negotiating security goals. The characteristics of security goals as done by Wolf [Wol01] are useful in negotiating conflicting security goals. Do the security goals weaken or strengthen other security goals attributed to the variation? Is it possible to take a weaker form of one of the security goals in order to solve the conflict? Once the security goals are negotiated or prioritized security preferences and variations can be revised. Next, we contextualize each of the security goals.

Contextualizing security goals: The episodes and variations define what the object of each security goal is. The security preferences and elicited security goals identify the users who actively participate in realizing the security goal and the users the security goals are directed at. Next, we analyze which users have a vested interest in the security goals and why.

Not all users with a security role in an episode variation will share the complete list of security goals associated with an episode. The relationship of each security role to a variation is different and hence may represent diverging interests for each security goal. For each security goal, the requirements team shall identify why this is the case from the perspective of the security role. They shall define the scope of the security goal and analyze how the security goal impacts the other security goals the user has an interest in.

The contextualization of security goals is important for the implementation of security requirements. Often the implementation of security measures or the restriction of functionalities does not guarantee that the security requirements are fulfilled. Certain behavior beyond the scope of the security mechanisms may violate a security requirement in an unexpected way. In addition, the translation of security requirements into design can be very complex. It is often unclear which security requirements depend on each other and what purpose they serve.

In contextualization, the scope of a security requirement determines the application range of the security measures implemented or the extent of the restrictions put on the functionalities of the system. The rationale of a security requirement determines why a security requirement is important and how it interacts with the other security requirements of a given user. Similarly, the rationale of security requirements allows the designers of a system to keep in mind the perspectives of the users on the security requirements. The following is the rationale for security goal PhD.3 of the hidden meeting of the PhD meetings for the security role attendee of hidden meeting. The security goal is directed at the organizer of the hidden PhD meeting:

Not being able to attend a meeting using an anonymous pseudonym is a restriction of the security goals of any user of Intranet. In the hidden PhD meeting one way of strengthening confidentiality

and unobservability is keeping a strict list of the users informed of the hidden PhD meeting. Since the attendees of the hidden PhD meeting have an interest in keeping their meeting hidden, they have the security goal attributed at the Organizer that PhD students shall not be invited and therefore shall not attend the hidden PhD meeting using an anonymous pseudonym.

It is possible for users to be invited to hidden meetings using all other forms of pseudonyms, as long as they can be traced back to the identity of the user by the organizer. The security goal PhD.3 can be reformulated to express this:

PhD.3a an organizer can invite users to a hidden PhD meeting only using a public or private personal pseudonym but not an anonymous pseudonym.

The next task is to refine these security goals by studying them as a whole, in order to finalize the security requirements elicitation process.

4.4 Security requirements elicitation

The last step of our method is to compose the security goals of the variations into the security requirements of the system and is depicted in the circle in Figure 1. There are several ways of achieving this:

Combine Episodes: The requirements team studies the possible combinations of episodes, and examines how the security goals articulated for the episodes interact. Certainly conflicts and redundancies will arise. In addition, it may be necessary to state global security goals that apply to combinations of episodes. Processes for refining security goals in order to deal with these tasks will be defined.

Combine Roles: The requirements team investigates possible combinations of roles. They identify which functional roles, security roles, and social roles may be combined. Conflicting, redundant and missing security goals in the combination of roles will be refined.

Once these refinements are completed, a final set of security goals are summed up and organized as the security requirements document.

5 Related work

The security requirements elicitation method in this paper is based on existing studies of multilateral security and security requirements engineering. The method CREE, for *Confidentiality Requirements Elicitation and Engineering* [GJO+05], specializes the method presented here. It focuses on confidentiality requirements and employs use cases as a means to describe functional requirements. CREE puts less emphasis on variating episodes to come up with confidentiality requirements. It has been applied to the *Technology Assisted Practice Application Suite* (TAPAS), which is part of the Canadian National Health Infrastructure.

Security requirements engineering has over the years proven to be a challenging task. This is especially the case because pinpointing what security requirements are has been difficult. Despite all these challenges, the demand for developing security requirements elicitation methods for the changing requirements of networked environments is great. We shall now review some of the approaches that have been developed in the last few years.

Fontaine [Fon01] develops a formal framework for security requirements elaboration based on the goal oriented requirements elicitation defined by van Lamswerde [vL00]. He integrates two formal methods, the Ponder policy-based framework and the KAOS goal-oriented framework to integrate security requirements into the functional requirements process. Fontaine only considers high-level (global) security goals for the system after defining the functional requirements of the system. He adds that since the security goals are global and not local they cannot be assigned to particular agents. Different to our method, Fontaine first assumes that security goal conflicts only arise through obstacles or threats introduced by outside attackers of the system. Second, he assumes that agents of the system do not have conflicting security goals, since they all have the common high-level security goals.

Moffett et. al. [MN03] specify the relationship between security requirements and the specification of software behavior using Michael Jackson's problem frames. They suggest a method in which requirements and design are not separated. The authors state that security needs to be considered in every relevant domain –be it the machine, social, or any further domain relevant to the system. For this, they suggest a Security Requirements Framework. Through the utilization of problem frames in the framework it is possible to analyze multiple domains and elicit a complete list of security goals and requirements. Nevertheless, the framework lacks a method for eliciting the conflicting security requirements of the stakeholders of the system.

Lui et al. [LYM03] emphasize the social aspects of security and privacy requirements, and consider security and privacy concerns in the design of an agent based health information system as a case study. They propose a framework based on the agent-oriented requirements modeling language i*. They use this framework to clarify the relationships among stakeholders and, in particular, to find vulnerabilities in their organizational relationships. Common to our approach, Lui et al. point out the need to analyze the social context of stakeholders. But where we emphasize the multilateral point of view that each stakeholder has his or her own legitimate interests in the system, Lui et al. focus on attacker analysis, and ways a possible attacker might exploit vulnerabilities to compromise security. Thus, their analysis addresses the level of system requirements, where the different views of stakeholders have already been refined to a coherent set of requirements, and the decisions that are necessary when designing a secure system based on those requirements.

Similarly, Mouratidis et al. [MGM03] adopt the i* modeling framework for their methodology Tropos. Tropos deals with non-functional requirements, including security requirements, throughout system requirements analysis, system design, and implementation in a homogenous way. The method analyses dependencies between actors, goals and tasks just as in Lui et al. [LYM03] and mention that in the early requirements phase the security goals of the actors should be analyzed. Nevertheless, the authors offer no tools or descriptions of how this can be done.

6 Conclusion

The proposed method attends to some of the gaps still existing in the different security requirement elicitation methods. Both the problem of the contextualization of the end-users security goals, as well as the elicitation of security requirements together with functional requirements are attended to.

The task of identifying the security requirements of the stakeholders of a system is complex. By introducing helping concepts like roles and episodes, we propose a systematic method to manage this complexity. In this paper we illustrate and validate our method with the Intranet example. We identify the elicitation of security requirements through composition of security goals, episodes and roles as issues for future research. Finally, we are looking forward to developing a graphical notation which allows for a precise and simple modeling of security requirements.

References

[AE01] A. I. Antón and J. B. Earp. Strategies for Developing Policies and Requirements for Secure E-Commerce Systems. In A.K. Ghosh, editor, *Recent Advances in E-Commerce Security and Privacy*, pages 29–46. Kluwer Academic Publishers, 2001.

[Fon01] P. Fontaine. *Goal-Oriented Elaboration of Security Requirements*. PhD thesis, Departement D'Ingenierie Informatique, Université Catholique de Louvain, 2001.

[GJO+05] S. Gürses, J. H. Jahnke, C. Obry, A. Onabajo, T. Santen, and M. Price. Eliciting Confidentiality Requirements in Practice. In *Proc. 15th Annual International Conference hosted by the IBM Centers for Advanced Studies (CASCON 2005)*. IBM, Canada, 2005.

[LYM03] L. Liu, E. Yu, and J. Mylopoulos. Security and privacy requirements analysis within a social setting. In *Proc. 11th IEEE Requirements Engineering Conference*, pages 151–161. IEEE Press, 2003.

[MGM03] Haralambos Mouratidis, Paolo Giorgini, and Gordon Manson. Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In *CAiSE 2003, LNCS 2681*, pages 63–78. Springer Verlag, 2003.

[MN03] J. Moffett and B. Nuseibeh. A Framework for Security Requirements Engineering. Technical Report YCS 368, Dept of Computer Science, University of York, 2003.

[RPM99] K. Rannenberg, A. Pfitzmann, and G. Müller. IT Security and Multilateral Security. In G. Müller and K. Rannenberg, editors, *Multilateral Security in Communications – Technology, Infrastructure, Economy*, pages 21–29. Addison-Wesley, 1999.

[vL00] A. van Lamsweerde. Requirements Engineering in the Year 00: A Research Perspective. In *Proc. 22nd International Conference on Software Engineering*, pages 5–19. ACM Press, 2000.

[Wol01] G. Wolf. *Characteristika von und Wechselwirkungen zwischen Schutzzielen sowie deren Umsetzung in einer Benutzungsschnittstelle*. PhD thesis, Fakultät Informatik, TU Dresden, 2001.