KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Privacy Threats in Software Architectures

Kim Wuyts

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

January 2015

Privacy Threats in Software Architectures

Kim WUYTS

Examination committee:
Prof. dr. A. Bultheel, chair
Prof. dr. ir. W. Joosen, supervisor
Prof. dr. ir. R. Scandariato, co-supervisor
Prof. dr. D. Clarke
Prof. dr. ir. B. Preneel
Prof. dr. ir. E. Steegmans
Prof. dr. B. Berendt
Prof. dr. ir. A. Rashid
(Lancaster University, UK)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

January 2015

ISBN 978-94-6018-950-0
D/2015/7515/6

Acknowledgements

While this is the last section to write, this will be the first (and possibly only) section you read. Since I have your attention, I would like to take this opportunity to express my gratitude towards all the people that helped me obtain this PhD.

Wouter, thank you for giving me the opportunity to work at DistriNet and pursue a PhD. I am really grateful for your believe in me and my work.

Riccardo, thank you for your day-to-day guidance, feedback and motivational talks. You helped me grow from the hesitant junior to the independent researcher I am today.

I also would like to thank the members of the jury for helping me put my work in perspective. Your elaborate feedback did not only help me improve my text, but also triggered some new ideas for future LINDDUN research.

I am also thankful for all the fruitful collaborations which lead me to new and interesting insights. Mina, thanks for the interesting discussions that resulted in the creation of LINDDUN. Bart, thank you for your guidance in the creation of the original privacy taxonomy. Also, thank you, Griet, for sharing your legal expertise. I really enjoyed the work we did together. I would also like to thank Seda. Even though our collaboration has not (yet) resulted in a publication, I found our discussions and your feedback truly inspirational. Also, I am grateful to all the people who participated in our empirical studies.

I would also like to thanks all the (ex-)colleagues whom I crossed paths with in the past eight years.
I always enjoyed the talks with my office mates. I especially want to thank

Philippe, Koen, and Thomas for the interesting work-related discussions and the fun informal chats.
I am also grateful for the many inspirational discussions and fruitful collaborations, as well as the pleasant talks during coffee and lunch breaks, and informal chitchats in the hallway. Thank you: Alex, Aram, Bart E, Bart VB, Bert, Christophe, Davy, Dimitri, Dominique, Dries, Eric, Jef, Klaas, Koen B, Koen Y, Kristof V, Laurens, Lieven, Maarten, Mario, Marko, Milica, Nelson, Phil, Philippe, Pieter, Rafa, Raoul, Rula, Rutger, Sam, Stefan VB, Stefan W, Steven O, Steven V, Thomas H, Tom G, and Wouter DB.
I want to add a special thanks to Annick, Ghita, and Katrien for their help with all the practical issues I encountered throughout the years (from booking meetings and assisting in administrative tasks to cleaning up abandoned wine bottles), and for the fun girl talks. Also, thank you, Marleen, Esther, and Karen, for taking care of all my administrative questions with a smile.
Also, I cannot forget to thank my fellow 'networking experts', who helped me close many receptions: Dries, Jef, Koen Y, Kristof V, Stefan W, and Steven V.

Of course, I would also like to thank my family and friends for their support throughout the years.
I am especially grateful to my parents for their continuous motivation and care (Thanks for proofreading my text, Mom!), the '*light*' lunches and dinners during the busy months of writing a thesis and building a house, and their genuine interest in my thesis and all related practicalities (No, Mom, I will not just wear jeans but actual 'grown-up' clothes. Yes, Dad, there will be a reception and you are also invited.).
I would also like to add a special thanks to my parents-in-law for their support. Also thank you for inviting us to have our 'vacation' with you in the Provence. Being able to reward myself after a morning of writing with a refreshing splash in the pool, made it my most productive, or at least my most pleasant, week of thesis writing.
Also thanks to my old *Informatica* study buddies, Birgen, Jelle, Jo, Tias, Pieter, and Steven for the many IT-related discussions but even more so for the friendship.
And a special mention to Inge. Thanks for your endless enthusiasm and interest, and for introducing me to all the do's and don'ts of commuting to Leuven.
And finally, last but not least, thank you, Frederik ('Fredje'), for all your patience and support during the past years, your understanding the past months when our vacation was once more delayed due to my changing deadlines, and the peptalks during the past weeks. Bedankt!

Abstract

With privacy becoming a key concern in modern society, it is important that privacy measures are strongly incorporated whenever digital data are involved. Unfortunately, privacy is often neglected when engineering software systems and only introduced as an afterthought. Retrofitting privacy concerns in an existing system is however not straightforward. In recent years, a different attitude towards privacy has emerged, which is known as '*Privacy by Design*.' One of its core principles states that privacy should be embedded in the early stages of the software development lifecycle, rather than having it as an add-on. Hence, privacy should become an essential component in the software development process.

This thesis adheres to the Privacy by Design paradigm as it proposes and validates LINDDUN, a privacy threat modeling methodology that helps software engineers with limited privacy expertise to introduce privacy early on in the software development lifecycle. LINDDUN is a model-based approach that leverages a data flow diagram (DFD) as representation of the system to be analyzed. This DFD will serve as basis for the analysis, as each of its elements is systematically examined thoroughly for privacy threats. The methodology is also knowledge-based. It provides an overview of the most common attack paths associated with the set of privacy threat categories contained in the acronym LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance). The attack paths are represented as threat trees that detail possible causes of threats. Each tree presents the attack paths that are related to one threat category which is applied to one particular DFD element type (entity, data flow, data store, or process).

LINDDUN is not a single off-the-shelf technique that you can plug-and-play. It is a reasonably complex methodology and its success is affected by the interaction between the analyst and the methodology itself. Therefore, in order to validate LINDDUN and its applicability, we did not merely look at the methodology itself, but we also investigated this human-methodology dependency and reflected upon it. We performed a multi-faceted, empirical validation of LINDDUN comprising three studies involving analysts of different seniority to evaluate its ease of use and overall performance.

As a final contribution, we enhanced the LINDDUN methodology based on the empirical results. The main changes include an update and extension of the threat trees and their descriptions, and a repositioning of the security threats.

Beknopte samenvatting

Privacy is een kostbaar goed geworden in onze moderne samenleving. Het is dan ook uitermate belangrijk dat de nodige privacy maatregelen genomen worden worden om digitale data te beschermen. Jammer genoeg wordt privacy nog te vaak verwaarloosd tijdens het software ontwikkelingsproces waardoor het slechts achteraf in overweging wordt genomen. De laatste jaren is er gelukkig een nieuwe houding tegenover privacy ontstaan, die bekend staat als 'Privacy-door-Ontwerp' (ook wel 'Privacy by Design' genoemd). Een van de basisprincipes hiervan stelt dat privacy reeds moet ingebed worden in de eerste fases van de ontwikkelingscyclus. Privacy hoort dus een essentiële bouwsteen te worden in het software ontwikkelingsproces.

Dit proefschrift volgt het Privacy by Design paradigma en stelt LINDDUN voor, een methode die privacybedreigingen modelleert om software-ontwikkelaars met beperkte privacy expertise te helpen bij het introduceren van privacy in de eerste fasen van het ontwikkelingsproces. LINDDUN is een modelgebaseerde aanpak die start van een dataflowdiagramma (DFD) als representatiemodel van de gegevensstroom van het te analyseren systeem. Deze DFD legt de basis voor de analyse aangezien elk van zijn elementen uitgebreid zal onderzocht worden op privacybedreigingen. De methode biedt ook privacykennis aan. Zo is er een catalogus die de meest voorkomende aanvalspaden beschrijft die geassocieerd worden met de verzameling privacybedreigingscategorieën waarvan de Engelse termen omvat zitten in het acroniem LINDDUN (koppelbaarheid, identificeerbaarheid, onweerlegbaarheid, detecteerbaarheid, vrijgave van informatie, onwetendheid, en non-conformiteit). Deze aanvalspaden worden weergegeven als bedreigingsbomen die de mogelijke oorzaken van bedreigingen omschrijven voor elk van de bedreigingscategorieën gekoppeld aan

een specifiek elementtype van de DFD (entiteit, gegevensstroom, gegevensopslag of proces).

LINDDUN is geen eenvoudige techniek die volledig geautomatiseerd kan worden. Het is een vrij complexe methode waarvan het succes afhankelijk is van de interactie tussen de analist en de methode. Om LINDDUN en zijn toepasbaarheid te valideren, hebben we dus niet enkel de methode zelf bekeken, maar hebben we ook de wisselwerking tussen de mens en de methodiek bestudeerd. We hebben een veelzijdige empirische validatie van LINDDUN uitgevoerd die bestond uit drie studies met medewerking van analisten met een uiteenlopende anciënniteit om het gebruiksgemak en de algemene prestaties van de methode te evalueren.

Als laatste contributie, hebben we de LINDDUN methode verbeterd op basis van de empirische resultaten. De voornaamste wijzigingen bestaan uit een actualisering en uitbreiding van de bedreigingsbomen en hun beschrijving, en een herpositionering van de beveiligingsbedreigingen.

Abbreviations

AOP	Aspect Oriented Programming
BPMN	Business Process Modeling Notation
CORAS	Conducting security Risk Analysis
DFD	Data Flow Diagram
DPD	Data Protection Directive
DREAD	Damage, Reproducibility, Exploitability, Affected users, Discoverability (mnemonic for risk rating security threats)
EHR	Electronic Health Record
eRISE	European Risk and Security Requirements Engineering
ERM	Entity-Relationship Model
FIPPs	Fair Information Practice Principles
FN	False Negative
FP	False Positive
FPFSD	Framework for Privacy-Friendly System Design
ICT	Information and Communication Technology
IOI	Item of Interest
IoT	Internet of Things

KAOS	Keep All Objectives Satisfied (goal-oriented requirements engineering)
LINDDUN	Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, and Non-compliance (privacy threat modeling)
MAC	Message Authentication Code
MPRA	Multilateral Privacy Requirements Analysis
MUC	Misuse Case
NIST	National Institute of Standards and Technology
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
OTR	Off-The-Record messaging
OWASP	Open Web Application Security Project
PbD	Privacy by Design
PET	Privacy Enhancing Technology
PII	Personal Identifiable Information
PRET	Privacy Requirements Elicitation Technique
PriS	Incorporating Privacy Requirements into the System Design Process
PRoPAN	Problem-based Privacy Analysis
QTMM	Quantitative Threat Modeling Methodology
ReMeS	Remote Measurement, Monitoring, and Control System
RQ	Research Question
SDL	Security Development Lifecycle
SEI	Software Engineering Institute
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, and Elevation of privilege (Security threat modeling)
TP	True Positive
UML	Unified Modeling Language

Contents

1

Introduction

"We would have lived our lives differently if we had known they would one day be searchable."

The above is a strong quote from Alexia Tsotsis [Tso10], an influential blogger, which, although slightly exaggerated, clearly sets the scene of today's society. As personal information becomes ubiquitous, privacy is a key issue nowadays. Data that used to be stored offline (or even on paper) are now part of a highly-integrated e-society: health, financial and social data are only one mouse-click away. This abundance of digital information can however easily lead to privacy issues. Cases of information leaks have been in the news regularly (e.g. credit card data stolen from the Playstation network [BF11], unprotected customer data at the Belgian Railways [nmb13], etc.), and even more alarming were the revelations on government mass surveillance.

With privacy becoming such a key concern in modern society, it is highly important that privacy measures are strongly incorporated whenever digital data are involved. All systems that handle data - whether it concerns collecting, processing or sharing - should invest in procedures to ensure the privacy of the data subjects involved. Unfortunately, privacy is often still neglected when engineering software systems. It is sometimes introduced as an

afterthought. Retrofitting privacy concerns in an existing system is however not straightforward. Design and implementation decisions will impact and often conflict with the privacy requirements, which makes the resolution of privacy issues even more challenging.

In recent years, a different attitude towards privacy is emerging, which is known as '*Privacy by Design*' [Cav09, vRGB+95]. The concept has been introduced in the late 90's by Ann Cavoukian, a Canadian Information & Privacy Commissioner. One of its core principles states that privacy should be embedded in the early stages of the software development lifecycle, rather than having it as an add-on. Even though techniques, such as aspect oriented programming (AOP) [KLM+97], can resolve certain issues later on, existing privacy-related AOP solutions are limited. And, more importantly, privacy concerns can already impact the architectural design. This makes it imperative that privacy becomes an essential component in the software development process. This concept has been slowly gaining importance. Nowadays, privacy experts agree that Privacy by Design should be the norm [SC09, LSK12, hor14].

This thesis also adheres to the Privacy by Design paradigm as it proposes and validates LINDDUN, a privacy threat modeling methodology that aids the software analyst to introduce privacy early on in the software development lifecycle.

1.1 Privacy by Design using LINDDUN

Although Privacy by Design is an emerging domain, supporting techniques are lacking.

Extensive research exists regarding privacy solutions; for example, anonymous communication systems such as mix-networks [Cha81] and onion routing [GRS96] were already developed several decades ago. Also privacy properties, such as anonymity, unlinkability and unobservability, have been documented in detail by Pfitzmann and Hansen [PH10]. Unfortunately, methods to incorporate these properties into the software development lifecycle are missing. Either existing approaches include most of the core properties but lack an actual elicitation process [KKG08]. Or, generic requirements engineering methods, such as i* [Yu97] and KAOS [VL09], can be used but they do not provide any privacy-specific support. Alternatively, several methodologies focus on data protection legislation [Jen05, MMZ08, SC09, YC02] by translating regulatory rules into software requirements that are easier to grasp and implement. The main goal of these methodologies is however to obtain legal compliance. Data protection legislation does contain privacy guidelines, but these are too abstract

and miss some of the core privacy properties (as for example described by Pfitzmann [PH10]).

A *first contribution* of this thesis is a new privacy threat modeling methodology that helps software engineers with limited privacy expertise to introduce privacy early on in the development lifecycle. Determining appropriate privacy requirements to be fulfilled by a specific software system is far from trivial. People tend to find it hard to formulate precise privacy demands. It is in fact easier to reason about which privacy violations should be prevented. Indeed, for example, patients participating in a medical trial will be more likely to state that they do not want their identity revealed to the researchers than requiring a data minimization technique that generalizes the identifiers of the collected medical data to ensure anonymity (such as k-anonymity [Swe02b]). Threat modeling is a structured approach that enables the analyst to identify and address the privacy flaws within an application. Threat modeling is a key activity to elicit the potential pitfalls of a software system. It is mainly used in security analysis where it is considered one of the pillars in the construction of a secure system. In the context of security, Microsoft's STRIDE is a well-known, mature technique that is used by professionals in order to discover the security threats of software systems [HL06]. In previous work, we have put STRIDE to the test of empirical investigation and we have characterized its strength and efficiency [SWJ13]. With reference to privacy, however, a similar strong contender was missing in the threat modeling area. Therefore, we embarked on the creation of a privacy-focused threat modeling technique called LINDDUN [DWS+11]. It is a knowledge-based methodology that provides a repository of the most common attack paths associated with a set of privacy threat categories contained in the acronym LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance). LINDDUN is markedly inspired by STRIDE and is complementary to it. For instance, both techniques are model-based[1] as they start from a representation of the system-under-investigation that is described as a data flow diagram. Following the Twin Peaks approach [Nus01], this model can be constructed based on the requirements specification or the architectural diagram. Clearly, the level of abstraction influences the granularity of the model and its associated results.

[1] We use the term *model-based* to indicate that the methodology depends on a graphical representation (or model) as basis for the analysis. This should not be confused with model-driven development.

1.1.1 LINDDUN Threat Modeling

LINDDUN encourages analysts to think about privacy concerns in a systematic way. The methodology consists of three problem-oriented steps to elicit privacy threats and three more solution-oriented steps that aid at translating the elicited threats into privacy enhancing solutions.

Problem-oriented LINDDUN Steps. These first three steps of the LINDDUN threat modeling methodology are considered the core steps as they aid the analyst to elicit privacy threats. In the first step, LINDDUN leverages a data flow diagram (DFD) as representation of the system to be analyzed. This DFD will serve as basis for the analysis, as each of its elements is systematically examined for privacy threats. The creation of the DFD should not be considered a major overhead, as there is an easy transition from more mainstream UML models (such as the component-connector model) to the corresponding DFD. The second step maps each of the DFD elements to their corresponding LINDDUN threat categories and thereby identifies potential threats. The third step examines all potential threats to determine which threats are in fact applicable. To aid the analyst, LINDDUN provides an overview of the most common attack paths associated with the set of privacy threat categories contained in the acronym LINDDUN. The attack paths are represented as threat trees that detail possible causes of threats. Each tree contains the attack paths that are related to one specific threat category, which is applied to one particular DFD element type (entity, data flow, data store, or process). This privacy knowledge base supports and encourages also non-privacy-experts to reflect on potential privacy concerns. Each of the nodes in the threat tree that is applicable to a specific DFD element is a viable threat and should be documented using a misuse case template (i.e. a use case from the perspective of the attacker). Note that documenting threats is not necessarily limited to this step. On the contrary, privacy threats can already emerge earlier on in the software development process. Evidently, these threats should be documented as soon as they surface. Also, given the nature of the Twin Peaks approach, initial high-level privacy threats can be extended with more detailed threats in a subsequent iteration of the development cycle.

Solution-oriented LINDDUN Steps. The three remaining steps guide the analyst into the solution space. In the fourth step, the analyst should prioritize the elicited threats based on their risk level. LINDDUN does not restrict the analyst to a dedicated risk assessment method. It refers to established risk analysis techniques for more details [FKG+02, CMU, OWA]. The analyst can thus plug in the assessment method of his choice. The two final steps aid in the selection of appropriate privacy enhancing technologies to mitigate the elicited

threats. In the fifth step, the threats' corresponding privacy requirements (in the original LINDDUN methodology) or privacy mitigation strategies (in the improved LINDDUN methodology) are elicited, which enable targeted selection of matching privacy enhancing solutions in the sixth and final step.

The LINDDUN methodology is getting more attention in the research community. It has been successfully applied by independent researchers [HBQ12, Bec12, KLK+14, ML14], and has been discussed in the Threat Modeling book by Shostack [Sho14], who is considered a leader in the field.

1.2 Empirical Validation of LINDDUN

LINDDUN is not a single off-the-shelf technique that you can plug-and-play. It is a rather complex methodology and its success is affected by the interaction between the analyst and the methodology itself. The applicability of each threat should be interpreted. For example, identifiability of a user is a severe threat for online voting, while the goal of social networking systems such as Facebook is precisely to share personal identifiable information. Similarly, assumptions about specific parts of the system require human reasoning. Determining, for example, whether the internal communication can be considered secure or is susceptible to insider threats cannot be fully automated.

In order to validate LINDDUN and its applicability, we thus do not merely need to look at the methodology itself, but we need to investigate this human-methodology dependency and reflect upon it. For our *second contribution*, we therefore conducted two descriptive studies to empirically validate the LINDDUN methodology and evaluate its ease of use and its overall performance (completeness, correctness, and productivity). A final study was performed to determine in which areas LINDDUN could still be improved when compared to privacy experts. Note that these studies specifically focus on the elicitation of privacy threats, and therefore only examine the first three problem-oriented steps of the LINDDUN methodology.

1.2.1 Approach

In this thesis we present a multi-faceted, empirical evaluation of LINDDUN comprising 3 studies. This is a novel approach, as empirical validation has not yet been applied that often in security and privacy research.

Threat modeling can be applied at different levels of abstraction. Depending on the definition of the assets that are being analyzed, it can be performed either

early (i.e. at the requirements phase) or later (i.e. at the architecture stage). In this respect, threat modeling is an essential activity for both requirements engineers and software architects. Therefore, in this thesis we take the stance of both roles.

In the *first study*, we asked 8 participants to analyze the *requirements* of a software system. The participants received a description of the system domain and of the functionality the system had to support. They used LINDDUN to discover the main privacy threats the system should prevent.

In the *second study*, we asked 54 participants to analyze the *architecture* of a software system. The participants received the blueprint of the system design, including several architectural diagrams. They used LINDDUN to discover the main privacy weaknesses in the design. Next to the different perspective (requirements vs. architecture), size is also a differentiator in the two studies as the first study is decidedly smaller. In general, a larger size is to be preferred from an empirical perspective, because the conclusions of a larger study are statistically stronger. Nevertheless, a smaller study provides the opportunity to monitor the activities of the participants more closely. This turns into a more insightful understanding of the dynamics that led to the observed results.

In both cases, we used the empirical technique of *descriptive studies*. Contrary to a controlled experiment, in a descriptive study a technique is characterized as such, rather than compared to another technique. A descriptive study is instrumental in order to understand a technique and eventually formulate research hypotheses to be further investigated by means of comparative experiments. For instance, in the two descriptive studies we observed that LINDDUN provides a good level of guidance to the participants. For example, as described later, LINDDUN provides an extensive catalog of privacy threats that the participants can use as inspiration during their analysis.

However, we also observed that this guidance has the tendency to limit the creative thinking of the participants. The participants do not discover threats that are not mentioned in the catalog. Therefore, the reliability of LINDDUN (in terms of coverage of threat space) is a key aspect to investigate. We addressed this aspect in the *third empirical study* presented in this thesis. We have conducted a case study, which is an empirical technique to compare different treatments. Specifically, we compared the threats identified by LINDDUN to those independently obtained by a small panel of 3 privacy experts.

1.2.2 Research Questions

In particular, we set out to answer the following research questions during the evaluation:

- *RQ1 - Correctness.* How many threats discovered by LINDDUN are correct?
- *RQ2 - Completeness.* How many potential threats go undetected?
- *RQ3 - Productivity.* How many valid threats are identified in a given time frame?
- *RQ4 - Ease of use.* Do analysts perceive the methodology as easy to learn and apply?
- *RQ5 - Reliability.* How do LINDDUN results compare to those of privacy experts?

The first four research questions will be examined by the first and the second study. Research question 5 will be evaluated during the third empirical study.

The evaluation in this thesis focuses mainly on the quantitative aspects of LINDDUN. While the qualitative aspects are evidently also interesting, we explicitly aimed for a quantitative analysis. First of all, we did not find other privacy approaches that are comparable to our methodology. Second, we want to understand all the positive and negative aspects of the methodology itself and use these quantitative results to improve LINDDUN. In future work, we want to evaluate this streamlined version in industry and focus on the more qualitative aspects of the methodology.

1.3 Evolution of LINDDUN

Although the overall results of the empirical evaluation were convincing, there was still room for improvement.

Based on the results of the empirical evaluation, we introduced some improvements in the LINDDUN methodology. As LINDDUN is a knowledge-based approach, it provides a threat tree catalog to support the elicitation of threats. The participants indicated that this catalog was helpful but could benefit from a more detailed explanation of the individual threats. Also, the reliability study showed that some privacy threat categories were insufficiently

covered by LINDDUN. We therefore extended and clarified even further the privacy knowledge the methodology provides. In addition, we added more structure in the methodology itself and its documentation to increase the usability even further.

1.4 Contributions

In summary, the contribution of this thesis is threefold.

1. We proposed *LINDDUN, a privacy threat modeling technique.* To the best of our knowledge, LINDDUN is the first technique to elicit threats that focus on privacy. The creation of LINDDUN was joined work with Mina Deng [DWS+11].
2. As all new techniques require a form of validation, our second contribution consists of the *thorough evaluation of the LINDDUN methodology* by means of three empirical studies. The evaluation takes into account both the perspective of requirements engineers and software architects.
3. Based on the results of the evaluation, several *improvements have been applied to the LINDDUN methodology* to increase its correctness, completeness, productivity, reliability and ease of use.

1.5 Overview

Chapter 2 presents background information related to this thesis. It will first introduce privacy concepts and definitions to position LINDDUN and provide information on the privacy concepts the methodology uses. Second, Privacy by Design principles are explained together with existing privacy requirements and threat modeling techniques.

The LINDDUN methodology itself is highlighted in Chapter 3. Each of the 6 methodology steps are detailed and the chapter also includes an overview of all privacy threat trees that support the elicitation process.

The main part of this thesis contains the empirical evaluation of LINDDUN. Chapter 4 describes the evaluation of our methodology when applied at requirements engineering level. Chapter 5 defines the evaluation of the methodology at architectural level. This chapter actually consists of two studies: evaluating the general performance of LINDDUN when applied on a

software architecture, and evaluating the reliability of LINDDUN by comparing its results with those of privacy experts.

Based on the results of these empirical studies, we have introduced some improvements to the LINDDUN methodology, as presented in Chapter 6.

Chapter 7 concludes this thesis with a summary of the contributions and a provides an outlook onto future work.

2

Background and Related Work

This chapter sets the scene for the remainder of the thesis. As privacy is a broad concept, the following sections scope the privacy landscape and set out the context in which our work is established.

First, the general privacy concepts that are used throughout the thesis are introduced. To conclude, existing privacy requirements techniques and a background on threat modeling are presented.

2.1 Introduction to Privacy

The concept of privacy differs among different communities. This section provides a summary of the definitions and vocabulary that are used as background for our work. Furthermore, this section presents a set of taxonomies that aim to structure the field.

2.1.1 Privacy Definitions

Solove [Sol06] accurately summarized the problem of privacy: "Privacy is a concept in disarray. Nobody can articulate what it means. As one commentator has observed, privacy suffers from an embarrassment of meanings."

Indeed, there exists an abundance of definitions trying to scope the concept of 'privacy.' As there still is no single widely accepted definition, this section summarizes the most referenced definitions.

Warren and Brandeis are considered the first authors of a publication detailing the right to privacy. They define privacy as *"the right to be let alone"* [WB90]. Although the definition originally aimed at the protection of individuals against gossip and slander, through time it has gained a wider meaning. Digital privacy researchers have been interpreting the definition as "an autonomous (digital) sphere in which the data about persons are protected so that unauthorized others cannot access it, also known as data confidentiality" [Gür10]. In this interpretation of privacy, it is hence important to avoid making personal data (i.e. data relating to an identifiable person) available to a larger audience.

Westin, on the other hand, defines privacy as "*the right of the individual to decide what information about himself should be communicated to others and under what circumstances*" [Wes70]. This definition relates privacy to the right to control the information that is revealed to others. This concept is often referred to as information self-determination [Gür10, RP09], as the user himself is able to determine which information he is sharing when and how.

Related to this idea of self-determination is the concept of identity construction. A third definition therefore states that privacy is "*freedom from unreasonable constraints on the construction of one's own identity*" [Agr99]. In the offline world, people will share different information with different people. One will, for example, share different stories with friends, parents, or acquaintances. In the online world, this phenomenon is similar: individuals will want to reveal different information in different contexts.

Nissenbaum [Nis04] introduced '*contextual integrity*' as an alternative benchmark for privacy. This approach requires adequate protection for privacy to norms of specific contexts, demanding that information gathering and dissemination be appropriate to that context and obey the governing norms of distribution within it.

2.1.2 Privacy Vocabulary

This section highlights the privacy-related vocabulary that is used throughout the thesis. We do not intend to propose any definitions ourselves, but we summarize the terms as they are described by the European Data Protection Directive (DPD)[Eur95]. The Directive regulates the protection of individuals with regard to the processing of personal data.

First of all, it is important to note that data protection legislation in particular, but also privacy research in general, is mainly concerned about protecting *personal data* (sometimes also referred to as personal identifiable information or PII). General facts (e.g. the sky is blue) do not require special privacy preservation as they are considered general knowledge. Privacy is person-bound and concerns the protection of data that can be linked to a certain individual (or set of individuals). Personal identifiable information has been defined as "any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity" [Eur95]. Thus, according to legislation, information does not require privacy protection when it has been anonymized. Note however that in this era of big data, full anonymity is hard, if not impossible [Tuc13]. De-identification (removing pseudo-identifiers) will simply not suffice to anonymize information, and even more advanced anonymity techniques cannot guarantee full anonymity when data are linkable. A more extensive privacy protection will hence be essential.

Another privacy concept related to data processing is '*data subject*.' This refers to the individual that is linked to the personal identifiable information. This is not necessarily the creator of the information. In e-health applications, for example, patient data are created and managed by physicians while the data actually involve a particular patient (i.e. data subject).

2.1.3 Privacy Properties

To have a solid base for the types of privacy threats handled by the proposed LINDDUN framework, we have elaborately studied definitions of privacy properties. In this section, we discuss the set of properties that sets the scope of the privacy domain in this thesis. Most privacy properties in the LINDDUN framework comply with the terminology proposed by Pfitzmann and Hansen [PH10], as it is widely recognized in the privacy research community.

Privacy is sometimes distinguished as *hard privacy* and *soft privacy* [Dan12]. Hard privacy relates to *data minimization*, and is based on the assumption that personal data is not divulged to third parties. The system model of hard privacy is that a data subject provides as little data as possible in order to reduce the need to "trust" other entities. The data subject himself will be the active security party (i.e. is responsible for hiding information).

Soft privacy, on the other hand, is based on the assumption that the data subject has already lost control of his personal data and has to trust the honesty and competence of the data controllers. Soft privacy is mainly achieved by data security (e.g. access control) and regulatory implementations (e.g. policies, consents, purpose-based access and processing, audit, etc.). It will be up to the data controller (i.e. receiver of the information, in general this will be the back-end system) to ensure data protection. Unfortunately, this implies that it will be difficult for the data subject to verify how his data are collected and processed.

The following paragraphs enlist a number of privacy properties that are implemented by LINDDUN and thus are used throughout this thesis.

Unlinkability. Unlinkability refers to hiding the link between two or more actions, identities, and pieces of information. Examples of unlinkability include hiding links between two anonymous messages sent by the same person, two web page visits by the same user, entries in two databases related to the same person, or two people related by a friendship link in a social network.

Unlinkability is defined by Pfitzmann and Hansen. as [PH10]: *"Unlinkability of two or more items of interest (IOIs, e.g. , subjects, messages, actions, etc.) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not."*

Anonymity and Pseudonymity. Anonymity refers to hiding the link between an identity and an action or a piece of information. Examples are the anonymous sender of an email, writer of a text, person accessing a service, person to whom an entry in a database relates, and so on.

Anonymity is defined as: *"Anonymity of a subject from an attacker's perspective means that the attacker cannot sufficiently identify the subject within a set of subjects, the anonymity set"* [PH10]. Anonymity can also be described in terms of unlinkability. If one considers sending and receiving of messages as attributes; the items of interest (IOIs) are who has sent or received which message. Then, *"anonymity of a subject with respect to an attribute may be*

defined as unlinkability of this subject and this attribute." For instance, *sender anonymity* of a subject means that to this potentially sending subject, each message is unlinkable. Correspondingly, *recipient anonymity* of a subject means that to this potentially receiving subject, each message is unlinkable.

Pseudonymity suggests that it is possible to build a reputation on a pseudonym and use multiple pseudonyms for different purposes. Examples include a person publishing comments on social network sites under different pseudonyms and a person using a pseudonym to subscribe to a service.

Pfitzmann et al. [PH10] defines pseudonymity as: *"A pseudonym is an identifier of a subject other than one of the subject's real names. Pseudonymity is the use of pseudonyms as identifiers. A subject is pseudonymous if a pseudonym is used as identifier instead of one of its real names."* Pseudonymity can also be perceived with respect to linkability. Whereas anonymity and identifiability (or accountability) are the extremes with respect to linkability to subjects, pseudonymity is the entire field between and including these extremes. Thus, pseudonymity comprises all degrees of linkability to a subject.

Plausible Deniability. Plausible deniability refers to the ability to deny having performed an action that other parties can neither confirm nor contradict. Plausible deniability from an attacker's perspective means that an attacker cannot prove a user knows, has done or has said something. Sometimes, depending on the application, plausible deniability is desirable over non-repudiation (i.e. a security property to ensure accountability). For instance, in an application used by whistleblowers, users will want to deny having ever sent a certain message to protect their safety. Other examples include off-the-record conversations, the possibility to deny the existence of an encrypted file, deny that a file is transmitted from a data source, or deny that a database record belongs to a person.

The relation between non-repudiation and plausible deniability is described by Roe in [Roe97]: *"The goal of the non-repudiation service is to provide irrefutable evidence concerning the occurrence or non-occurrence of an event or action. If we believe that there is a need for this as a security service [...] we must also concede that some participants desire the opposite effect: that there be no irrefutable evidence concerning a disputed event or action."* This "complementary service" is plausible deniability.

In particular, it ensures that "an instance of communication between computer systems leaves behind no unequivocal evidence of its having taken place. Features of communications protocols that were seen as defects from the standpoint of non-repudiation can be seen as benefits from the standpoint of this converse

problem, which is called plausible deniability."

The security property 'non-repudiation' and the privacy property 'plausible deniability' are mutually exclusive. This should however not cause any conflicts, as systems will either require strong non-repudiation properties to ensure accountability, while others will require plausible deniability. For e-commerce applications, non-repudiation is an important security property. Imagine a situation where a buyer signs for a purchased item upon receipt, the vendor can later use the signed receipt as evidence that the user received the item. For other applications, such as off-the-record conversations or online voting, participants may desire plausible deniability for privacy protection such that there will be no record to demonstrate the communication event, the participants and the content. In this scenario, non-repudiation is a privacy threat.

Undetectability and Unobservability. Undetectability and unobservability refer to hiding the user's activities. It is, for example, impossible to know whether an entry in a database corresponds to a real person, or to distinguish whether someone or no one is in a given location.

Undetectability is defined as: "*Undetectability of an item of interest (IOI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not. If we consider messages as IOIs, this means that messages are not sufficiently discernible from, e.g. , random noise"* [PH10]. For anonymity and unlinkability, not the IOI, but only its relationship to the subject or other IOIs is protected. For undetectability, the IOIs are protected as such.

Unobservability is defined as: "*Unobservability of an item of interest (IOI) means undetectability of the IOI against all subjects uninvolved in it and anonymity of the subject(s) involved in the IOI even against the other subject(s) involved in that IOI*" [PH10]. The definition suggests that unobservability is undetectability by uninvolved subjects combined with anonymity even if IOIs can be detected. Consequently, unobservability implies anonymity, and unobservability implies undetectability. It means, with respect to the same attacker, unobservability reveals always only a subset of the information anonymity reveals. In later sections, we focus on undetectability, since unobservability is in fact a combination of undetectability and anonymity.

Confidentiality. Confidentiality refers to hiding the data content or controlled release of data content. Examples include transferring encrypted email, applying access control to a classified document or a database containing sensitive information.

NIST [MGK09] describes confidentiality as follows: *Confidentiality means preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.* Although confidentiality is a security property, as the definition above states, it is also important for preserving privacy properties, such as anonymity and unlinkability. Therefore, confidentiality is also considered an important privacy objective.

Awareness. Although the aforementioned privacy properties are often considered as the core properties, we also consider the following two properties, namely content awareness, and policy and consent compliance, important privacy objectives due to their significance to privacy and data protection. With the emerging of Web 2.0 technologies, users tend to provide excessive information to service providers and lose control of their personal information. Therefore, the awareness property is proposed to make sure that users are aware of their personal data and that only the minimum necessary information should be sought and used to allow for the performance of the function to which it relates.

Awareness is a rather broad concept. Endsley [End95] defined situation awareness as "*the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future.*" Awareness in computer-supported cooperative work has been defined as "*an understanding of the activities of others, which provides a context for your own activities*" [DB92]. Sohlenkamp [Soh98] defined awareness as "*an understanding of the state of a system, including past activities, present status and future options.*" In this thesis, awareness refers to an understanding of the consequences of sharing personal information in the past, present and future.

The more personal identifiable information a data subject discloses, the higher the risk is for privacy violation. To ensure awareness, a number of technical enforcement tools have been developed. For instance, the concept of personal information feedback tools has been promoted [LHDL04, PK09] to help users gain privacy awareness and determine themselves which personal data to disclose.

The Platform for Privacy Preferences Project (P3P) [W3C] has been designed to allow websites (as data controllers) to declare their intended use of the information that they collected about the browsing users (as data subjects). P3P addresses the awareness property by making users aware of how personal data are processed by the data controller (i.e. the individual or group who determines the purposes and means of the processing of personal data).

Although not necessarily privacy-oriented, another responsibility of the user, within the realm of content awareness, is to keep user's data up-to-date to prevent

wrong decisions based on incorrect data. This means that the data subject or the data controller (depending on the applications) is responsible for deleting and updating inaccurate information. For example, it is crucial to maintain the patient's data in an e-health application. Imagine a doctor forgetting to mention that the patient is a diabetic, the absence of information could cause fatal consequences for patients taking medication without considering negative side effects on diabetics. Note that the definition of awareness is altered in the improved version of LINDDUN (see Chapter 6). As awareness in the current definition puts quite some responsibility on the user, this will be shifted towards the system side.

Compliance The policy and consent compliance property requires the whole system as data controller to inform the data subject about the system's privacy policy, and allow the data subject to specify consents in compliance with legislation, before users accessing the system. According to the definitions from the EU Directive 95/46/EC [Eur95]: *"Controller shall mean the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data."* *"The data subject's consent shall mean any freely given specific and informed indication of his wishes by which the data subject signifies his agreement to personal data relating to him being processed."*

A policy specifies one or more rules with respect to data protection. These are general rules determined by the stakeholders of the system. A consent specifies one or more data protection rules as well, however, these rules are determined by the user and only relate to the data regarding this specific user. The policy and consent compliance property essentially ensures that the system's policy and the user's consent (usually specified in textual form) are indeed implemented and enforced.

This property is also related to legislation. There are a number of legal frameworks addressing the raised concerns of data protection, such as the Health issued the Insurance Portability and Accountability Act (HIPAA) [HIP06] in the United States, the Data Protection Directive 95/46/EC [Eur95] in Europe, and the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data [OEC80].

One example of consent compliance can be found in an e-health context. In some countries, healthcare professionals are only allowed to access medical information if the data subject has given informed consent (or in case of emergency).

There are initiatives to protect data subjects and create openness. It is clearly important to ensure that internal rules actually comply with those promised in

Table 2.1: Summary of privacy vocabulary

Privacy Concept	Description
Personal Identifiable Information (PII)	Information which can be linked back to an individual
Data Subject	Individual that is linked to the PII
Item of Interest (IOI)	information related to an individual (e.g. subjects, messages, actions, etc.)
Unlinkability	Not being able to distinguish whether 2 IOIs are related
Anonymity	Not being able to identify the subject within a set of subjects
Plausible deniability	Being able to repudiate having performed an action
Undetectability	Not being able to distinguish whether an IOI exists
Unobservability	Undetectability against all subjects involved
Confidentiality	Authorized restrictions on information access and disclosure
Awareness	Being conscious about consequences of sharing (PI) information
Compliance	Following regulations and internal business policies

policies and consents. Unfortunately, few technical solutions exist to guarantee compliance. A possible non-technical solution is to use employee contracts to enforce penalties (e.g. , get fired or pay fines) to ensure compliance. Another solution is to hire an auditor to check policies compliance. Eventually, necessary legal actions can be taken by data subjects in the case of non-compliance.

To conclude, we revisit these privacy properties together with the legal vocabulary described in the previous section by summarizing their definitions in Table 2.1 for easy reference.

2.1.4 Privacy Taxonomies

Similar to the myriad of available privacy definitions, there have been several attempts to structure and classify privacy concepts. First we discuss taxonomies that address privacy from a legal perspective. Second, we summarize the privacy classifications that are specifically targeted to software engineering.

Note that we have also created our own privacy taxonomy to classify privacy solutions. It is described in Chapter 6 where it is linked to LINDDUN.

Solove's Taxonomy Solove presents a taxonomy of privacy violations from a legal perspective [Sol06]. Even though the taxonomy does not discuss digital privacy, but describes privacy in general, it provides some useful insights in the matter. Solove makes a distinction between 4 basic groups of harmful activities: information collection, information processing, information dissemination, and invasion.

In the first group, information collection, he included two types of privacy violations: surveillance, which he defines as "the watching, listening to, or recording of an individual's activities", and, interrogation which consists of various forms of probing for information.

The second group of harmful activities, information processing, considers the use, storage, and manipulation of data that have already been collected. It consists of 5 types of violations: aggregation (i.e. combining data related to an individual), identification (i.e. linking data to individuals), insecurity (i.e. carelessness in protecting stored data), secondary use (i.e. using data for a different purpose than it was collected for), and exclusion (i.e. when the data subject is unaware about the data others have about her).

The third group concerns information dissemination and contains 7 violation categories: breach of confidentiality (i.e. not keeping a person's information confidential although promised), disclosure (i.e. revealing truthful 'sensitive' information about a person), exposure (i.e. revealing someone's nudity, grief, or bodily functions), increased accessibility (i.e. amplifying information accessibility), appropriation (i.e. use of one's identity to serve another one's purpose), and distortion (i.e. dissemination of false information).

The final group concerns invasion, and unlike the previous groups, does not necessarily involve personal information. It consists of invasion violations (i.e. invasive acts that violate a person's tranquility) and decisional interference violations (i.e. governmental incursion into a person's private decisions).

FIPPs. The Fair Information Practice Principles (FIPPs) [fip73] are a set of guidelines proposed by the United States Federal Trade Commission. They can be considered the foundation of all current data protection legislation. They, for example, served as basis for the guidelines of the Organisation for Economic Cooperation and Development (OECD) [OEC80], and the European Data Protection Directive [Eur95], among other influences. There are 5 core FIPP categories which can be described as follows:

- Notice/Awareness: Consumers should be properly informed before collecting personal information.
- Choice/Consent: Consumers must be able to choose how their personal information will be used. This holds in particular to secondary use (e.g. registration to a mailing list or transfer of information to third parties).
- Access/Participation: An individual should be able to access data about himself and to contest that data's accuracy and completeness.
- Integrity/Security: Data should be accurate and secure.
- Enforcement/Redress: There should be enforcement measures in place to ensure the FIPPs are being followed. Three alternative approaches have been suggested: self-regulation, private remedies, or government enforcement.

The FIPPs are also used as basis for other privacy taxonomies. Microsoft's Privacy Guidelines [mic08], for example, are based on the core concepts of the FIPPs.

Also, the privacy taxonomy defined by Anton et al. [AER02], which was created to classify and analyze privacy goals and requirements, is based on the FIPPs. However, their framework does not only contain the 5 FIPPs as privacy protection goals, but also includes a set of privacy vulnerability goals that are related to existing threats. These vulnerability goals include information monitoring, information aggregation, information storage, information transfer, information collection, information personalization, and contact.

European Data Protection Legislation. Legislation is a complex matter. It is often vague and formulated in an ambiguous way which makes it very hard to implement. Note that data protection and privacy legislation should not be used interchangeably. De Hert and Gutwirth [DHG06] define privacy as a tool of opacity to limit power by setting normative rules. Data protection, on the other hand, is defined as a tool of transparency to channel and allow power. Data protection legislation is however more in line with the privacy requirements for software systems that are covered in this thesis.

Many claim that trying to hard-code all legislative rules should be avoided, as a techno-regulation only moves the bureaucratic overhead to system developers [KL14]. Privacy does not only require technological measures, but needs organizational measures as well. In addition, it is hard to foresee all potential domains and contexts (and their corresponding regulations) in which a software product will be used. And even if all regulations are properly identified, they

often have a fuzzy, open-ended description. Nevertheless, some of the data protection legislation will impact system requirements, or can, with some minor effort, be incorporated into the system's design.

Guarda and Zannone [GZ09] summarize the *European Data Protection Directive* [Eur95] in the following 9 principles:

1. Fair and Lawful Processing: the collection and processing of personal data shall neither unreasonably intrude upon the data subjects' privacy nor unreasonably interfere with their autonomy and integrity, and shall be compliant with the overall legal framework.
2. Consent: personal data shall be collected and processed only if the data subject has given his explicit consent to their processing.
3. Purpose Specification: personal data shall be collected for specified, lawful and legitimate purposes and not processed in ways that are incompatible with the purposes for which data have been collected.
4. Minimality: the collection and processing of personal data shall be limited to the minimum necessary for achieving the specific purpose. This includes that personal data shall be retained only for the time necessary to achieve the specific purpose.
5. Minimal Disclosure: the disclosure of personal data to third parties shall be restricted and only occur upon certain conditions.
6. Information Quality: personal data shall be accurate, relevant, and complete with respect to the purposes for which they are collected and processed.
7. Data Subject Control: the data subject shall be able to check and influence the processing of his personal data.
8. Sensitivity: the processing of personal data, which are particularly sensitive for the data subject, shall be subject to more stringent protection measures than other personal data.
9. Information Security: personal data shall be processed in a way that guarantees a level of security appropriate to the risks presented by the processing and the nature of the data.

As the DPD was created in a time where the Internet was still in its infancy, a proposal has been drafted in 2012 that proposes a reform of the current legislation to strengthen online privacy rights [Eur12]. Key changes include the

'right to be forgotten' and the requirement that consents necessary for data processing are given explicitly. The 'right of data portability' allows easier access to one's own data and more transparency about how data are handled is required as well. Also the responsibility and accountability for those processing personal data is increased by applying principles like 'Privacy by Design.'

The *E-Privacy Directive* [Eur02] was enacted in 2002 and supplements the DPD as it focuses on data protection in the digital age. It regulates the electronic communications sector and was amended in 2009 [Eur09]. It is mainly known for requiring the user's consent before storing cookies, and is therefore often referred to as the 'Cookie Directive.' In addition, the directive regulates online spam by imposing an opt-in regime, where unsolicited emails can only be sent with prior agreement of the recipient. Also the storage and processing of both traffic data and location data are included in the Directive. Traffic data should be erased or made anonymous as soon as they are no longer needed for the purpose of the transmission. Processing of these data can only occur when the data are anonymized or when the subscriber has given his consent.

Although data protection legislation is complex and often ambiguous, some rules can be automated in software systems. In recent years, research has emerged that aims at abstracting rights and obligations from legal documents and that provides traceability between (written) privacy policies and their implemented software counterparts [AER02, BA08, SBSCB06, CHCGE10, YA10, AGR14].

Privacy Paradigms. Gürses [Gür10] has classified the privacy technology landscape according to three privacy paradigms: privacy as control, privacy as confidentiality, and privacy as practice.

Privacy as control aims at providing data subjects with a means to control disclosure to their own data. Also organizational means to define and enforce data security policies and prevent abuse of unauthorized access fall into this category. Examples of related technologies include privacy settings, access control and auditing to facilitate appropriate data usage.

Privacy as confidentiality puts less trust into the organizations. This paradigm prevents disclosure of information or at least minimizes the disclosure as much as possible to avoid linkability to the data subject. Example technologies are anonymous authentication protocols and anonymous communication networks.

Privacy as practice focuses more on the social aspect of privacy and aims at make information flows more transparent through feedback and awareness tools. Note that these three paradigms are not mutually exclusive.

Privacy by Design. Privacy by Design [vRGB+95, Cav09, GGTD11] is a concept that was developed in the late 90's by Ontario's Information and Privacy Commissioner Ann Cavoukian. It is an engineering approach that focuses on the entire process starting from privacy and data protection principles.

Privacy should not merely be assured by compliance and regulatory frameworks, as there is no use in hard-coding legal rules if the system itself does not have a solid security and privacy foundation.

Privacy by Design can be accomplished by applying the 7 Foundation Principles:

1. *Proactive not reactive; Preventative not remedial*: Privacy threats should be anticipated and prevented, rather than remedied after they have occurred.
2. *Privacy as default setting*: Privacy should be the standard. Personal data should be automatically protected, even without any actions from the individual himself.
3. *Privacy embedded into design*: Privacy should not be considered as an add-on, but should be embedded in the design and architecture of software systems and business activities in general.
4. *Full functionality - positive sum, not zero-sum*: Privacy should coexist with other business interests. Unnecessary trade-offs should however be avoided (e.g. privacy vs. security or privacy vs. performance). One should strive for a positive sum.
5. *End-to-end security - full lifecycle protection*: Privacy requires security throughout the entire lifecycle of the personal data, ensure secure cradle to grave management of all data.
6. *Visibility and transparency - keep it open*: Privacy objectives and promises stated by the business should be followed and systems should be operating accordingly. These system objectives should also be brought to the users' attention.
7. *Respect for user privacy - keep it user-centric*: Privacy interests of the user are most important, thus measures to empower the user should be applied.

Privacy by Design has been heavily promoted in the privacy community. Indeed, providers have to enable users to better protect their personal data. The approach has already obtained positive results. For example, the roll-out of a smart grid system in Ontario was successful as it included Privacy by Design

principles early on in the development process [Kre13]. The approach does however not always result in a positive outcome. The ELENA project in Germany, which was designed to collect income information for employees, was discontinued even though data protection authorities were already involved in the early stages of the project [Kre13, Sch10]. Despite the privacy requirements that were included, the project was heavily criticized by privacy activists. One of the critiques concerned the inclusion of all data of the conventional paper forms, although the need for some of these data fields was doubtful. This experience illustrates that Privacy by Design cannot be successful when it is reduced to merely an endorsement of security and data protection functionality, without acknowledging evolvable requirements.

2.2 Privacy Requirements Methodologies

Requirements engineering is concerned with the elicitation, evaluation, specification, analysis and evolution of the objectives, functionalities, qualities and constraints to be achieved by a software-intensive system within some organizational or physical environment [VL09]. This definition can become particularized to *privacy requirements engineering* which is the elicitation, evaluation, specification, analysis and evolution of privacy objectives and constraints to be achieved by a software system. Indeed, there is a need for a systematic approach to incorporate privacy into the software system [AGR13]. This section summarizes the related work in this area.

PRoPAN. Beckers et al. [BFHM14] created a four-phase approach to semi-automatically identify privacy threats. First, a context diagram and problem frame diagrams for the set of functional requirements are drawn. Second, the privacy requirements that the system should enforce are added. Based on this input, threat graphs are automatically generated in the third step of the approach. These privacy threat graphs are then analyzed during the final phase. The uncovered threats can be resolved by modifying or adding a requirement to either prohibit the counter-stakeholder to gain personal information or to prohibit that personal information is processed or stored in a certain domain. While this approach provides some automated support for certain steps, it does not provide any privacy knowledge to the analyst. Also, the privacy requirements that are added in step 2 are very high-level (e.g. preserve anonymity) and the approach does currently not support the analysis of external attacker threats.

PriS. PriS [KKG08] is a privacy requirements method which covers the core privacy properties. The method consists of four phases: elicit privacy-related goals, analyze the impact of privacy goals on organizational processes, model affected processes using privacy-process patterns, and identify the techniques that best support or implement the above mentioned processes. The method however does not specify the elicitation process of privacy requirements, which is one of the key features of a privacy requirements analysis methodology; it merely provides a set of concepts and a systematic translation of privacy requirements into system models. Process patterns are provided for each privacy property, which specify the specific privacy-related activities that should be implemented. For each pattern, also a mapping with existing privacy techniques has been made. PriS can thus be classified as a method that focuses on the translation of privacy requirements to privacy solutions, which can only be executed in a later stage of the requirements analysis process.

FPFSD. Spiekermann and Cranor [SC09] developed a framework for privacy-friendly system design (FPFSD) where they make a distinction between two different privacy approaches. Privacy-by-policy introduces the notice and consent approach based on the Fair Information Practice Principles (FIPPs [fip73]) and implies that the user is informed about what information is used and why. In addition, the user can decide not to provide data. The second approach, privacy-by-architecture, encourages the storage of data at the client-side instead of having the companies themselves store privacy sensitive information. The FPFSD categorizes systems according to 4 privacy stages. Stages 0 and 1 have a privacy-by-policy approach and their data are still easily identifiable. Stages 2 and 3 follow the privacy-by-architecture approach, and work with (pseudo-)anonymous data. Systems are categorized into their corresponding stage of this framework by their system characteristics, e.g. unique identifiers across databases will lead to privacy stage 0, while k-anonymity [Swe02b] (with a large k-value) corresponds to privacy stage 3.

MPRA. Gürses [Gür10, GÏ13] proposes a multilateral privacy requirements analysis technique (MPRA). It consists of three main phases: stakeholder analysis, functional analysis, and privacy analysis. First the actors and stakeholders are determined. For each of them, the functional goals are determined and linked to domain assumptions. Also, an information model is created. In the final phase, the privacy concerns are determined based for each stakeholder, an relate to the information model and functional goals that were identified in the previous step. These concerns can also be translated into privacy threats that are documented as misuse cases. Finally, the concerns (or threats) are transformed to privacy goals as a justifiable approximation.

Based on these goals, suitable privacy solutions can be chosen and implemented. As this impacts the functional goals, the results of each step will be revised iteratively.

Privacy in the Cloud. Mouratidis et al. [MIKG13] focus on cloud applications in particular and propose a framework that supports elicitation of security and privacy requirements. A cloud service provider can then be selected based on its satisfiability to the security and privacy requirements. The framework consists of a language (based on Secure Tropos) and a process (based on PriS) to support the security and privacy analysis. The process consists of three steps. The first (optional) step is security and privacy threat cataloging which aims to create a reference point based on previous experience for later use. Second, the security and privacy analysis activity consists of two sub-activities: defining the organizational context, where organizational goals, actors and dependencies, plans and resources, and security and privacy goals are identified; and defining security and privacy concerns, which consists of identifying the security and privacy requirements, measures and mechanisms. The third and final step is the selection of the cloud service provider based on the degree of satisfaction of the obtained security and privacy mechanisms by potential cloud providers. Although this is a framework that provides detailed steps to analyze security and privacy requirements, no actual security and privacy knowledge is available.

Adaptive privacy. Omoronyia et al. [OCS+13] propose a framework to support selective disclosure of personal information in software applications with a frequently changing context. The framework focuses on privacy awareness requirements (PAR) and describes 1) how to identify the attributes to be monitored in order to detect privacy threats, 2) how to discover privacy threats before personal information is disclosed, and 3) how to determine the severity, as well as benefits related to a discovered threat. The framework however requires that the privacy requirements of all agents are already defined, and does not provide guidance on how to obtain them. An initial privacy analysis is hence still required.

STRAP. STRAP [Jen05] is a privacy analysis framework that was developed based on the analysis results of 6 existing frameworks (Risk models [HNL04], Patrick and Kenny [PK03], i* framework [YC02], Langheinrich [Lan01], Bellotti and Sellen [BS93], and Heuristic evaluation [NM90]). It is a five-step method, which consists of goal-oriented analysis, vulnerability analysis, goal refinement and design, design evaluation, and iteration. The vulnerability analysis step is the main phase where the existing frameworks are combined. The analysis

relies on a set of analytical questions to determine the capture and use of information. Second, heuristics are used to identify potential problems based on common flaws and requirements. These heuristics can be categorized according to the US FIPPs [fip73]: notice/awareness, choice/consent, integrity/security, enforcement/redress. Although these categories, based on data protection legislation and guidelines, are very important privacy concerns, the core properties such as anonymity, unlinkability, and undetectability are neglected.

Microsoft privacy guidelines. The privacy guidelines provided by Microsoft describe some basic privacy concepts [mic08], such as different types of consents or data minimization concepts. In addition, a number of guidelines are presented for selected scenarios concerning the following principles: notice, choice, onward transfer, access, security, and data integrity. However, it only contains a flat list of the required and recommended guidelines and does not intend to describe a more structured approach. These guidelines can still be used as inspiration to determine possible threats, e.g. to extend our catalog of threat trees.

In the documentation of Security Development Lifecycle (SDL) version 3.2 [sdl08], privacy is also partially considered, but only at a generic level. For example, the threat modeling process described in the fourth stage of SDL only mentions that a design review with the privacy expert is necessary. SDL also presents ten general privacy guidelines. An example guideline indicates that it is important to collect the least sensitive form of data. At this stage, privacy is not yet well integrated in SDL.

i∗. Yu and Cysneiros [YC02] presented a framework using $i*$, an agent-oriented requirements modeling language, to deal with privacy requirements. This framework however focuses on reasoning about privacy and does not provide a structured methodology to examine the different privacy objectives. Liu et al. [LYM03] proposed a framework also using $i*$ to deal with security and privacy requirements, which was inspired by the work of Yu and Cysneiros. They use four different analysis techniques to create a complete model: attacker analysis, dependency vulnerability analysis, countermeasure analysis and access control analysis. This framework is again mainly meant to reason about privacy but lacks the necessary knowledge to empower the (non-expert) privacy analyst.

PRET. Miyazaki et al. [MMZ08] defined a computer-aided privacy requirements elicitation technique (PRET). This technique returns the appropriate requirements for the system to be compliant with the law, based on a questionnaire that the system engineer fills out.

None of the methods described in this section fitted our needs. We therefore examined existing threat modeling approaches to investigate how this concept can be applied to privacy. These approaches are summarized in the following section.

2.3 Threat Modeling

This section provides an overview of existing threat modeling techniques. As privacy threat modeling research is still less established, we mainly focus on security threat modeling techniques.

Microsoft's STRIDE receives the most attention in this section, as it has been used as foundation for our LINDDUN methodology.

2.3.1 STRIDE

Security, in contrast to privacy, has already been well integrated in software development. An example hereof is Microsoft's Secure Development Lifecycle (SDL) [HL06], which is a well-established methodology. To build a secure software system, an important aspect is to consider how an attacker might compromise the system by exploiting design flaws in order to build the necessary defense mechanisms in the system. In this respect, threat modeling plays a key role. SDL has integrated a systematic approach for security threat modeling using STRIDE. Below, we briefly review the STRIDE threat modeling process, which consists of nine high-level steps.

Step 1: Define use scenarios. System designers need to determine which key functionality is within the scope.

Step 2: Gather a list of external dependencies. Each application depends on the operating system it runs on, the database it uses, and so on. These dependencies need to be defined.

Step 3: Define security assumptions. In the analysis phase, decisions are often based on implicit assumptions. Therefore, it is important to note down all the assumptions, in order to understand the entire system comprehensively.

Step 4: Create external security notes. Because each external dependency can have its implication on security, it is useful to list all the restrictions and implications introduced by the external security notes. An example of such a security note is to specify which ports are open for database access or HTTP traffic.

Step 5: Create one or more data flow diagrams (DFDs) of the application being analyzed. The software system being analyzed is decomposed into relevant (either logical or structural) components, and for each of these parts the corresponding threats are analyzed. This process is repeated over an increasingly refined model until a level is reached where the residual threats are acceptable.

The system is graphically represented using a data flow diagram (DFD), with the following elements: data flows (i.e. communication data), data stores (such as logical data or concrete databases and files), processes (i.e. units of functionality or programs) and external entities (i.e. end-points of the system such as users, external services).

An example DFD is shown in Figure 2.1 to illustrate a use case application (Social Network 2.0) that is discussed throughout this thesis. This Social Network 2.0 application is an abstract representation of a social network, where online users share personal information such as relationship status, pictures, and comments with their friends. In the DFD, the user is represented as an entity to interact with the system. The Social Network 2.0 application contains two processes (the portal and the service) and one data store containing all the personal information of the users. Note that the model in Figure 2.1 has been kept high-level for the sake of simplicity in this example. In practice, the DFD will be more detailed. Evidently, the more fine-grained the DFD, the more detailed the elicited threats will be.

Step 6: Determine threat types. The STRIDE threat taxonomy is used to identify security threat types. STRIDE is an acronym for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. These threats are the negation of the main security properties, namely confidentiality, integrity, availability, authentication, authorization and non-repudiation.

Step 7: Identify the threats to the system. Each element of the data flow diagram is assigned to a set of susceptible threats. Table 2.2 gives an overview

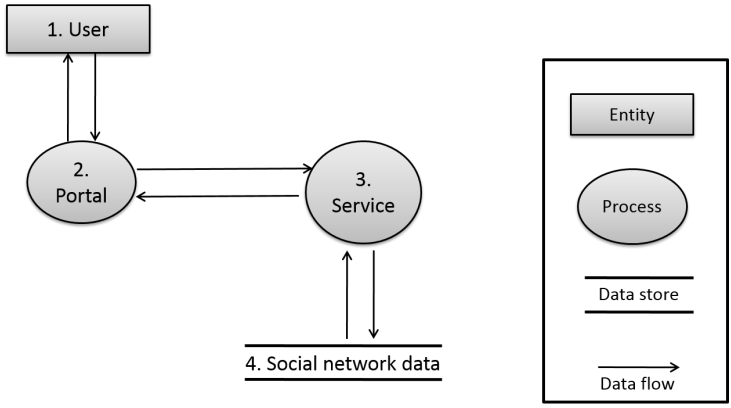


Figure 2.1: The Data Flow Diagram (DFD) of the Social Network 2.0 application

Table 2.2: Security concerns with corresponding security threats and DFD elements susceptible to threats (DF-Data flow, DS-Data store, P-Process, E-External entity), proposed by the Security Development Lifecycle (SDL) [HL06].

Security property	Security threat	DF	DS	P	E
Authentication	Spoofing			×	×
Integrity	Tampering	×	×	×	
Non-repudiation	Repudiation		×	×	×
Confidentiality	Information Disclosure	×	×	×	
Availability	Denial of Service	×	×	×	
Authorization	Elevation of Privilege			×	

of the different DFD elements with the corresponding security threats they can be subject to (marked with ×).

To identify which threats are applicable to a specific system, threat tree patterns can be used. For each valid intersection in Table 2.2, a threat tree pattern suggests the possible security-related preconditions for the STRIDE category, in order to help analysts determine the relevance of a threat for the system. An example threat tree is presented in Figure 2.2. Each path of the threat tree indicates a valid attack path. Note that some trees cascade. For example, the tree in Figure 2.2 shows the main conditions that could lead to tampering threats against a process. Either the process has a corrupted state by lack of input validation or because of privileges that wrongly lead to access to the

process' memory, or a subprocess is susceptible to tampering, or false credentials are provided because of a spoofing threat or because of a failure to check the call chain and code that calls your code (callers) or code that your code calls (callees) was wrongly trusted. The node indicated as a circle (or oval) in the threat tree indicates a root threat. These are the main STRIDE threats which, indirectly, can lead to another root threat, e.g. someone can indirectly tamper with a process by spoofing an external entity. The node indicated as a rectangle suggests a concrete threat in an attack path. The arrows connecting the nodes in general refer to a *OR* relation among the various preconditions, unless it is indicated explicitly with "AND" to refer to an *AND* relation. This means that (unless an explicit 'AND' relationship is indicated) the branches of the tree are disjunctive and each of the child nodes can lead to the root threat. This implies that all of the branches and corresponding leaf nodes have to be analyzed and mitigated in order to counter the root threat.

The identified threats are listed in a large table consisting of the STRIDE threat type and the specific DFD elements to which the threat applies.

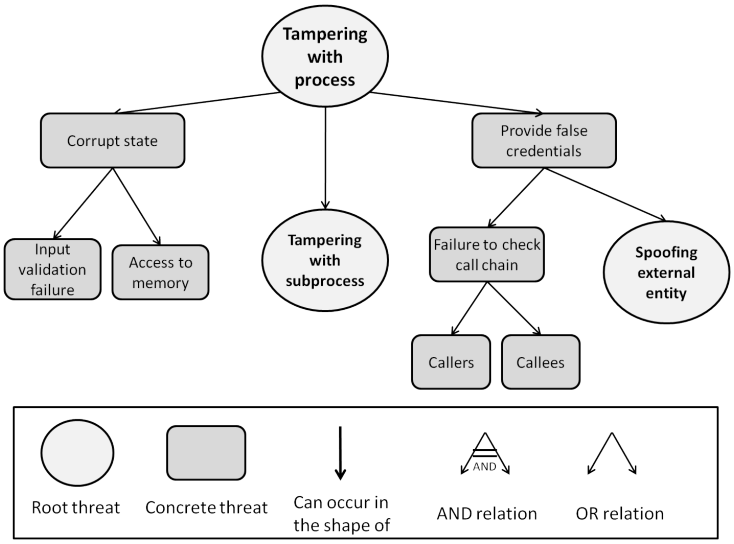


Figure 2.2: Example security threat tree pattern of tampering a process [HL06]

Step 8: Determine risk. For each threat, the appropriate security risk level has to be determined, which can be used to define the priorities of the threats to be resolved.

Step 9: Plan mitigation. In the final step of the methodology, the risk of the threat is reduced or eliminated by introducing proper countermeasures and defenses. Mitigating a risk to the threat corresponds to eliminating one attack path in the threat tree. An overview of some possible mitigation technologies linked to each security property is provided.

2.3.2 Alternative Threat Modeling Techniques

STRIDE comes with the main advantage of an extensive, reusable knowledge base (i.e. the threat tree patterns). However, some alternatives to elicit security threats exist.

Attack Trees. The root node of an attack tree [Sch00] describes the high-level attack which is further decomposed into lower-level attack branches. Each node represents a step that must be successfully executed in order to complete the attack represented by the parent node. Nodes can be composed in conjunctions and disjunctions. Attack trees can have both a graphical and a textual representation.

Misuse Cases. Misuse cases or abuse cases [Ale03, SO01, SO05, MF99] are similar to regular use cases, however with a focus on the attacker's actions. Misuse cases have a textual representation, similar to use cases, and can also be represented in a misuse case diagram, which summarizes all existing misuse cases for a certain system and their impact on the system's use cases. The technique can be used to elicit security or privacy threats; the technique however does not provide methodological guidance to discover additional threats. Opdahl and Sindre [OS09] have made an experimental comparison between attack trees and misuse cases of which the main finding was that attack trees are more effective for finding threats. Attack trees encourage the use of standard textbook threats and decomposition in lower-level threats. Misuse case analysis focuses more on user-level and organizational threats.

KAOS. KAOS [VL09], a goal-oriented requirements analysis framework, has been extended with anti-goals to support the modeling of threats. Such anti-goals express the goals of an attacker who tries to abuse the system. Although no actual methodology exists to determine the threats, the root anti-goals are created by negating all the positive system goals. Next, the anti-goals are refined into trees. The formal nature of KAOS is an advantage which makes it possible to determine completeness of the (anti-)goals.

Safety Threat Modeling. Also in the field of safety, threat modeling techniques are being applied to systems requiring a high level of reliability. Note that these systems are not software systems, but systems that require high reliability, such as nuclear power plants.

Failure Mode and Effects Analysis (FMEA) [fmeon] dates back to the 1950's, where it was created to analyze malfunctions of military systems. The technique systematically reviews all components to identify failure modes, failure causes and its effects.

A hazard and operability study (HAZOP) [haz02] is a systematic examination of processes to evaluate possible safety issues. A multidisciplinary team carries out the analysis during a set of meetings by applying a set of HAZOP guide words to each section of the process to identify deviations. It was initially developed to analyze chemical process systems, but has later been extended to other types of systems.

2.4 Conclusion

Privacy is important, especially in today's society where personal data are ubiquitous. Unfortunately, privacy is often neglected during software development. Privacy requirements methodologies are emerging, but they do not fully meet the expectations. Either they fail to provide substantial methodological guidance throughout the analysis or they lack the required privacy support. To fill this gap, we propose the LINDDUN methodology in Chapter 3.

The background study described in this chapter can be considered a useful building block in the creation of LINDDUN. For example, the related work on privacy requirements, and on threat modeling techniques in particular, provided useful insights. LINDDUN is in fact inspired by STRIDE, a well-established threat modeling approach to identify security threats.
Also, the existing privacy classifications were instrumental for the LINDDUN methodology. Although LINDDUN is not a compliance technique, it does implement several principles imposed by data protection legislation (e.g. consent, minimality, awareness, etc.) and explicitly draws attention to the need of regulatory compliance. Furthermore, the privacy properties, summarized in Section 2.1.3, form the basis of the threat categories that are handled by LINDDUN.
Finally, LINDDUN also adheres to the Privacy by Design principles as it aims at introducing privacy as early as possible in the software development lifecycle.

3

The LINDDUN Threat Modeling Methodology

In this chapter[1], we present LINDDUN, a systematic approach for privacy threat modeling. The methodology was inspired by the STRIDE threat modeling methodology for security. LINDDUN aids at identifying privacy threats in a software system and provides support to elicit its corresponding privacy requirements and even to select appropriate mitigation strategies.

"LINDDUN" is an acronym which reflects the threat categories it aims to uncover: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, and Non-compliance. These threat types are briefly described in Section 3.3. Their corresponding 'positive' privacy properties have already been described in Section 2.1.3.

The building blocks of the LINDDUN methodology are depicted in Figure 3.1. A distinction can be made between LINDDUN's core steps that are situated in the problem space and aid at eliciting privacy threats, and the LINDDUN

[1]This chapter is based on the journal paper "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements" [DWS+11]

steps that are positioned in the solution space and translate threats into privacy solutions.

First of all, a data flow diagram is created based on the high-level system description. This is followed by mapping privacy threats to the DFD elements using Table 3.2 as a guide to determine the corresponding threats. In particular, a number of privacy threat tree patterns are presented in Section 3.4 to detail the privacy threat instances in a structured fashion, by providing an overview of the most common preconditions of each threat. The identified privacy threats that are relevant to the designated system are documented as misuse cases. Furthermore, all the potential privacy threats that are suggested by the privacy threat trees are evaluated and prioritized via risk assessment. Note that LINDDUN does not provide explicit risk analysis support. We merely refer the analyst to established risk assessment techniques. Hereafter, the privacy requirements of the system are elicited from the misuse cases based on the threat tree branches they correspond to. Finally, appropriate privacy enhancing solutions are selected according to the privacy requirements. Table 3.3 provides an overview of the state-of-art privacy enhancing techniques and the mapping to their corresponding privacy objectives.

Note that we provide an updated version of LINDDUN in Chapter 6 as the results of the studies described in Chapters 4 and 5 proposed some improvements to the methodology.

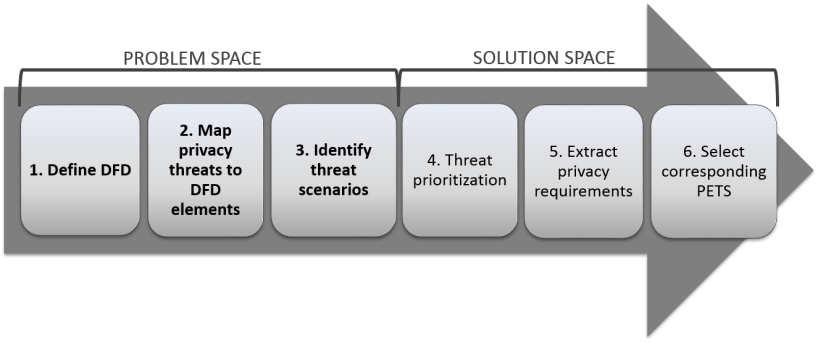


Figure 3.1: The LINDDUN methodology steps

The fact that the LINDDUN framework and STRIDE are based on similar approaches creates synergy. Therefore, the privacy and security analysis can be nicely integrated into SDL. Nevertheless, LINDDUN can also be performed independently.

3.1 LINDDUN in the Software Development Lifecycle

Earlier on, we mentioned that threat modeling plays an important role in the early stages of the software development lifecycle. Requirements engineering is the first development stage, which results in the documentation of the combined expectations of the system's stakeholders, e.g., in the form of functional and quality requirements. Architectural design translates these requirements to a structural blueprint of the system by means of a series of design choices about how functionality is decomposed and responsibilities are distributed. The result is a technical documentation in the form of, among others, architectural diagrams. These two steps should however not be seen as strictly isolated, subsequent steps [Nus01]. There is a synergy between both steps, as a decision at the architecture level can impose changes to the requirements.

Threat modeling can (and should) be applied in both stages. Indeed, it is advised to perform continuous threat modeling, which implies that all software specifications are analyzed for threats. Threat modeling focuses on the potential weaknesses that might allow an attacker to cause the damage or loss of an asset, such as sensitive data. Threat modeling can be applied at different levels of abstraction, depending on the granularity of the considered assets. At the requirements level, assets are more abstract, such as information objects and functionality in the use case scenarios. At the level of architectural design, assets are more tangible, such as computational components and communication channels.

LINDDUN is a *model-based* technique. That is, threat analysis in LINDDUN relies on a model of the assets in the system-under-analysis. The model is documented as a data flow diagram (DFD) and spans the entire analysis process. As the DFD can be constructed starting from either the requirements specification or the architectural diagram, LINDDUN can be used in both development phases. Obviously, the different level of abstraction influences the granularity of the analysis results.

The following section provides an overview of the six LINDDUN methodology steps. The chapter concludes with some thoughts related to the coverage of the LINDDUN methodology.

Figure 3.2 provides a graphical overview of the running example, a privacy-aware social networking system. This example has been kept simple and high-level on purpose. A more extensive example of LINDDUN can be found in [WSJ].

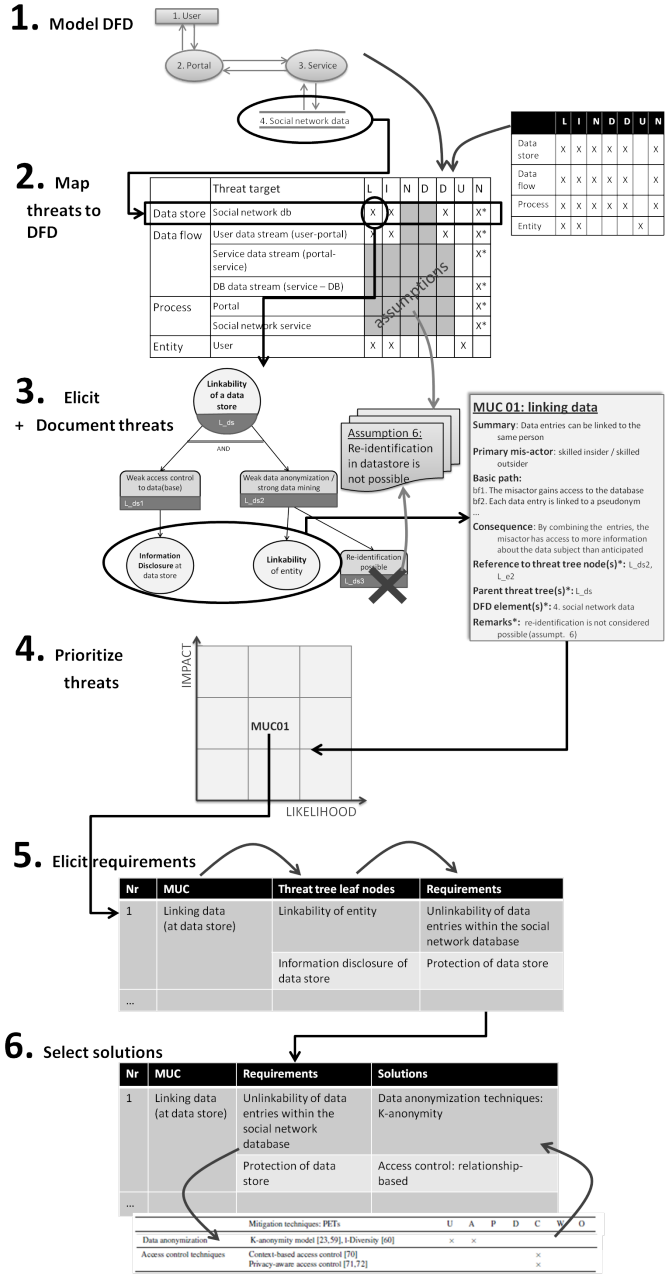


Figure 3.2: A step-by-step overview of the LINDDUN methodology using a simple social network system as running example

3.2 Step 1: model DFD

First, a data flow diagram (DFD) is created. DFD was chosen as suitable representation of a software system based on two reasons. First, a DFD is proven to be sufficiently expressive in a number of case studies examined by the authors. Second, DFD is also used by the SDL threat modeling process, hence by deploying the same modeling technique an interesting synergy can be created between the proposed framework and the STRIDE process.

This DFD model is based on the available system description, which can be at the level of both requirements and architecture. A DFD is a structured, graphical representation of the system using 4 major types of building blocks: external entities, data stores, data flows, and processes. The DFD describes the way information flows throughout the system via data flows (DF) that connect external entities (E), such as system users or 3rd-party services, with processing nodes (P), which are active elements, and data stores (DS), which are passive containers of information.

The creation of the DFD is an important part in the analysis. If the DFD would be incorrect, the analysis results would be wrong as well. Since privacy focuses on the protection of a user's personal information, it is important to consider where the information will be stored or passed by, as these are the crucial elements for building in privacy. The granularity of the DFD will therefore also impact the level of detail of the elicited threats.

Running example: Social Network 2.0. In our running example, Alice is a registered user of a social network. Each time Alice updates her friends list, she first connects to the social network's web portal. Accordingly, the portal communicates with the social network's server, and eventually, the friendship information of Alice and all other users of that social network is stored in a database.

The DFD for the Social Network 2.0 application is shown in step 1 of Figure 3.2. Note that, for the sake of simplicity, running example represents a quite abstract a social networking system. In practice, the DFD will be more detailed. A more extensive execution of LINDDUN can be found online [WSJ], where a step-by-step application of LINDDUN to a patient community system is provided.

Table 3.1: LINDDUN Privacy Threat Categories

Privacy properties	Privacy Threats
Unlinkability	**L**inkability
Anonymity and pseudonymity	**I**dentifiability
Plausible deniability	**N**on-repudiation
Undetectability and unobservability	**D**etectability
Confidentiality	**D**isclosure of information
Content awareness	Content **U**nawareness
Policy and consent compliance	Policy and Consent **N**on-compliance

3.3 Step 2: Map DFD Elements to LINDDUN Threat Types

As shown in Table 3.1, the methodology considers seven types of threats. LINDDUN is the mnemonic acronym that we use.

Our threat categorization, derived from the desired privacy properties as discussed in Section 2.1.3, considers privacy threats from the attacker's perspective. The (rather straight-forward) relationship between the LINDDUN privacy threat categories and the privacy properties is depicted in Table 3.1.

Each DFD element type is potentially subject to specific privacy threats. LINDDUN has identified the following 7 high-level categories:

- *Linkability (L)* occurs when one can sufficiently distinguish whether 2 items of interest (IOI, such as requests from a user) are related
- *Identifiability (I)* occurs when it is possible to pinpoint the identity of a subject (e.g., a user)
- *Non-repudiation (Nr)* occurs when it is possible to gather evidence so that a party cannot deny having performed an action
- *Detectability (D)* occurs when one can sufficiently distinguish whether an IOI exists, e.g., in a system
- *Disclosure of information (Di)* is the exposure of information to individuals who are not supposed to have access to it
- *Unawareness (U)* occurs when the user is unaware of the information he is supplying to the system and the consequences of his/her act of sharing

Table 3.2: Mapping LINDDUN components (privacy threats) to DFD element types (E-Entity, DF-Data flow, DS-Data store, P-Process)

Threat categories	E	DF	DS	P
Linkability	×	×	×	×
Identifiability	×	×	×	×
Non-repudiation		×	×	×
Detectability		×	×	×
Information **D**isclosure		×	×	×
content **U**nawareness	×			
policy/consent **N**oncompliance		×	×	×

- *Non-compliance (Nc)* occurs when the system is not compliant with the (data protection) legislation, its advertised policies and the existing user consents

As shown in Table 3.2, LINDDUN provides a table that indicates which threat categories are applicable to each DFD element type.

In the table, a '×' means that the DFD element type in the corresponding row is susceptible to the privacy threat category of the selected column. For example, only an external entity (e.g. a user) can be aware of the consequences of introducing personal information into the system. Therefore, the concrete threats for that category should be examined in relation to the elements of that type in a DFD built during the previous step.

In essence, each DFD element is subject to certain privacy threats, and the nature of the potential privacy threat is determined by the DFD element type. For example, a data flow is subject to a number of privacy threats such as identifiability, linkability, detectability, non-repudiation, and information disclosure. The following paragraphs explain how privacy threats affect DFD elements.

The nature of *linkability* indicates that the threat category affects DFD elements by pair. In other words, linkability of a DFD element refers to a pair (x_1, x_2), where $x \in \{E, DF, DS, P\}$ is the linkable IOI. Obviously, linkability of an entity, from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker can sufficiently distinguish whether these entities are related or not. Similar reasoning applies for data flows, data stores, and processes.

The *identifiability* threat category affects all four DFD elements. Essentially, identifiability at each DFD element refers to a pair (x, y), where $x \in \{E\}$ is the identifiable subject, and $y \in \{E, DS, DF, P\}$ is the attribute identifiability relates to.

Non-repudiation, the opposite of plausible deniability, is a privacy threat that affects data flow, data store and process elements. Even though entity is the only DFD element being able to (non-)repudiate, the non-repudiation privacy threat category actually applies to data flow, data store, and process elements. Similar to linkability and identifiability, non-repudiation at each DFD element refers to a pair (x, y), where $x \in \{E\}$ is the non-repudiating subject, and $y \in \{DS, DF, P\}$ is the attribute it relates to.

Detectability threats occur at data flow, data store, and process elements when the attacker can sufficiently distinguish whether it exists or not.

Information disclosure threats affect data flow, data store, and process, referring to the exposure of information at these DFD elements to individuals who are not supposed to have access to it. As this is also a security threat category included in Microsoft's STRIDE, we reused their mapping (and in the next step also their threat trees).

The *content unawareness* threat is only related to an external entity, since only an entity (data subject or data controller) can be aware of the consequences of (over)sharing personal information and should be able to access his information.

Policy and consent non-compliance is a threat that affects the system as a whole, because each system component (including data flow, data store and process) is responsible to ensure that actions are taken in compliance with privacy policies, legislative rules, and data subjects' consents.

Running example: Social Network 2.0. Considering the Social Network 2.0 application, the list of generic privacy threats to the modeled system is depicted in Step 2 in Figure 3.2. This is obtained by gathering the elements from the DFD (Step 1) and then determining the susceptible threats with Table 3.2 (also shown in the right of Step 2 in Figure 3.2).

The intersections marked in grey in the table of step 2 in Figure 3.2 are potential threats that have been considered as irrelevant to the specific usage scenario in this example. Each intersection that is indicated with a $\times$ shows that there will be a privacy threat at the corresponding DFD element. These items are the threats which we will actually consider.

For ease of example, we assume that DFD elements within the system boundaries

are trustworthy. We trust the processes as well as all data flows between trusted processes and between processes and data store. Therefore, we will not discuss linkability, identifiability, and information disclosure threats on these elements. We however do not trust the user and its communication with the portal and we also want to protect the data store containing all the user's information. Note that these kind of assumptions should be made explicit during the analysis (as explained below). If it is later on decided that certain assumptions do not hold after all, it should be easy to trace the impact of that decision throughout the analysis process.

In the following step, each of the × elements will be examined more closely to elicit the detailed threats. In the example of Figure 3.2, linkability threats for the data store are being scrutinized.

3.4 Step 3: Elicit Privacy Threats

The third step can be considered as the core execution step of the LINDDUN threat modeling methodology. As it is a quite labor-intensive task, we have divided it into three sub-activities: refining threats via threat tree patterns, documenting assumptions, and documenting threats using a threat template. Please keep in mind that most of these trees receive an update in Chapter 6 based on the results of the descriptive studies described in the following chapters. The latest version of the trees can also be found on the LINDDUN website [LIN].

3.4.1 Refine Threat via Threat Tree Patterns

For each "×" in the mapping table (see step 2), LINDDUN provides a list of concrete threats that need to be considered.

To ease the navigation, the threats are organized into trees.

The privacy threat trees are inspired by Secure Development Lifecycle (SDL) and based on the state-of-art privacy developments. These threat trees reflect common attack patterns and help application designers think about privacy conditions in the system. This section itemizes the privacy threat tree patterns and discusses how the analyst can determine which threats are relevant to the system.

It is assumed that the ideal pattern library is sufficiently large to ensure that all possible violations will be covered. The threat trees depicted in this section

present our original effort[2]. Comparable to the SDL threat tree patterns, the privacy threat trees are susceptible to a continuous improvement process, based on both the existing and the newly discovered threats.

In Chapter 6, we propose a set of improvements to the trees and the threat tree catalog in general, based on the results of the empirical studies.

For each marked intersection in Table 3.2, a threat tree exists showing the detailed preconditions for this specific threat category to materialize. The preconditions are hence vulnerabilities that can be exploited for a privacy attack scenario.

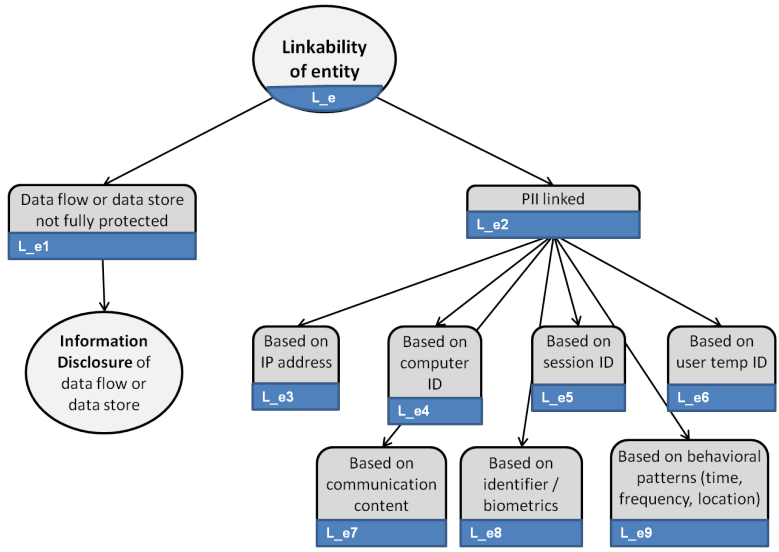


Figure 3.3: Threat tree for linkability of an entity

Linkability of entity. *Linkability of entity* refers to an attacker who can sufficiently distinguish whether two or more entities are related or not within the system. This implies that different pseudonyms can be linked to each other. The threat tree pattern is depicted in Figure 3.3. One precondition is that data flow or data store is not fully protected (e.g. unencrypted), which leads to the Information Disclosure threat of data flow and data store. The second precondition is that Personal Identifiable Information (PII) can be linked, e.g.

[2]The original trees have been actually extended with labels for easy reference in this chapter.

based on the user's temporary ID, IP address, behavioral patterns such as time, frequency and location, session ID, identifier and biometrics, computer ID, communication content or any combination of these factors. The aforementioned data store refers to the identity management database or any other database which contains personal identifiers of users. Having accessed such a data store, the attacker could easily link different pseudonyms to the same user.

Linkability of data flow. *Linkability of data flow*, as presented in Figure 3.4, suggests two possible preconditions. One precondition is that the data flow is not fully protected (e.g. unencrypted), because of information disclosure threats of the data flow. The other precondition describes linkable communication, either due to insecure anonymity systems being deployed or using non-anonymous communication which can be easily linked. When no anonymous communication is deployed, basically the same preconditions apply as for the linkability of entity threat. Messages are linked to each other by user's identifiable information (e.g. based on user temporary ID, IP address, behavioral patterns such as time, frequency and location, session ID, identifier and biometrics, computer ID, communication content or any combination of these factors). Alternatively, when an insecure anonymity system is deployed, traffic analysis is possible to extract information out of patterns of traffic; passive attacks (e.g. intersection attacks) and active attacks (e.g. N-1 attacks) are possible to link entities together. An overview of these attacks can be found in [DDS09].

Linkability of data store. Two preconditions correspond to the threats of *linkability of a data store*, as shown in Figure 3.5. First, there is insufficient access control of the data store leading to the information disclosure threat at a data store. Second, insufficient data anonymization is applied or strong data mining is possible in the data store, meaning that the stored information still contains sufficient references to the corresponding data subject, which makes it possible to link different data items to each other within the same database. Another possibility is that data can be linked from one database to another, and hence re-identification [Swe02b] is possible. Note that this tree contains an "and" relationship, meaning that the threat can only occur if there is weak access control to the database, *and* one of the leaf nodes of the weak anonymization branch apply as well.

Linkability of process. The threat tree of *linkability of process* suggests that the only way to prevent different actions being linked to the same subject is by gaining access to the process, as illustrated in Figure 3.6.

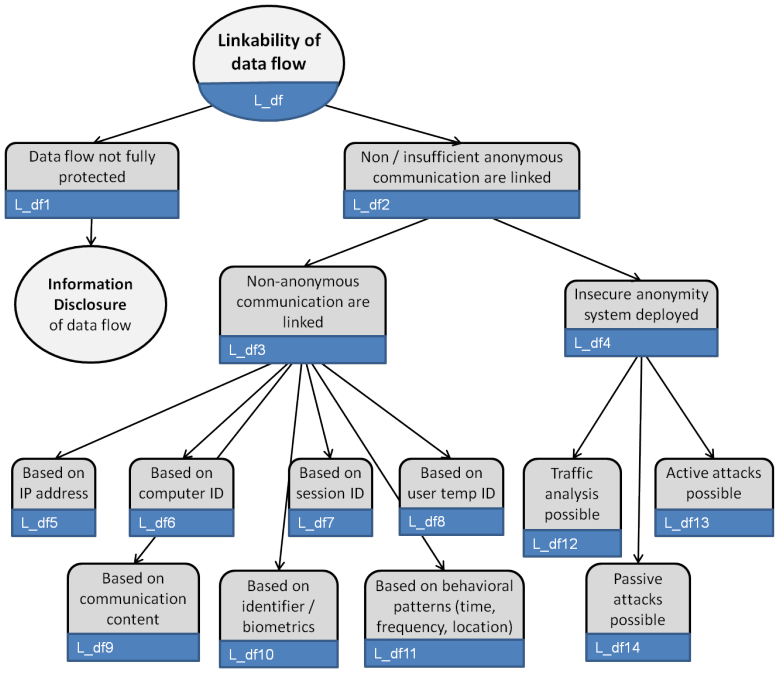


Figure 3.4: Threat tree for linkability of a data flow

Identifiability of entity. Figure 3.7 shows the threat tree pattern for *identifiability of entity*. It gives an overview of the most common situations where an identifiability threat can occur related to an entity. A first precondition is when the e-id is used as login (i.e., the user's actual identity is used), meanwhile the data flow between the user and login portal is not sufficiently protected (i.e., the threat of information disclosure of data flow), and thus the user's identity will be exposed. A second possibility occurs when a secret (e.g. a password) is used as log-in and the relationship between this secret and the user can be revealed. This can happen if the identity management database in not secure (e.g. passwords are stored in clear); if the communication channel is insecure and the communicated passwords are weak and can be connected to the user (e.g. using a birth date as password); or if replay attacks are possible (e.g. a keylogger is installed, the communication can be eavesdropped or the person entering the secret can be observed). A third possible precondition for the threat is the use of a token as log-in which is weakly implemented or physically insecure. The final precondition is that biometrics are used as log-in, which means that biometrics are retrievable and can be linked to an entity. It is due

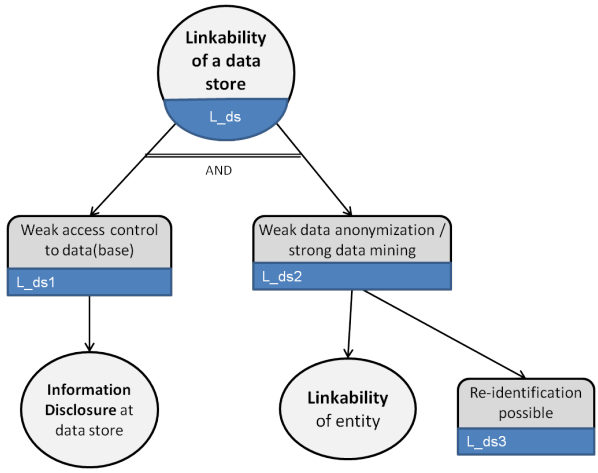


Figure 3.5: Threat tree for linkability of a data store

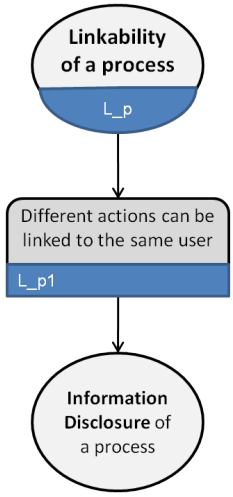


Figure 3.6: Threat tree for linkability of a process

to information disclosure of the identity management database or of the data flow which contains the biometrics, and linkability at identity management data store.

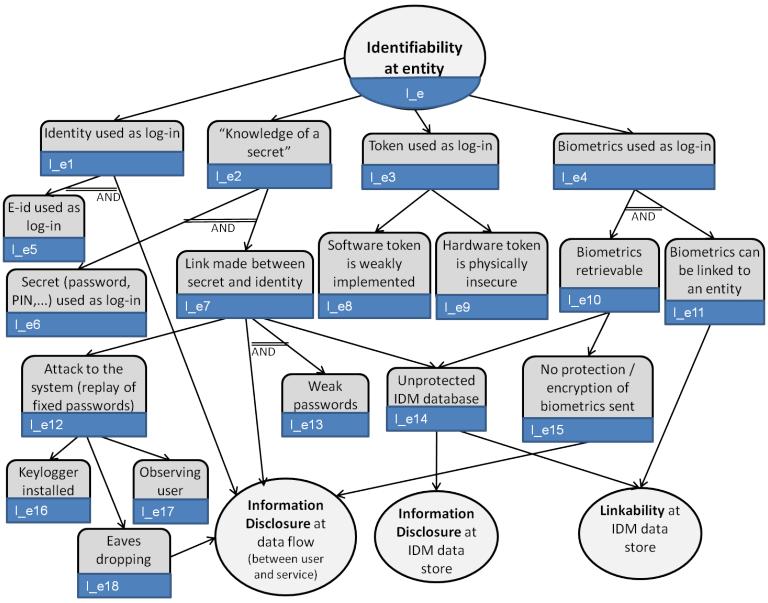


Figure 3.7: Threat tree for identifiability of an entity

Identifiability of data flow. The threat tree of *identifiability of data flow* is presented in Figure 3.8. Similar to the linkability of data flow threat tree, identifiability of data flow is possible when data flows are not fully protected, because of information disclosure threats to the data flow; or when the communication can be traced to an entity due to little or insecure anonymous communication. When no anonymity system is deployed, communication can be traced to an entity by means of identifiable information (e.g. based on user temporary ID, IP address, behavioral patterns such as time, frequency and location, session ID, identifier and biometrics, computer ID, communication content or any combination of these factors). Alternatively, when an insecure anonymity system is deployed, traffic analysis, passive attacks, and active attacks [DDS09] are possible to identify the particularly entity of interests.

Identifiability of data store. The preconditions for *identifiability of data store*, as presented in Figure 3.9, are similar to the preconditions of linkability at a data store. There is insufficient access control of the data store because of information disclosure threat and or insufficient data anonymization or strong data mining techniques were applied in the data store. This weak anonymization means that either the information stored still contains sufficient references to

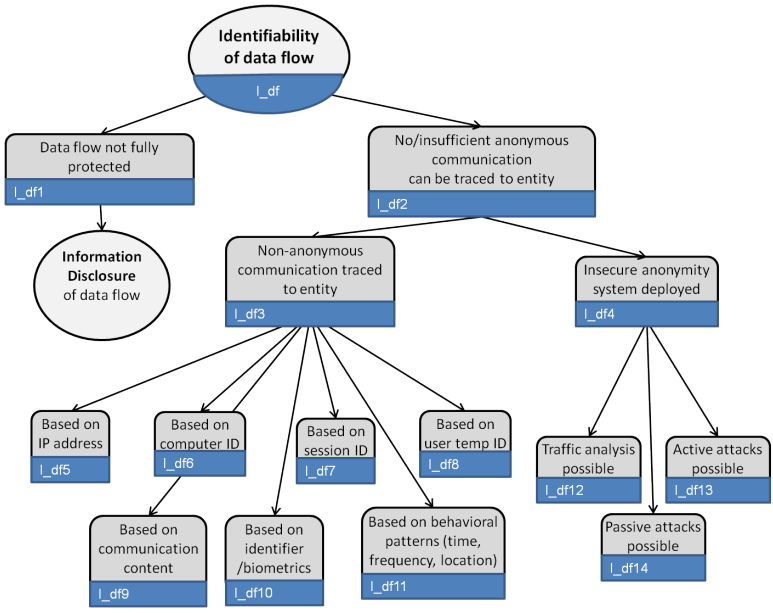


Figure 3.8: Threat tree for identifiability of a data flow

the corresponding person to reveal the person's identity, and hence it refers to the identifiability threat of entity, or the data can be linked with other relevant information which can lead to re-identification of the data subject.

Identifiability of process. The threat tree of *identifiability of process* is presented in Figure 3.10. The only condition for the identifiability threat at a process is when access to the process is not sufficiently secured, and thus it refers to the threat of information disclosure of a process.

Non-repudiation of data flow. Four general preconditions are applicable to the threat tree of *non-repudiation of data flow*, as presented in Figure 3.11. One precondition is insufficient obfuscation for data sources or data flows, which means that the attacker can gain access to at least a part of the data flow or data source. This can occur in a number of cases, for example, there is no automatic replay of broadcasts, such that the sender of a file is sufficiently distinguishable from those who are merely relaying it. Another example is when a complete decrypted log of all network connections to and from a user's computer is disclosed, resulting in the disclosure of the origin of data flow.

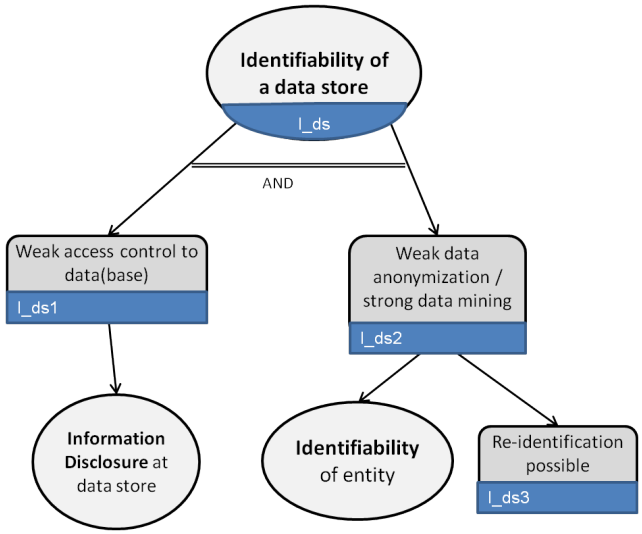


Figure 3.9: Threat tree for identifiability of a data store

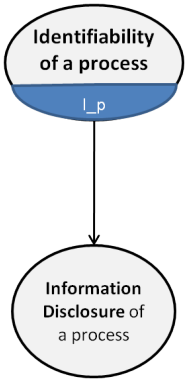


Figure 3.10: Threat tree for identifiability of a process

The final examples are that there is insufficient protection against censors or insufficient obfuscation of data extensions, such that operators or users of the network are able to know where the data comes from.

The second precondition of this threat is that no or weak deniable encryption is used to protect the data flow. One possible attack path is to prove data is encrypted, either due to the encrypter who proves the data is obviously an

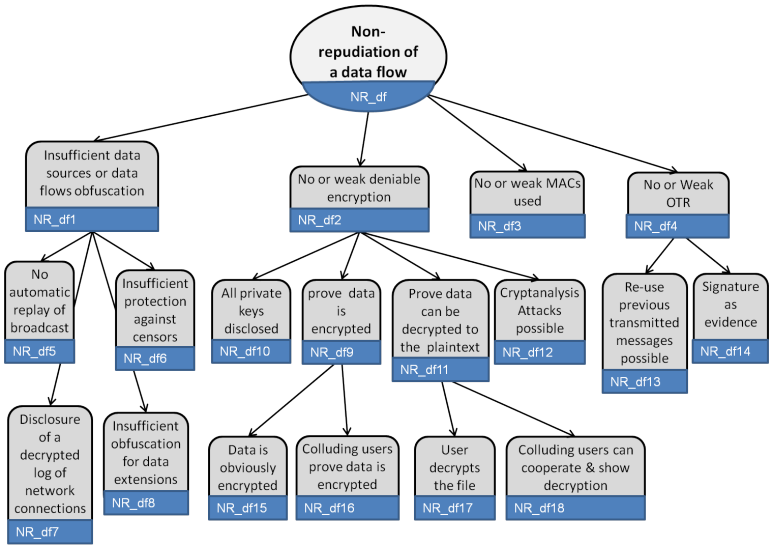


Figure 3.11: Threat tree for Non-repudiation of a data flow

encryption or by colluding users who prove together that the data is encrypted. The second attack path is to prove data can be decrypted to a valid plain text, which can occur when the encrypter decrypts the file or colluding users can cooperate and show the decrypted message. The third attack path shows that all private keys are disclosed, and the last path suggests that cryptanalysis is possible to attack the used encryption scheme.

The third condition is that there are little or weak message authentication codes (MAC) used to ensure integrity of data flow content, such that an attacker can forge authentic looking messages and pretend that a certain data flow comes from a subject.

The final precondition indicates that there is little or a weak Off-the-Record Messaging (OTR) used, such that in a conversation it is not possible to provide both deniability for the conversation participants and confidentiality of conversations content at the same time. Possible attack paths include replaying of previous transferred messages, and the use of signatures to demonstrate communication events, participants and communication content.

Non-repudiation of data store. The threat tree for *non-repudiation of data store* is depicted in Figure 3.12. Three preconditions apply to this threat. First,

no or only weak deniable encryption is used to protect the data, such that data can be proven to be encrypted or can be easily decrypted to a valid plaintext. Second, non-repudiation can occur when there is weak access control to the database, because of information disclosure threats at the data store. Finally, the threat can occur when subjects who want deniability are not able to edit data in the database to cover their tracks, as it is either impossible to remove or alter the subject's own data, or impossible to remove or alter someone else's data concerning the subject.

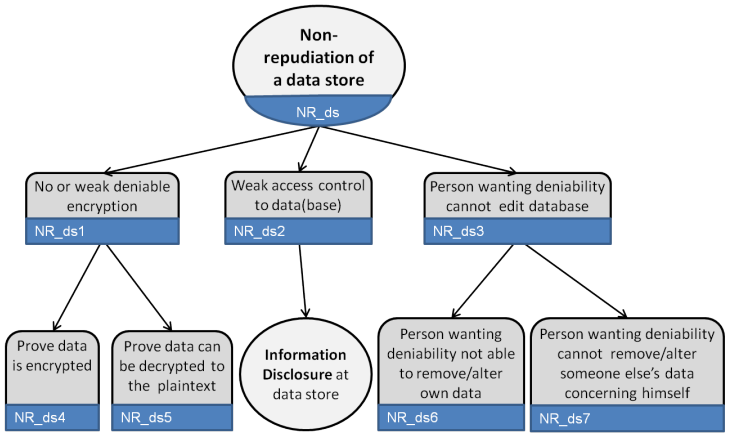


Figure 3.12: Threat tree for Non-repudiation of a data store

Non-repudiation of process. *Non-repudiation of process*, as depicted in Figure 3.13 can be achieved in two ways: either the process looses its confidentiality and information disclosure attacks at the process are possible, or the process uses a secure log to create an overview of all actions, which can evidently be traced back to the user.

Detectability of data flow. The threat tree of *detectability of data flow*, as depicted in Figure 3.14 suggests five preconditions for the threat to occur. One precondition is that the system lacks a covert channel. This can happen when the covert channel uses too much bandwidth from a legitimate channel, resulting in the detection of the covert communication. It can also be because the patterns or characteristics of the communications medium of the legitimate channel are controlled or examined by legitimate users, e.g. checking file opening and closing operations patterns or watching the timing of requests, such that covert communication is detected. The second precondition describes side

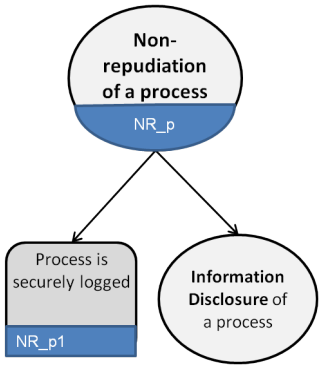


Figure 3.13: Threat tree for Non-repudiation of a process

channel analysis on timing information, power consumption, electromagnetic leaks, etc. as an extra source of information which can be exploited to detect the communication. The third precondition occurs when weak information hiding techniques are used, which enables a number of steganalysis. Another precondition is when there is no or insufficient dummy traffic sent, such that messages fail to appear random for all parties except the sender and the recipient(s). Finally, detectability threats of the data flow can occur because of a weak spread spectrum communication, resulting in deficiencies in the establishment of secure communications, resistance to natural interference and jamming, and detection prevention.

Detectability of data store. As shown in Figure 3.15, *detectability threats in a data store* can occur if there is insufficient access control, which leads to the information disclosure threats for security, or if insufficient information hiding techniques are applied, such that information from a data store is revealed due to weak steganography algorithms employed.

Detectability of process. Similar to the previously described threats related to a process, the *detectability of process* threat, depicted in Figure 3.16, also refers to the threat of information disclosure of a process.

Information disclosure of data flow, data store, and process. The threat tree concerning *information disclosure of data flow, data store, and process*, depicted in Figure 3.17, refers to the security threat tree of information disclosure. This illustrates the fact that privacy properties are part of security properties, and

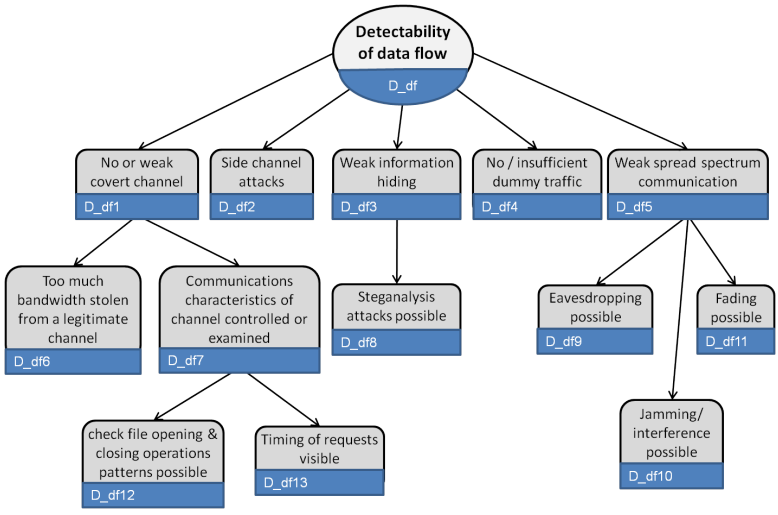


Figure 3.14: Threat tree for detectability of a data flow

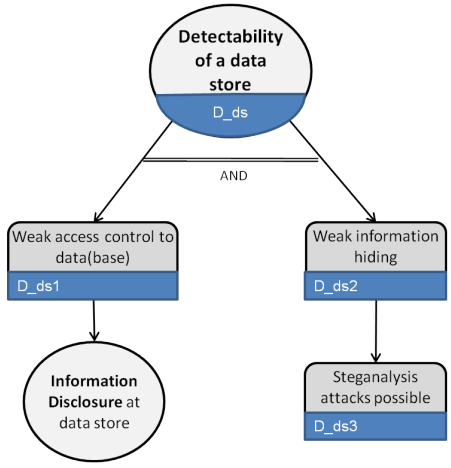


Figure 3.15: Threat tree for detectability of a data store

privacy may depend on security. For more information about the information disclosure threats we refer to SDL [HL06].

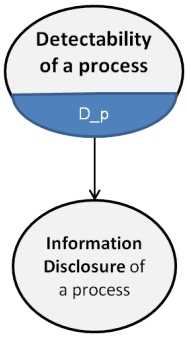


Figure 3.16: Threat tree for detectability of a process

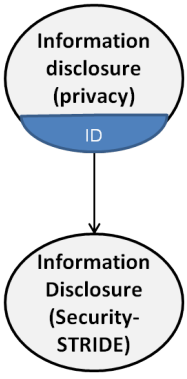


Figure 3.17: Threat tree for Information Disclosure

Content unawareness of entity. *Content unawareness of an entity* can occur in two situations: either the date subject provides more personal identifiable information than required, which has a negative influence on all the hard privacy objectives; or the data subject does not keep the information updated or does not remove (or request removal of)outdated information, which can lead to wrong actions when decisions are based on this information[3].

Consent and policy noncompliance of the system (data flow, process and data store). The user's privacy can be violated when internal system rules do not correspond to privacy policies provided to the user. This can occur when an

[3]Note that the branch related to wrong decisions is removed from the awareness tree in the updated version as described in Section 6.6.7.

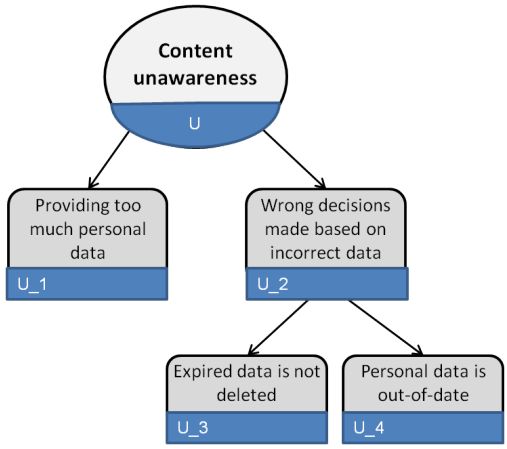


Figure 3.18: Threat tree for Content Unawareness

attacker tampers with the internal policies, which is actually a security threat; or when the policy rules are incorrectly managed or updated (according to the user's requests) by the system administrator.

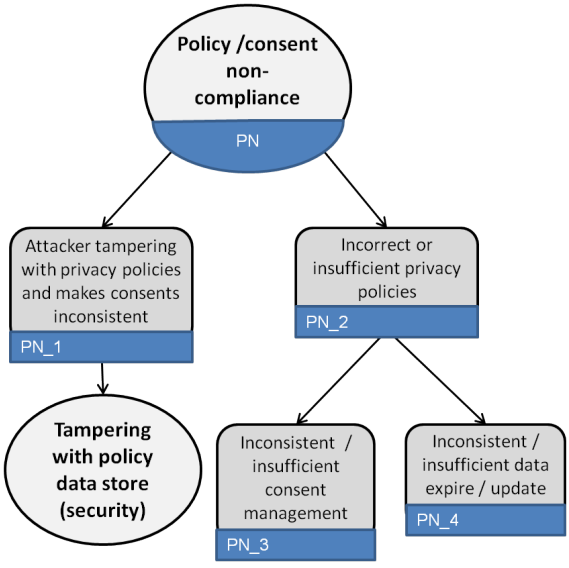


Figure 3.19: Threat tree for policy and consent noncompliance

3.4.2 Document Assumptions

When eliciting threats using the LINDDUN threat trees, often certain leaf nodes or even entire branches will be considered as not relevant to the system-to-analyze and will thus not be considered during the privacy analysis. It is important to also document these decisions.

Assumptions are explicit or implicit choices to trust an element of the system (e.g., human, piece of software) to behave as expected.

Clearly, the analyst needs to make assumptions that are used to reason about whether a threat (or a whole category) is relevant in the system under analysis. These assumed properties or assertions act as domain restrictions. They can for example state that a certain branch in a threat tree does not apply to the system that is being analyzed, hence limiting the scope of the LINDDUN analysis.

The analyst is incentivized to document these assumptions and argumentations as free-form text. Preferably a link to the corresponding misuse case(s) is added for traceability purposes, indicating why a certain alternative path is considered irrelevant or why a certain threat is highly likely to occur. When one of the assumptions is altered in the process, it is important to easily track the required changes in the privacy analysis results.

Running example: Social Network 2.0. In the running example, we have made the assumption that re-identification is not possible in the data store (as shown in step 3 of Figure 3.2), thus eliminating one branch of the threat tree [4].

General assumptions. Note that the documentation of assumptions already starts during step 2 of the LINDDUN methodology. When adding DFD elements, the mapping table, and thus the number of threats to examine, grows exponentially. This number can be limited by making more general assumptions (i.e. spanning an entire threat category or DFD element).

Related to the running example, as shown in the mapping table of step 2 in Figure 3.2, certain threat categories have been considered irrelevant (marked in grey). This limits the number of threats that need to be considered throughout the analysis. Internal DFD elements are, for example, considered trustworthy. Also non-repudiation and detectability threats are considered irrelevant for social networks (after carefully examining the corresponding threat trees). Finally,

[4] Note that the re-identification assumption has only been made for the sake of example. In practice, assumptions should only be made if there is sufficient reason to eliminate a certain threat tree branch. This reasoning should be documented as well.

non-compliance threats should not be applied to a specific DFD element, but are applicable to the entire system (indicated by X* in the table). Again, note that these assumptions are made to simplify the example. In practice, each assumption should be carefully thought through and should be substantiated by adequate reasoning.

These assumptions have a large impact on the privacy results as they eliminate entire threat trees from the LINDDUN analysis. It is hence imperative to explicitly document these assumptions for traceability purposes.

3.4.3 Document Threats using Threat Template

The result of the elicitation process should be a collection of threat scenarios that need to be documented. To this aim, LINDDUN proposes the use of misuse cases [Ale03, SO01, SO05, MF99]. In particular, a misuse case can be considered as a use case from the misactor's point of view. A misactor is someone who intentionally or unintentionally initiates the misuse case. We chose misuse cases because they represent a well established technique to elicit threats.

The proposed misuse case structure, which is based on the template provided by Sindre and Opdahl [SO01] is described below (optional fields are indicated with *):

- *Summary:* provides a brief description of the threat.
- *Assets, stakeholders and threats**: describes the assets being threatened, their importance to the different stakeholders, and what is the potential damage if the misuse case succeeds.
- *Primary misactor*: describes the type of misactor performing the misuse-case. Possible types are insiders, people with a certain technical skill, and so on. Also, some misuse cases could occur accidentally whereas other are most likely to be performed intentionally.
- *Basic Flow*: discusses the normal flow of actions, resulting in a successful attack for the misactor.
- *Alternative Flows**: describes the other ways the misuse can occur.
- *Trigger**: describes how and when the misuse case is initiated.
- *Preconditions**: precondition that the system must meet for the attack to be feasible.

- *Leaf node(s)**: refers to the leaf node(s) of the threat tree(s) the threat corresponds to.
- *Root node(s)**: refers to the root node(s) of the threat tree(s) that were examined for the threat.
- *DFD element(s)**: lists all DFD elements to which this threat is applicable.
- *Remarks(*)*: Although optional, related assumptions should be mentioned here.

The preconditions refer to the leaf nodes of the threat tree patterns and the basic (and alternative) flow describes how an attacker could exploit these weaknesses of the system in order to mount an attack. Assets, stakeholders and threats describe the impact the threat can have. It is an optional field, however it can be useful to determine the risk (step 4 of LINDDUN) as this is based on the impact and likelihood of the threat. Also trigger and preconditions are optional fields as they do not represent the core threat. They can however help reason about the threat.

Leaf nodes, root nodes, and DFD elements are LINDDUN-specific fields which are optional but highly recommended. The leaf and root nodes make it easy to trace the threat to its corresponding threat tree(s) and vice versa. When combining these references with the assumptions, it becomes rather straightforward to determine whether all threat trees and branches have been fully explored. The DFD elements list can be used to group threats. Often one threat is applicable to similar DFD elements (e.g. internal flows, external flows, etc.). To avoid an explosion of threats and obtain a clear, organized overview, these similar threat can be grouped into one.

Finally the remarks field is also optional, although it will usually be filled in with references to the assumptions that correspond to the documented threat.

Note that the LINDDUN-specific fields were added after the original publication of the LINDDUN paper. They were however used during the empirical studies, where they were very helpful both for the participants as well as for the assessors of the studies.

Documenting threats should not necessarily be limited to this step. On the contrary, as privacy threats can already emerge earlier on in the software development process, these threats should be documented as soon as they surface.

Running example: Social Network 2.0. In our running example, we assume that communication and processes within the social network service provider

are trustworthy. However, we want to protect the data store against information disclosure (such as unauthorized access due to a spoofing attack).

The data controllers could be users, social network providers, and application providers.

To illustrate how to create a misuse case based on the threat tree patterns, consider the threat tree of linkability at the data store (see step 3 in Figure 3.2). The threat tree indicates linkability can occur when neither the data store is sufficiently protected against information disclosure nor sufficient data anonymization techniques are employed. These are the preconditions of the misuse case. To create the attack scenarios, it is clear that the attacker first needs to have access to the data store, and secondly, either the user (as the data subject) can be re-identified (as the basic flow) or the pseudonyms can be linkable (as the alternative flow). We have however made the assumption that re-identification is not possible, thus only the threat remains that the data store is susceptible to information disclosure threats and the entity is linkable. If the re-identification assumption would not have been made, the leaf node would have resulted in an additional threat that needed to be documented.

The aforementioned misuse case is presented in this section. The additional nine misuse cases applicable to the social network example are described in [DWS+11].

MUC 1 – Linkability of social network database (data store)

Summary: Data entries can be linked to the same person (without necessarily revealing the person's identity)

Assets, stakeholders and threats*: Personal Identifiable Information (PII) of the user.

- The user:
  - Data entries can be linked to each other which might reveal the person's identity
  - The misactor can build a profile of a user's online activities (interests, actives time, comments, updates, etc.)

Primary misactor: skilled insider / skilled outsider

Basic Flow:

1. The misactor gains access to the database
2. Each data entry is linked to a pseudonym
3. The misactor can link the different pseudonyms together (linkability of entity)

4. Based on the pseudonyms, the misactor can link the different data entries

Trigger*: by misactor, can always happen.

Preconditions*:

- no or insufficient protection of the data store
- no or insufficient data anonymization techniques or strong data mining applied

Leaf node(s)*: L_ds1, ID_ds, L_ds2, L_e2

Root nodes(*): L_ds

DFD elements (*): social network data

Remarks(*):

- re-identification is not considered possible (*see assumption 3*)
- MUC 7 - information disclosure at the data store - is a precondition for this threat

3.5 Step 4: Prioritize Threats

Similarly to STRIDE, LINDDUN can suggest a (large) number of documented threats. Before the process moves forward, the identified threats must be prioritized. Due to time or budget limitations, often only the most important ones will be considered for inclusion in the requirements specification and, consequently, in the design of the solution. Risk assessment techniques provide support for this stage. In general, risk is calculated as a function of the likelihood of the attack scenario and its impact. The risk value is used to sort the MUCs: the higher the risk, the more important the MUC is.

The LINDDUN framework (similarly to STRIDE) is independent from the risk assessment technique that is used. The analyst is free to pick the technique of choice, for instance the OWASP's Risk Rating Methodology [OWA], Microsoft's DREAD [Mica], NIST's Special Publication 800-30 [NIS], or SEI's OCTAVE [CMU]. These techniques leverage the information contained in the misuse cases, as the involved assets (for the impact), and the attacker profile as well as the basic/alternative flows (for the likelihood). Many of the above-mentioned techniques include privacy considerations when assessing the impact of a threat. However, as a research challenge, a privacy-specific risk assessment technique is worthwhile to be investigated, as the on-field experience reveals any inadequacy

of state-of-the-art techniques. This goes beyond the scope of this work.

3.6 Step 5: Elicit Privacy Requirements

Although the core of LINDDUN is problem-oriented, it also provides some guidance in the solution space. Privacy threats are clearly not the final goal of privacy requirements engineering. The elicited threats need to be translated into requirements (i.e. a positive translation of the threats) and should eventually be integrated into the system as privacy enhancing solutions.

The system's positive requirements can be extracted from the misuse cases and the LINDDUN trees to which they are associated. A straightforward mapping of threat categories to positive goals can be found in Table 3.1 when read from right to left. It is however advised to not limit the requirements elicitation to this basic mapping but to elicit more fine-grained requirements that correspond to the applicable LINDDUN tree nodes[5]. Ideally, you should take each of the leaf nodes (or tree branches) into account that led to the threat you are translating into requirements. The causes of the threat (i.e. the leaf nodes) should be mitigated by positive requirements.

Running example: Social Network 2.0. As shown in Figure 3.2, the threat related to linking data in the database corresponds to 2 leaf nodes in the threat tree (i.e. information disclosure of the data store and linkability of entity) and thus should be translated in the 2 corresponding privacy requirements: unlinkability of entities in the database and protection of the data store (confidentiality).

Note that it is possible that certain requirements can (partially) mitigate multiple threats.

3.7 Step 6: Select Privacy Enhancing Solutions

Similarly to security, privacy requirements can be satisfied via a range of solution strategies:

[5]The idea of translating all applicable leaf nodes into privacy requirements is an extension that was added after the publication of the LINDDUN paper.

1. *Warn the user* could be a valid strategy for lower risk (but still relevant) threats. However precautions have to be taken so that users, especially non-technical ones, do not make poor trust decisions.
2. *Removing or turning off the feature* is the only way to reduce the risk to zero. When threat models indicate that the risk is too great or the mitigation techniques are untenable, it is best not to build the feature in the first place, in order to gain a balance between user features and potential privacy risks.
3. *Countering threats with either preventive or reactive privacy enhancing technology* is the most commonly used strategy to solve specific issues.

This section mainly focuses on the last strategy. When countering threats with technology is chosen as the mitigation strategy, system designers have to identify the appropriate privacy enhancing technology (PET). We summarize the state-of-art PETs in Table 3.3 and map these techniques to each of the corresponding privacy properties of Table 3.1. As a result, improved guidance is provided to the system designers over the solution selection process.

Note that the PETs categorization is inspired by the taxonomies proposed in [WSDDJ09, KKG08]. Further, Table 3.3 introduces some key primitives of state-of-art of privacy technologies. New privacy enhancing solutions keep emerging; therefore a complete list of PETs and best practices for choosing the appropriate mitigation is beyond the scope of this thesis. The latest development of privacy enhancing technologies can be found at [PET].

In summary, privacy protection solutions boil down to either technical or legal enforcement. In general, privacy technology enables functionality while offering the highest protection for privacy.

Running example: Social Network 2.0. The threat related to linkability in the database resulted in two privacy requirements (as identified in Step 5): unlinkability of entities in the database, and protection of the data store. Using these requirements, the solution table (Table 3.3 proposes data anonymization techniques such as k-anonymity [Swe02b], and access control techniques such as context-based access control [GMPT01]. The full table of the selected PETs for the running example can be found in the LINDDUN paper [DWS+11].

3.8 Discussion

We conclude this chapter with some reflections on the LINDDUN methodology.

Table 3.3: Mapping privacy objectives with privacy enhancing techniques (U – Unlinkability, A – Anonymity / Pseudonymity, P – Plausible deniability, D – Undetectability / unobservability, C – Confidentiality, W – Content Awareness, O – Policy and consent compliance of the system)

	Mitigation techniques: PETs	U	A	P	D	C	W	O
Anonymity system	Mix-networks (1981) [Cha81], DC-networks (1985) [Cha85, Cha88], ISDN-mixes [PPW91], Onion Routing (1996) [GRS96], Crowds (1998) [RR98], Single proxy (90s) (Penet pseudonymous remailer (1993-1996), Anonymizer, SafeWeb), anonymous Remailer (Cipherpunk Type 0, Type 1 [Bac], Mixmaster Type 2 (1994) [Mixa], Mixminion Type 3 (2003) [Mixb]), and Low-latency communication (Freedom Network (1999-2001) [BGS01], Java Anon Proxy (JAP) (2000) [BFK00], Tor (2004) [DMS04])	×	×			×		
	DC-net & MIX-net + dummy traffic, ISDN-mixes [PPW91]	×	×		×	×		
	Broadcast systems [PW85, WP90] + dummy traffic	×	×		×			
Privacy preserving authentication	Private authentication [AF04, ABB+04]	×	×					
	Anonymous credentials (single show [BC93], multishow [CL04])	×	×					
	Deniable authentication [Nao02]	×	×	×				
	Off-the-record messaging [BGB04]	×	×	×		×		
Privacy preserving cryptographic protocols	Multi-party computation (Secure function evaluation) [Yao82, NN01]	×				×		
	Anonymous buyer-seller watermarking protocol [DBPP09]	×	×			×		
Information retrieval	Private information retrieval [CGKS98] + dummy traffic	×	×		×			
	Oblivious transfer [Rab81, Cac98])	×	×			×		
	Privacy preserving data mining [VBF+04, Pin02]	×	×			×		
	Searchable encryption [ABC+05], Private search [OS05]		×			×		
Data anonymization	K-anonymity model [Swe02b, Swe02a], l-Diversity [MGKV06]	×	×					
Information hiding	Steganography [AP98]	×	×		×			
	Covert communication [MNCM03]	×	×		×			
	Spread spectrum [KM01]	×	×		×			
Pseudonymity systems	Privacy enhancing identity management system [HBC+04]	×	×					
	User-controlled identity management system [CPHH02]	×	×					
	Privacy preserving biometrics [STP09]	×	×			×		
Encryption techniques	Symmetric key & public key encryption [MOV97]					×		
	Deniable encryption			×		×		
	Homomorphic encryption [FG07]					×		
	Verifiable encryption [CD98]					×		
Access control techniques	Context-based access control [GMPT01]					×		
	Privacy-aware access control [CF08, ACK+09]					×		
Policy and feedback tools	Policy communication (P3P [W3C])							×
	Policy enforcement (XACML [oo], EPAL [IBM])							×
	Feedback tools for user privacy awareness [LHDL04, PK09, LBW08]						×	
	Data removal tools (spyware removal, browser cleaning tools, activity traces eraser, harddisk data eraser)						×	

3.8.1 DFD as Model for LINDDUN

The decision to use a data flow diagram (DFD) as model for our methodology was straight-forward. LINDDUN is based on STRIDE, a well-established technique that uses DFD as starting point. Nevertheless, other techniques exist as well to model the system-to-analyze. This section briefly discusses some alternatives and their advantages and disadvantages compared to DFD.

BPMN. One of the best known modeling languages is the business process modeling notation (BPMN) [bpm], which is a flowcharting technique used for creating graphical models of business process operations. It is characterized by flow and connecting objects such as events, activities, gateways and connections. The main difference with DFDs is the system's view it represents. BPMN focuses on the business activities and thereby lacks the overall system model (i.e. which components are connected how). Because this information is actually the core input for the LINDDUN methodology as the mapping table and threat trees explicitly refer to system element types (i.e. data stores, entities, data flows, and processes), BPMN will not be suitable to integrate into LINDDUN without some radical changes. Nevertheless, it might be useful to have, on top of the DFD, a representation of the actual activity flows, such as BPMN, to easily track where certain data will end up. In practice however, we did not yet encounter a system where this additional information was required and could not be extracted from the DFD or the provided system specification.

UML. Another well known modeling language is the Unified Modeling Language (UML) [uml]. It is actually a collection of various diagrams that represent different views of the system. The component-connector diagram is most closely related to a DFD (as is illustrated in the empirical study in Chapter 5). Most of the elements have a one-to-one mapping, and only minor changes are required. Thus using a component-connector diagram as starting point for the analysis will require minimal effort for LINDDUN integration. Other diagrams, such as the activity or sequence diagram can also be used in addition to the component-connector diagram to extend the understanding of the information flows, although this is not required.

Data models. One aspect that is not highlighted by a DFD is the representation of information. As privacy concerns data, it is thus important to have a clear understanding about what data exactly is sent and processed in which parts of the system. The DFD contains annotations indicating what types of data are being communicated via the data flows, however a precise overview

of what these data types contain is not included. In order to perform a detailed privacy analysis, a definition of each data type should thus be available that illustrates the different data elements that are included in the communicated information (e.g. a patient medication entry includes a patient id, a medication id, and a possible link to the prescription). This level of detail allows the privacy analyst to encounter more fine-grained privacy threats, as the specific data elements can imply different privacy risks that might be overlooked when only considered the general information type. No specific representation is required; the data can be represented as entity-relationship model (ERM), using a UML class diagram or can just be sketched. In general this information will already be available in the architectural description (or in a more high-level format at requirements level).

In summary, DFD is considered a sufficient tool to model the information flows throughout the system. A detailed privacy analysis can nevertheless benefit from additional information provided in the requirements or architectural description such as an activity flow or a data model. They should however not be created merely for LINDDUN, but are required in any case to guarantee a successful (functional) analysis of the requirements and architecture.

3.8.2 Coverage of LINDDUN

The threat categories of LINDDUN have not been chosen lightly. An extensive study preceded the creation of the methodology to determine the most common and most pressing privacy threat types.

Concerning the hard privacy threats we mainly focused on Pfitzmann's terminology proposal for Anonymity, Unobservability, Pseudonymity, and Identity Management [PH10] as it is widely recognized in the privacy research community.

We can thus assume that the hard privacy properties have been covered. Nevertheless, we have also verified our classification according to other privacy conceptualizations (which were described in Section 2.1) to ensure coverage.

Solove's Taxonomy. As previously described, Solove's taxonomy makes a distinction between 4 groups of privacy violations. As it is not specific to digital privacy, certain categories are not considered relevant. LINDDUN mainly corresponds to the information processing and information dissemination violations. Concerning information processing, we can see a clear mapping between Solove's aggregation and LINDDUN's linkability threats. Also identification is considered a threat category in both classifications. Exclusion

in Solove's taxonomy can be mapped to LINDDUN's awareness branch related to accuracy[6]. Solove's insecurity corresponds to STRIDE as a whole (which is also referenced in LINDDUN). Finally, secondary use is not considered in LINDDUN explicitly as it is closely related to data protection compliance (only use and share data if the data subject has consented to the specific purpose). The second group of Solove's group that corresponds to LINDDUN is information dissemination. This group can actually be entirely mapped to LINDDUN's information disclosure threats, as Solove's privacy violations either directly concerns disclosure of data (breach of confidentiality, disclosure, exposure), or a possible consequence of it (blackmail, appropriation, distortion). Also Solove's increased accessibility violations can be mapped to LINDDUN's information disclosure threats, as accessibility will impact the risk of information disclosure. Solove's information collection category, which contains surveillance and interrogation violations, does not have a direct mapping. LINDDUN focuses explicitly on how systems can protect the users' privacy, thus we assume legitimate data collection. However, we believe that basic mitigations of information disclosure and spoofing threats (e.g. avoiding eavesdropping, phishing, etc.) can be applied to resolve Solove's information collection violations. Finally, Solove's invasion group is more general and does not always involve actual data. We therefore argue that it is not applicable to the type of privacy analysis LINDDUN aims at. In summary, all of Solove's privacy violations that are applicable to a software system have been included in LINDDUN, together with several other important categories (e.g. detectability and plausible deniability).

PRiS. PRiS [KKG08] is a method for incorporating privacy requirements into the design process by means of patterns. The method has identified 8 high-level privacy requirements categories: identification, authentication, authorization, data protection, anonymity, pseudonymity, unlinkability and unobservability. The first three categories are related to access control and hence covered by LINDDUN's information disclosure threats. The fourth category, data protection, concerns EU data protection legislation, and is thus covered by LINDDUN's non-compliance threats. Finally, the remaining three PRiS categories are also covered by LINDDUN's hard privacy threat categories, respectively identifiability, linkability, and detectability. In addition to the categories of PRiS, LINDDUN also includes threat related to unawareness and non-repudiation.

[6]Note that the concept of accuracy in the context of privacy is slightly updated in the improved version of the threat trees as is discussed in Chapter 6. The updated threat tree nodes will correspond to Solove's taxonomy even more closely.

In-House Privacy Taxonomy. In previous work we have created our own privacy taxonomy [WSDDJ09] in order to classify existing privacy enhancing technologies (PETs) to enable a structured selection of appropriate privacy solutions. The taxonomy approaches privacy from two directions. The proactive branch concerns the concealment of the association (between an identity and his data). This branch is related to LINDDUN's hard privacy properties, with a focus on those related to entity- and flow-related threats. The reactive branch of the taxonomy concerns the guarding of the association (in situations when the association has already been created). This branch corresponds to LINDDUN's soft privacy and security properties, and also includes hard privacy properties related to the data store. All taxonomy categories have been covered by LINDDUN.

Note that in Chapter 6, we will create an even stronger link between LINDDUN and our in-house taxonomy.

3.9 LINDDUN in the Related Work

LINDDUN has been gaining attention in the privacy community. It has been referenced and applied by several researchers and been used as inspiration for other privacy requirements approaches. This section summarizes the most important references.

LINDDUN in Shostack's Threat Modeling book. Adam Shostack is one of the developers of Microsoft's STRIDE. In his latest book [Sho14], which can be considered *the* threat modeling reference work, he describes LINDDUN as "one of the most serious and thought-provoking approaches to privacy threat modeling."

LINDDUN to ensure privacy in call detail records. LINDDUN has been applied in a privacy impact analysis for the analysis of communication records. Hofbauer et al. [HBQ12] introduce a six-step method to ensure privacy in a call detail records (CDR) analysis for voice-over-IP calls. First, they describe the context and stakeholders, after which they identify the personal information in the data. Third, they elicit privacy requirements from legal and cultural aspects. Fourth, they conduct a privacy threat analysis by applying LINDDUN to the CDRs personal information fields that were uncovered in step 2. By applying LINDDUN, they derive a set of privacy-related misuse cases. In the fifth step, the misuse cases are mitigated by proposing suitable privacy requirements and

mechanisms. Finally, they reconcile the legal, cultural and threat driven privacy requirements.

LINDDUN compared to other privacy requirements techniques. Beckers [Bec12] has compared and evaluated PriS, FPFSD and LINDDUN using a conceptual framework for privacy requirements engineering. First he mapped main concepts from the conceptual framework to each of the methods which included: personal information, privacy goal, privacy requirement, specification stakeholder, counter-stakeholder, domain knowledge, anonymity, unlinkability, unobservability, pseudonymity, threat, vulnerability, and risk. Using his framework, he analyzed the three methods which led to the following results. Specific risk assessment techniques are missing in all 3 examined methods. LINDDUN only focuses on privacy requirements and not so much on privacy goals (which are the more high-level descriptions). LINDDUN does however include additional privacy objectives that were not used in the comparison (plausible deniability, confidentiality, awareness, and compliance). Concerning the actual engineering approach, LINDDUN and FPFSD include domain knowledge and assumptions into the process, while PriS doesn't. PriS does however focus on multilateral privacy and explicitly incorporates different stakeholder perspectives. It also considers formality, which LINDDUN and FPFSD do not.

LINDDUN in European research projects. Recently, LINDDUN also has been applied in European research projects to analyze the system's privacy threats. In the Rerum project for reliable, resilient and secure IoT for smart city applications, LINDDUN is applied to the architectural framework for smart cities [ML14]. In the BioMedBridges project, LINDDUN is applied to the architecture created to build data bridges between biological and medical infrastructures in Europe [KLK+14]. In both projects, the security analysis is based on STRIDE, and LINDDUN was chosen as privacy analysis methodology given its complementary nature to STRIDE which enables a closely integrated security and privacy analysis.

LINDDUN in keynote at ACSAC. At the 2011 Annual Computer Security Applications Conference, Susan Landau, a visiting Scholar at Harvard, gave the keynote talk titled 'Privacy: it's All in the Use Case' [Lan]. In her presentation, LINDDUN is used to define real privacy threat categories which are used as guide throughout the talk to discuss federated identity management and RFID tag solutions.

LINDDUN used as basis for QTMM. Luna et al.[LSK12] proposed a quantitative threat modeling methodology (QTMM) to objectively draw conclusions about privacy-related attacks. Similar to LINDDUN, their method is based on STRIDE, and follows the same modeling steps. They however only propose 3 privacy-specific threat categories: linkability, unawareness, and intervenability which is much more limited than LINDDUN. Although intervenability is not explicitly mentioned in the LINDDUN threat categories, we do believe it is, at least partially, included in the awareness and compliance threats. From step 3 onward, QTMM differs from LINDDUN, as the method does not provide any privacy knowledge in the form of threat trees. It does however suggest the use of attack trees in the fourth step of the method (risk-based quantification), where they aim at quantifying the security and privacy risks associated to each element of an attack tree. For each misuse case scenario (for which the template from our LINDDUN paper was used), the corresponding attack tree is created and a DREAD-quantification scale[7] is propagated to the entire tree. Based on these quantifications, the trees can be prioritized. As our method does not provide any specific risk-assessment guidance, we believe (the fourth step of) QTMM can be integrated into the LINDDUN method to provide a more objective prioritization of elicited threats.

3.10 Conclusion

In this chapter, we have presented the LINDDUN threat modeling methodology. It is a model-based approach that provides guidance throughout the entire privacy analysis phase. In addition, it contributes privacy knowledge to aid the analyst in the elicitation of privacy threats.

In order to validate LINDDUN, we have conducted a multi-faceted, empirical evaluation. The following chapters explore how LINDDUN performs in terms of completeness, correctness, ease of use, productivity and reliability.

[7]DREAD is a methodology introduced by Microsoft to rank software bugs based on (D)amage potential, (R)eproducibility, (E)xploitability, (A)ffected users and (D)iscoverability.

4

Evaluating LINDDUN during the Requirements Engineering Phase

LINDDUN is a threat modeling technique that supports the elicitation of privacy threats during the early stages of the software development life-cycle. The first study described in this thesis is an exploratory descriptive study evaluating LINDDUN when applied during the requirements engineering phase[1]. This study was conducted during the eRISE challenge, which provided the infrastructure for the assessment of several security and privacy engineering methods [Uni]. This study enrolled three teams of participants, for a total of 8 individuals. The students came from two different academic institutions and the study was carried out across two locations in Italy and France. Two teams had to analyze the privacy concerns of an e-health system and one team had to analyze a smart grid system. The use of two different application scenarios was a constraint of eRISE. However, the two systems have comparable sizes and

[1]This chapter is based on the first part of the journal paper "Empirical evaluation of a privacy-focused threat modeling methodology" [WSJ14a]

complexity. The description of the two systems to the participants was provided by industrial domain experts, which were available both on site and offline. This is an added value in terms of the realism of the study. All three groups had to apply LINDDUN to elicit the privacy threats of the system they were assigned to. Using the high-level system description, the participants had to create a DFD of the system to be used as the starting point for the LINDDUN analysis. Next, the participants had to follow the first three methodological steps described in Chapter 3. Afterward, the participants had to provide feedback on their experience with LINDDUN by means of a group discussion. We analyzed both the results the participants documented in their reports (discovered threats), as well as the opinions of the participants with regard to their hands-on experience. During this study, we aimed at answering the following 4 research questions:

- *RQ1 - Correctness.* On average, how many threats uncovered by the participants are correct (true positives vs false positives)?
- *RQ2 - Completeness.* How many threats are undetected by the participants (false negatives)?
- *RQ3 - Productivity.* How many valid threats are identified by the participants in a given time frame?
- *RQ4 - Ease of use.* Did the participants perceive the methodology as easy to learn and apply?

This study is rather small in terms of number of participants (three teams). Therefore, no strong statistical significance is expected and the quantitative results are to be considered as exploratory. Nevertheless, we put quite some effort on gathering qualitative observations and received rich and varied feedback from the participants. In this respect, we used a combination of techniques (online forms and group discussion) in order to collect the participants' opinions.

4.1 Planning and Operation of the Study

The descriptive study analyzes the performance of LINDDUN when it is applied during the requirements engineering phase. This section describes both the planning as the actual execution of the study.

4.1.1 Experimental Object

The LINDDUN method was applied to two industrial application scenarios. One team was assigned to a smart grid system, while the two remaining teams had to analyze an e-health system. Both scenarios [Uni] were provided by partner companies, which are considered market leaders in the application domains of the two systems.

Health Care Application Scenario. The Health Care application focuses on the management of electronic healthcare records (EHR). The scenario enables the registration of new patients and assigning them to clinicians, accessing and updating a patient record, retrieving additional patient information from external sources and sharing patient data with authorized external parties.

Smart Grid Application Scenario. The Smart Grid application scenario describes an electricity network using information and communication technology (ICT) to optimize the transmission and distribution of electricity from suppliers to consumers. In particular, the scenario focuses on smart meters to record consumption of electric energy which communicate this information daily back to the utility for monitoring and billing purposes.

More information on both application scenarios can be found on the eRISE website [Uni].

4.1.2 Task

The study was conducted in the context of a European requirements engineering challenge (eRISE, [Uni]). eRISE aims at empirically evaluating security and privacy engineering, including risk analysis methods. It brings together researchers, practitioners and students to determine the effectiveness of security and privacy methods. During the eRISE challenge, 5 methods, including LINDDUN, were evaluated on 2 industrial application scenarios by 42 participants (15 students and 27 professionals) in total. A summary of this overall study is provided in Section 4.2.1. eRISE was conducted in three phases: 1) training phase, where the participants were made familiar with the methods and application scenarios by means of tutorials; 2) application phase, where the participants applied the method they learned to the application scenarios; 3) evaluation phase, where the participants provided feedback through focus group interviews and discussion sessions. Our study will evidently only report on the results obtained for the LINDDUN method.

The participants were asked to perform a privacy threat analysis (cf. first 3 steps of LINDDUN) of the application scenario they were assigned to. The participants were provided with a set of use cases and a general description of the application they needed to analyze and documentation describing the LINDDUN method in detail, including an online catalog containing the LINDDUN threat trees [Uni].

First, the participants had to create a DFD based on the application scenario description. Second, a mapping table needed to be created that clearly maps the DFD elements to the applicable threats. Finally, the mapping table was used as guideline to elicit all possible privacy threats using the LINDDUN threat tree catalog. The participants were encouraged to not only describe each of the uncovered threats using the provided misuse case template, but also to explicitly document all the assumptions they made regarding whether or not certain threats are applicable to the application scenario they were analyzing.

After they completed the LINDDUN analysis, the participants were asked to provide feedback on the methodology during a group discussion session.

4.1.3 Participants

The LINDDUN methodology was applied by 8 participants divided into 3 teams (2 teams of 3 participants, 1 team of 2 participants). In each team, there was one participant who was enrolled in a master course in Computer Science and Telecommunications at the University of Trento. These participants have a background in security engineering and information systems. The other participants were enrolled in a business Master Course in Audit for Information Systems at Dauphine University in Paris. They each have at least 5 years of working experience in the field of information systems auditing.

4.1.4 Design

All teams performed the same task (as described above), however 2 experimental objects were used. One team applied LINDDUN to a smart grid system, while the remaining 2 teams applied LINDDUN to an e-health application.

4.1.5 Research Hypotheses

Table 4.1 provides an overview of the terminology used in this chapter. The reports turned in by the participants have been assessed by experts. Each threat

Table 4.1: Terminology

Term	Meaning
True positive (TP)	Correct threat
False negative (FN)	Overlooked threat
False positive (FP)	Incorrect threat
Precision (Prec)	TP/(TP+FP)
Recall (Rec)	TP/(TP+FN)
Productivity (prod)	TP/time
Reliability (Rel)	$TP_{linddun}/(TP_{linddun} + FN_{linddun})$
Mean (μ)	Population mean

in the report has been marked as either correct (true positive) or, otherwise, incorrect (false positive). Correct results are threats that are a) relevant, privacy related in the context of the specific application and sound with respect to the assumptions documented by the participants, b) compliant to the LINDDUN method and c) documented with enough detail and reasoning (e.g., one-liners are not considered as being enough). Further, we have carried out a LINDDUN analysis of the systems used in the studies. Therefore, we have a 'reference solution' containing all the threats that an optimal execution of LINDDUN would yield. The reference solution was obtained by combining all threat tree leaf nodes with all DFD elements to which the corresponding threat tree applied. Since we trust that solution, we used it to spot whether the participants forgot to identify some threats (false negatives). Note that the hypotheses presented in this section will also be used for the study described in Chapter 5.

Correctness. Instead of using the total number of mistakes made by the participants, we measure correctness by means of precision (see Table 4.1), which scales the errors (false positives) with respect to the number of correctly identified threats (true positives). The rationale is that a participant identifying more threats will naturally incur in more errors. Our null-hypothesis is:

$$H_0^{Prec} : \mu\{Prec \stackrel{\text{def}}{=} \frac{TP_{participant}}{TP_{participant} + FP_{participant}}\} < 0.80$$

That is, our alternative hypothesis (i.e., expectation for a 'good' result) is that precision will be at least 80%, or greater. The selected threshold is the same used in a previous work, which studied a very similar technique (STRIDE) in comparable experimental conditions [SWJ13].

Completeness. We measure completeness by means of recall (see Table 4.1), which scales missing threats (false negatives) w.r.t. the number of correctly

identified threats (true positives). Our null-hypothesis is:

$$H_0^{Rec} : \mu\{Rec \stackrel{\text{def}}{=} \frac{TP_{participant}}{TP_{participant} + FN_{participant}}\} < 0.80$$

As before, our hope (alternative hypothesis) is that recall is at least 80%.

Productivity. We define productivity as the number of correct threats (TP) per hour. Our null-hypothesis is:

$$H_0^{Prod} : \mu\{Prod \stackrel{\text{def}}{=} \frac{TP_{participant}}{time}\} < 1\ threat/h$$

In a related study on security threat modeling (STRIDE), an average productivity of one threat per hour has been observed [SWJ13]. Therefore, our expectation is that LINDDUN performs comparably well, or better.

Ease of use. Ease of use is based on the perception of the participants concerning the difficulty of learning and applying the methodology. We determine the methodology's ease of use by means of the discussion sessions. As this is the first time a larger group of people applies the methodology, we cannot express any meaningful expectation.

4.1.6 Preparation of the Participants

Each application scenario was explained in detail during a 2 hour presentation by the industrial partners. The participants also received a set of use cases describing the system and a high-level sketch of the application. We remark that the two application scenarios had similar size and complexity.

After the presentation of the application scenarios, the participants were given a 3 hour tutorial on LINDDUN. The tutorial consisted of a short introduction to core privacy concepts, the actual LINDDUN methodology, and a step-by-step walk-through illustrating the application of the LINDDUN method to an example scenario. All material used during the eRISE challenge is available online [WSJ].

4.1.7 Execution of the Study

The study, and hence the output actually consisted of two parts. The participants were asked to apply LINDDUN on an assigned application scenario.

Their reports describing the elicited LINDDUN privacy threats was the main assignment. In addition, the second part of the study focused on the evaluation of LINDDUN by the participants by means of an extensive feedback session.

Applying LINDDUN After the tutorials, the participants received their assignment, which asked to apply the first 3 steps of LINDDUN and to identify the threats to the privacy of the system users. Although the teams worked on different application scenarios, they performed the same task. The assignment explicitly mentioned the different LINDDUN steps the participants had to apply and stated what output was expected. As supporting material, the teams were provided with the LINDDUN mapping table and the catalog of threat trees.

The participants spent 13 hours divided over 3 days working on the privacy analysis. During that time, they had to familiarize themselves further with the application, analyze it, as well as prepare 2 presentations about their results (one intermediary presentation and one with the final results). The preparation of the presentations is not part of the LINDDUN methodology. In fact, this is a constraint imposed by the eRISE challenge. After the final presentation, the teams also had to hand in a report that documented in detail the work they had done. The report included the DFD representation of the system, the assumptions they made when including/excluding certain threats, the threats they elicited, and the relative prioritization of these threats. The participants were asked to follow a given template when documenting the threats. The goal of using a template is ensuring that the outcome produced by the different teams would be easily comparable.

Evaluating LINDDUN At the end of the study, the experimenters and the teams gathered together for a discussion session of 1.5 hours.

First a post-it session was organized to reflect on the advantages and disadvantages of the methodology. The participants each had to write 5 positive and 5 negative aspects of the LINDDUN methodology on post-its, then they had to cluster all post-its according to the concepts they characterize and finally they had to prioritize the clusters.

The result was used as guidance for a group discussion which aimed at reviewing all features, both positive and negative, of the LINDDUN methodology. First the methodology process was discussed, second modeling language (i.e. DFD) was reviewed and finally an overall opinion on LINDDUN was discussed.

Table 4.2: True Positives, False Positives and False Negatives of study 1 (requirements phase)

TEAM	TP	FP	FN	Scenario
1	5	2	11	e-health
2	8	1	9	e-health
3	5	1	10	smart grid

4.1.8 Measurement Procedure

We assessed the reports by analyzing each of the applications in light of the assumptions made by the teams and the (sub) systems they selected. Accordingly, we determined whether the threats reported by the teams were correct (true positives and false positives) and whether there were missing threats (false negatives). We therefore applied LINDDUN to each of the application scenarios that were analyzed by the teams to create a baseline and compared them to the participants' results.

In addition to the reports of the study, we also gathered the participants' feedback from the group discussion session to analyze the methodology's ease of use.

4.2 Results

As the number of participants is relatively small, we report the descriptive statistics without attempting any statistical testing.

Correctness and Completeness (RQ1-RQ2) As shown in Table 4.2, on average 6 correct threats (TP) and as little as 1 incorrect threat (FP) were elicited by the participants. However, many threats (10 on average) were overlooked. The overall *precision* is 0.85, which is better than expected (0.80 in H_0^P). The *recall* however is only 0.375, which is much lower than the expectation (0.80 in H_0^R). This low value might be justified by the time constraints imposed on the participants but, this assumption cannot be validated.

Productivity (RQ3) As mentioned, the participants spent a total of 13 hours on the assignment, which results in a productivity of 0.46 threat per hour. This value is roughly 50% below our expectation (1 threat per hour is mentioned

in H_0^{Prod}). In practice, however, some of this time was spent on preparing presentations. Hence the actual productivity could be higher and the above-mentioned value must be interpreted as a lower bound. Unfortunately, our time-tracking data does not allow us to factor out the time spent on the mentioned extra activities.

Ease of use (RQ4) During a guided feedback session, the participants were asked about the LINDDUN process, the creation of the DFD and their overall opinion on the method. They agreed that the process was easy to apply and that the methodology steps are both clear and useful. The participants mentioned that the DFD creation was considered as being the most critical step. Indeed, if the DFD representation is inaccurate, the entire analysis is biased and could lead to imprecise results. Also, the explicit identification of the assumptions was considered challenging as this greatly influences the general outcome of the threat analysis. An assumption can, for example, state that the internal communication channel is considered secure (i.e. confidential and tamper-proof), which eliminates certain LINDDUN threats related to the specific data flow. Clearly, when this assumption is incorrect, some threats will be missed during the analysis.

The participants believed the threat trees were helpful and resulted in threats they would not have thought of. However, they would have preferred a more detailed explanation for each of the reference threats in the catalog. They also worried that using the catalog would limit their creative thinking. A suggestion made by the participants was to first have a brainstorming session in order to elicit an initial set of threats and only later use LINDDUN as a systematic gap analysis. Some participants thought the method was too long, while others argued that, if enough time would be available, the LINDDUN method would provide much more complete results than other techniques (e.g. COBIT[cob]). Finally, there was disagreement among the teams about whether they would have considered using LINDDUN in future industrial projects.

4.2.1 Results of eRISE Challenge

The study described in this chapter was part of the larger eRISE challenge, conducted by the University of Trento [MPP]. The eRISE challenge evaluated six privacy and security requirements methods and examined *how* academic requirements methods actually work when applied by someone different than the method designer, *what* aspects make them work, and *why*. The eRISE challenge was organized in 2011 and 2012. CORAS [LSS11], Security Argumentation [HLMN08] and Secure Tropos [MGM03] were evaluated in both editions. SI*

[GMMZ05] was evaluated in 2011, and SREP [MFMP08] and LINDDUN were evaluated in 2012. Note that LINDDUN is the only privacy method; all other methods relate to security.

The first question the evaluation attempted to answer is: 'Do these methods for eliciting security and privacy requirements really deliver what they promise?.' The participants feedback showed that four of the methodes (CORAS, Secure Tropos, Security Argumentation, and SI*) actually do not deliver what they promise. LINDDUN and SREP, however, do help identify privacy and security requirements respectively. In contrast to the other four methods, LINDDUN and SREP also do not require the analyst to have previous knowledge in privacy or security because the analyst is systematically guided through the identification of privacy and security threats. Participants indicated that the threat trees were useful as they helped them think about threats. One participant even stated that LINDDUN is the first method (s)he ever used that helped to identify threats like anonymity.

Overall, the eRISE study showed that SREP and LINDDUN have been the methods most appreciated by the participants. In fact, the more the participants have applied LINDDUN, the more they liked it.

4.2.2 Threats to validity

As mentioned, only 3 teams participated in this study and no strong statistical conclusions can be drawn.

The participants were introduced to the application scenario they had to analyze by means of a 2 hour tutorial. Although this tutorial was quite elaborate, it is possible that the participants were not completely familiar with the system by the time they started the LINDDUN analysis.

Also, during the LINDDUN tutorial an example was provided to the students. It is possible that the participants used the example as inspiration, which could have biased their results. We however decided that it was more important to provide the participants with sufficient material in order to ensure they fully grasped the LINDDUN methodology.

Finally, using (master and MBA) students might limit the generalization of the results to an industrial context. However, we remark that it is common practice to test new techniques via exploratory studies involving students [CJMS10, Tic00].

4.3 Conclusion

Overall the results of this first descriptive study are promising. The precision rate is slightly higher than expected. The completeness and productivity rate are unfortunately lower than expected, however these results might have been biased by the setup of the experiment. The general feedback on LINDDUN during the feedback session was also mainly positive. We can conclude that LINDDUN is a suitable method for privacy threat elicitation at the requirements level, even though there is still some room for improvement. The results of this study will hence be used in Chapter 6 as input for possible enhancements of the LINDDUN methodology.

Also the qualitative results obtained from the independent eRISE challenge are very optimistic. LINDDUN is one of the most appreciated methods (from the six methods that were examined) and is considered a valuable technique to elicit privacy threats, even without previous privacy knowledge.

5

Evaluating LINDDUN during the Architectural Design phase

This chapter[1] presents two distinct studies related to LINDDUN in the architectural design phase.

The first study is a large descriptive study assessing the application of LINDDUN as an architecture-level threat modeling technique. It has been carried out with the participation of 54 master students organized in 27 teams. The participants had an homogeneous background as they came from the same academic institution in Belgium. They were provided with an elaborate architectural description of a smart grid system, consisting of both diagrams and textual specifications. Note that the participants were already familiar with the application domain as they extensively worked on the domain analysis prior to the execution of the study. Further, during the domain analysis, an

[1] This chapter is based on the second part of the journal paper "Empirical evaluation of a privacy-focused threat modeling methodology" [WSJ14a].

industrial domain expert was present on site in order to answer the questions of the students. In the context of the study, the participants had to create a DFD starting from the architectural description and to perform the analysis steps prescribed by the LINDDUN method. They documented their results (discovered threats) in a report and also answered to the questions of a few questionnaires. This study is similar to the one executed during the requirements engineering phase (as explained in Chapter 4), and therefore examines the same four research questions (i.e. *correctness, completeness, productivity*, and *ease of use*).

The second study of LINDDUN in the architectural design phase is a case study that focuses on the reliability of the results produced by LINDDUN. Indeed, it could be that LINDDUN is flawed and does not guide the analyst to the discovery of important threats. Even an optimal execution of LINDDUN might result in undetected threats due to the limitations of the methodology itself, like an incomplete catalog of threats. In this second architectural study we asked a panel of three privacy experts to perform an independent threat analysis of a smart grid system (at the architectural level). We provided them with a detailed architectural description of the system, and asked them to analyze the system's weaknesses concerning privacy using their own expertise. Their results are then compared to the results obtained by an optimal execution of LINDDUN (i.e. a combination of all leaf nodes and all DFD elements to which the corresponding threat tree applies) on the same application scenario. The LINDDUN analysis is carried out by one of the inventors of the method and reviewed by another senior expert. The comparison mainly focused on the coverage of LINDDUN to identify whether any important class of privacy threats were missing. This allows us to provide an answer for the final research question in this thesis:

- *RQ5 - Reliability.* Is LINDDUN missing any important threats?

With only five experts (three privacy experts and two LINDDUN experts), the size of this case study is limited. Therefore, we do not have the ambition to draw statistically significant conclusions. However, given the knowledge and expertise of the participants, we have the opportunity to gather very valuable observations.

We conclude this chapter with a brief overview of related work and summarize our lessons learned when executing empirical studies.

5.1 LINDDUN as an Architecture-Level Technique: Planning and Operation

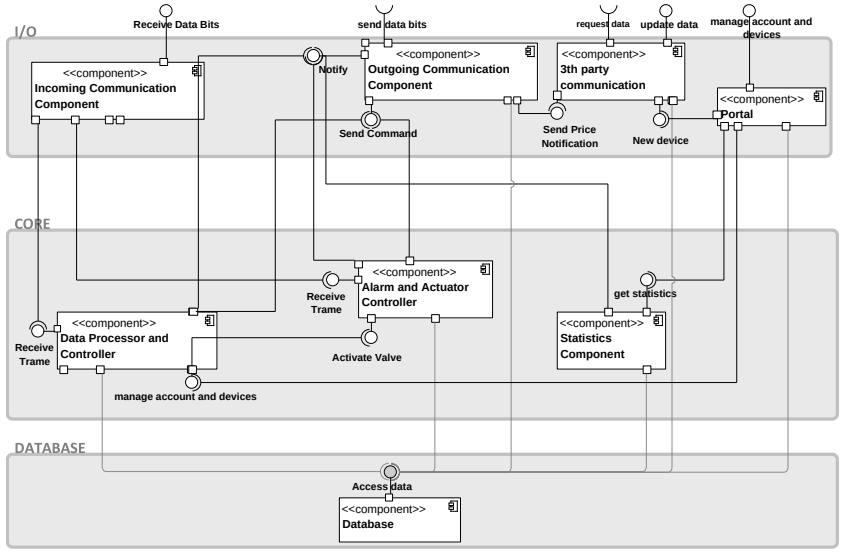
This study evaluates the use of LINDDUN as an architecture-level privacy threat modeling technique. This descriptive study analyzes the results of a set of participants applying the LINDDUN methodology in the context of a software architecture course. The setup was based on Scandariato et al.[SWJ13], as it has proven successful. All material used for the study, and its results are available online [WSJ]. This section will set out the planning and operation of the study, while Section 5.2 will discuss the actual results.

5.1.1 Experimental Object

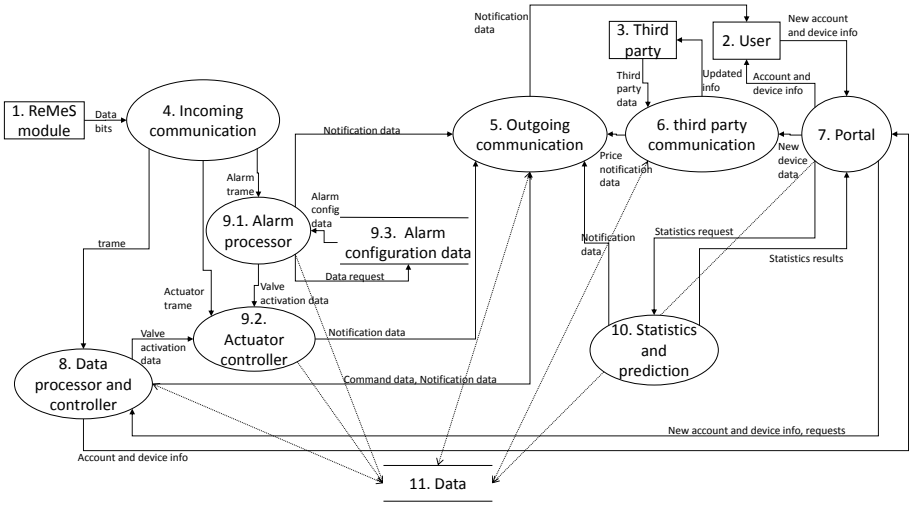
Instead of focusing on projects that are (artificially) shaped toward privacy issues (such as a whistle-blower or anonymous voting system), we opted to analyze a realistic general project: a smart grid remote measurement, monitoring and control system (ReMeS). This project was executed in close collaboration with an industrial partner who is in the process of developing a smart metering system that will be rolled out in the BeNeLux area in the near future. This collaboration guaranteed realistic requirements, which resulted in an interesting (privacy) analysis, especially since smart metering is becoming the next hot topic for privacy researchers [SZAG12]. Although the smart metering case does not have high privacy demands (such as a whistleblowing system or an anonymous browsing technique), it provides some challenging privacy issues which should already be dealt with in the design and architecture of the system [Cav10]. Indeed, in The Netherlands, two smart metering bills which required mandatory introduction of smart meters in each Dutch household were rejected due to insufficient ex ante privacy analysis [CK13].

Smart metering introduces intelligent devices in consumer homes to measure and control consumption of utilities (such as power, and gas). This will, for example, allow utility companies to use up-to-date remote measurements to forecast future consumption. Also consumers can benefit as smart devices can determine the most economical time to consume resources (e.g. when having varying electricity rates, a smart washing machine can be activated at the time of day when power consumption is the cheapest).

The main goal of the system is to bring added value to the companies providing the utility (e.g., gas, water and electricity). This is achieved by reducing their running costs and the administrative overhead related to measuring user consumption. In order to achieve this high level goal, ReMeS provides



(a) The (simplified) component diagram of ReMeS



(b) A sample DFD corresponding to the component-connector diagram

Figure 5.1: ReMeS: a smart grid remote measurement, monitoring and control system

remote measurements, leak and anomaly detection, consumption prediction, and invoicing. The (simplified) component diagram of the system-under-analysis is shown in Figure 5.1a. It is translated to a data flow diagram in Figure 5.1b, which shows just one possible DFD representation. In Figure 5.1b three different DFD elements (two processes and a data store) are derived from the 'Alarm and Actuator Controller' component. The other components are mapped to a single DFD element.

The participants of the study were already familiar with the ReMeS system. Indeed, the study was positioned at the end of the course, and prior to their participation to the study the students had to complete two projects revolving around the ReMeS system. For instance, in one of the projects, they had to specify in detail both the functional and the quality requirements for the system. In the second project, they had to design an architecture for the ReMeS system from the ground up.

To ensure a common starting point to all participants, an architecture for the ReMeS system has been created by the experimenters and provided to the participants. The usage scenarios supported by the system were explained in a 14-page document. Further, a 28-page document explained the main architectural diagrams (e.g., documenting the components-and-connectors view and the deployment view) as well as the main design choices. The document also contained a set of sequence diagrams describing the behavior of the system w.r.t. the key usage scenarios. Finally, the document contained a catalog of all supported interfaces.

5.1.2 Task

The participants have been asked to perform the privacy threat analysis (cf. the first 3 steps in Chapter 3) of the ReMeS architecture provided by the experimenters. In order to complete their task, the participants have been provided with the following supporting material:

- Two electronic documents describing the purpose of the ReMeS system and its software architecture.
- The documentation describing LINDDUN, including the course slides and a research paper.
- An online catalog containing privacy threat trees, which can be viewed at [LIN].

Starting from the provided architectural documentation, the participants had to create a DFD, which is used to guide the rest of the privacy analysis. The participants were instructed on how to map the elements of a component diagram to a DFD. For example, database components correspond to DFD datastores (DS). Users, hardware modules and 3rd party systems components correspond to external entities (EE). Finally, computational components map to processes (PN).

The participants were encouraged to explicitly document all the assumptions they made. Further, they had to describe all uncovered threats as misuse cases, according to a provided documentation template. The template has been used as a means to incentivate the participants to describe their findings with enough detail. This leaves less room for interpretation from the experimenters' perspective.

By participating in the study, the students had the opportunity to earn points for their final score in the course (up to 4 points of the maximum achievable score of 20 points). The rationale was to promote an adequate level of professionalism in the attitude of the students. They have not been judged only on the threats they uncovered, but also on the quality of their documentation (including the assumptions made) and the level of adherence to the LINDDUN methodology. At the end of the study, the participants have been given the option to opt-out from the study. In this case, we graded their work as a regular project and discarded all other measurements that were taken during the study.

5.1.3 Participants

The participants of this study are the students of a course on software architecture taught by the experimenters in the second semester of the first year of the master in Computer Science at KU Leuven in Belgium. We have observed a total of 54 students organized in 27 teams of two students[2]. The background of the participants can be considered rather homogeneous as the course is only available for students of the master program of Computer Science and Applied Informatics. The participants displayed a good level of commitment to the study, as testified by an average grade of 6.3 (out of 10) obtained by the students.

[2]Fourteen more students participated to the study but decided to opt-out at the end of it. Their work has not been included in the study. Eleven more students were discarded by the experimenters because they did not follow the instructions.

5.1.4 Design

All 27 teams performed the same task (as described above) on the same experimental object (the ReMeS system).

5.1.5 Hypotheses

As this study is quite similar to the study of LINDDUN at the requirements phase, we examine the same four hypotheses.

- *RQ1 - Correctness.* On average, how many threats uncovered by the participants are correct (true positives vs false positives)?
- *RQ2 - Completeness.* How many threats are undetected by the participants (false negatives)?
- *RQ3 - Productivity.* How many valid threats are identified by the participants in a given time frame?
- *RQ4 - Ease of use.* Did the participants perceive the methodology as easy to learn and apply?

The reader can refer Section 4.1.5 for further details.

5.1.6 Preparation of the Participants

The study was carried out in a master-level course, as described above. The participants received the training for the study as part of this course. The course focused on software requirements and architecture. It is organized in three parts (requirements elicitation, architectural design, and privacy analysis). The study was conducted during the third part of the course. The preparation of the participants included 2 lectures of 2.5 hours each. The first lecture gave an introduction to privacy concepts and covered the LINDDUN methodology in depth. During the second lecture, an example was given (privacy analysis of a patient community system) to illustrate the LINDDUN methodology in a step-by-step way. In the second lecture, the experimental object (i.e., the software architecture of the ReMeS system) was also presented. Finally, information about the study and the guidelines were given. This information included the participants' right to opt out of the study. Guidelines included the request to precisely track all the time spent in each of the LINDDUN steps using an open-source time tracking tool [kim], to fill in the entry and exit questionnaire,

and to closely follow the provided documentation template for the identified threats.

5.1.7 Execution of Study

After the instruction was concluded and prior to the beginning of the actual study, we asked the participants to fill in an entry questionnaire. The questions were meant to profile the participants with respect to their knowledge of security and privacy, as well as their understanding of both LINDDUN and the ReMeS system. We also asked a question concerning their working experience in the software industry, if any. The participants could fill in the questionnaire anonymously, if so desired.

The actual study took place during 2 supervised lab sessions of 2.5 hours each. The ReMeS system is rather large and complex. Its size is realistic with respect to real world projects. Therefore, the analysis could have not been completed during the two labs only. This explicit choice implied that the participants were expected to start working on the privacy analysis during the lab sessions and finish the assignment as homework. During the first lab session, the teams received their assignment and started working on the task. After the second lab session, they had 2 weeks to complete the assignment at home.

Before the first lab session started, all course material was made available to the participants via the course site. This included the tutorial slides, a step-by-step example, the architectural documentation and the time self-tracking tool. We also activated a web server containing the catalog of privacy threat trees. The catalog was organized in the shape of a web site, with each tree contained in a single page and an index of the trees always available as a sidebar. To access the site, a participant had to log in and we monitored the access of the teams to the web pages.

Both lab sessions were supervised by three teaching assistants who answered the participants' questions and monitored the adherence of the participants to the study protocol. An online discussion board was available during the time of the assignment, so even outside the lab sessions, the participants were able to ask for clarifications. As mentioned, the participants were already quite familiar with the ReMeS system, as they had to deal with it throughout the whole course. This level of familiarity removes some threats to the validity of the study, e.g., related to misunderstanding the experimental object or a steep learning curve. The participants confirmed in a questionnaire that they were very familiar with ReMeS (median of 4 on a scale of 1–strong disagree, to 5–strong agree).

At the end of the study, the teams had to prepare a written report. Each report contained a DFD representing the ReMeS architecture, a list of assumptions (and their rationale), the mapping table between the ReMeS DFD elements and the LINDDUN threat types, and the set of threats the participants identified. The threats appear in the report according to their order of discovery. Further, each threat is labeled according to its level of priority as low, medium or high.

The participants were also asked to keep track of the time they spent. This was repeatedly mentioned by the TAs during the lab sessions, and it was emphasized on the LINDDUN catalog website as reminder. To this aim, we provided a (simple) time tracking tool [kim]. Each team had a user account to log in and could select the type of activity the team was busy with, i.e., one of the steps among 1) DFD creation, 2) mapping, or 3) threat elicitation. The time tracking could be started and stopped with a single click. There was also an option to mention whether the participants were working in group (hence effort time had to be counted twice) or individually at that moment.

At the end of the study, the participants were requested to fill in an online questionnaire concerning the task they executed. The questionnaire was based on one used in a previous descriptive study [SWJ13] and contained 33 questions, most on a scale from 1 to 5, to investigate how easy the steps and the method as a whole were to learn and apply. Also, some open questions invited the participants to clarify their answers and provide feedback. The participants could fill in the questionnaire [WSJ] anonymously, if so desired.

5.1.8 Measurement Procedure

The reports handed in by the teams have been assessed by two experts (the author and a teaching assistant).

The DFD used by each team for the analysis could slightly vary from the provided component diagram, e.g., by adding or removing certain details. Figure 5.1b, for example, decomposes the Alarm and Actuator Contoller component into two more detailed processes and a dedicated alarm configuration data store, however other interpretations are possible as well. In general however, the teams used very similar DFDs as starting point for their analysis as they closely followed our provided procedure to derive DFD elements from a component diagram. For each team, the experts analyzed the documented DFD and assumptions and determined whether each of the identified threats was applicable to the system-under-analysis. Specifically, the experts determined the number of correct threats (TPs) and incorrect threats (FPs) based on the definitions given in Section 4.1.5. That is, correct threats are a) relevant, privacy related in the context of the ReMeS system and sound with respect to the assumptions

Table 5.1: Average size of the DFD constructed by the participants.

DFD ELEMENT	MEAN	STD. DEV.
Entities	6.62	1.05
Flows	53.97	16.49
Data stores	2.31	0.85
Processes	12.73	2.79
Total	75.62	19.09

documented by the participants, b) compliant to the LINDDUN method and c) documented with enough detail and reasoning. Also, the missing threats (false negatives) were counted by looking for LINDDUN threat tree nodes that were not addressed in the report.

About half of the reports were evaluated by both experts and results were compared. Only in a very low number of cases (4% of the 275 compared threats) the results varied, and after a discussion between the experts a consensus was reached. Given this low number of discrepancies and due to time constraints, it was decided to evaluate the remaining half of the reports separately, without further comparison. The experts also measured the size of the DFD in terms of included elements, as well as the number of assumptions.

The time tracking tool collected the time spent per LINDDUN step, however these figures strongly rely on the participants' diligence. Via the logs of the web server containing the threats catalog, we retrieved the page visits for each participant. Finally, we collected the participants' opinion on the methodology by means of the questionnaires.

5.2 LINDDUN as an Architecture-Level Technique: Results

This section describes the results of the descriptive study of LINDDUN when applied at the architecture level. First the descriptive statistics are presented, after which each of the four research questions are evaluated.

Table 5.2: Size of the reference solution per threat category.

Threat category	No. of threats
Linkability (L)	3
Identifiability (I)	2
Non-repudiation (Nr)	0
Detectability (D)	1
Disclosure of information (Di)	7
Unawareness (U)	3
Non-compliance (Nc)	3
Spoofing (S)	3
Total	22

5.2.1 Descriptive statistics

DFD. One of the key aspects of this study was the use of a realistic size project to ensure that the results are also applicable to industry cases. The average size of the DFDs created by the participants illustrates that the ReMeS case study is indeed a medium-to-large scale project with 75.6 elements on average (standard deviation is 19.09). The 95% confidence interval is [68.5, 82] (one-sample Wilcoxon test). Table 5.1 provides the total number of DFD elements, on average, as well as a more detailed overview of each element separately. As expected, the number of data flows (54 on average with a standard deviation of 16.49) has a large impact on the total number of DFD elements (Spearman correlation $\rho = 0.986$ $p < .05$).

Reference solution. A reference solution for the assignment has been created by two experts. It gives the reader a rough estimation of how many privacy threats are to be found in the system-under-analysis. The experts constructed a DFD and applied the LINDDUN method, which resulted in 22 threats, as shown in Table 5.2. Of course, the participants can make different assumptions and each team might have constructed a DFD that differs slightly from the reference solution. Hence, the reference solution is merely used as a guideline for the graders, who still have to carefully examine the reported DFD and assumptions made by each team.

The disclosure of information (Di) threats are the biggest contributor to the overall number of threats. Although this is often considered a security category, the users' privacy cannot be guaranteed without proper confidentiality measures in place. Also, because the impact of disclosure of information threats varies

depending on the type of flow (e.g. internal or external) and the type of data (e.g. credentials, general data or personal information), the number of threats in this category tends to grow easily.

Non-repudiation (Nr) is a category that only applies to some highly specific privacy applications (such as whistle-blowing or e-voting systems), and not to the smart grid system. In fact, for billing purposes any form of plausible deniability should be avoided. Therefore, no threats were elicited for this category.

The spoofing (S) security threats are highly relevant for privacy as they 'support' the privacy threats related to disclosure of information. However, LINDDUN does not deal with spoofing threats directly. Rather, the methodology reuses the threat trees for spoofing defined in Microsoft's STRIDE. In fact, the threat trees in the LINDDUN catalog refer to appropriate threat trees in STRIDE, which have been provided to the participants too as part of the lab material.

5.2.2 RQ1 – Correctness

Figure 5.2 reports the boxplot of true positives (correct threats) and false positives (incorrect threats). We can observe that the number of FPs is much lower than the number of TPs. Indeed, the average number of TPs is 11.75 (with a standard deviation of 3.07) and the 95% confidence interval is [10.5, 13] (one-sample Wilcoxon test). On average, only 5.07 FPs (standard deviation is 2.96) were found, with a 95% confidence interval of [4, 6.5] (one-sample Wilcoxon test).

Figure 5.3 shows the average number of true positives (white bars) and false positives (black bars) for each threat category. Similar to the reference solution, disclosure of information (Di) is the biggest contributor. Non-compliance (Nc) is the second largest contributor with 2 TPs. Non-compliance threats were extensively discussed during the lectures and in the supporting example. Because they are rather generic and can be easily applied to other systems, we believe this led to a relatively high rate of TPs in this category. As non-repudiation (Nr) does not apply to the ReMeS case study (as explained in Section 5.2.1), all reported threats of this category ended up being classified as FPs. No correct threats were reported for detectability. We believe the participants missed this threat because the corresponding category does not often apply to a system. Also, the impact and likelihood of the actual threat (detecting an alarm sent by an individual consumer) are low.

Table 5.3 analyzes the reasons that led the experts to categorize some identified threats as false positives (i.e., incorrect). The majority of the false positives

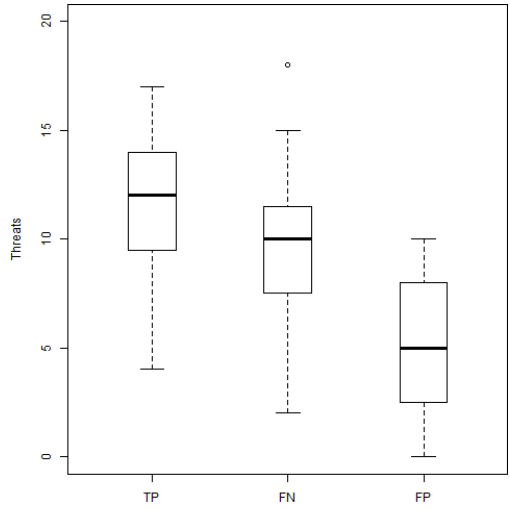


Figure 5.2: Boxplot of true positives, false negatives and false positives. The majority of the reported threats are correct (TP). However, many threats were overlooked (FN)

Table 5.3: Types of False Positives (FP)

FP type	Mean	Std. dev.
Not relevant	4.19	2.55
Not compliant	1.27	1.87

(4.19 threats on average, with standard deviation of 2.55) are irrelevant to the specific application given the explicit assumptions made by the participants or incorrect in general (e.g. not related to privacy). A few FPs were considered not compliant with the LINDDUN method (1.27 threats on average, with standard deviation of 1.87). For instance, it was not possible to link the threat to any of the LINDDUN threat trees, or the documentation was too poor to allow a fair assessment. Note that few threats were categorized as both not relevant and not compliant. Those threats are only counted in the "not relevant" category.

Precision. Although the number of true positives is high, there are also quite some false positives. Hence, the average precision is 0.71, which is not as high as postulated in our hypothesis. Indeed, the null-hypothesis H_0^P **cannot be rejected** ($p > 0.05$, one-tailed Wilcoxon test). However, the number of false positives can be considered acceptable.

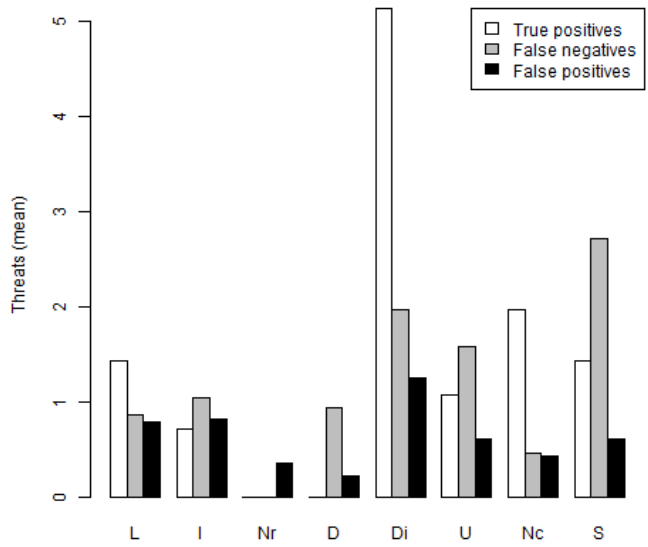


Figure 5.3: True positives, false negatives and false positives per LINDDUN threat category.

Perceived correctness. Half of the participants (55.5%) indicated in the questionnaire that they are confident that all their elicited threats are correct, while only a minority (16.5%) believes not all uncovered threats are correct. Also, little over a quarter (29%) does not know whether their threats are correct. This illustrates that participants tend to overestimate their level of correctness.

5.2.3 RQ2 – Completeness

As previously shown in Figure 5.2, the average number of overlooked threats (FN) is high, especially when compared to the number of TPs. The average recall is only 0.56 and therefore null-hypothesis H_0^R **cannot be rejected** ($p > 0.05$, one-tailed Wilcoxon test). Indeed, the low level of recall is a concern for the LINDDUN methodology. Ways to improve this drawback should be investigated, possibly with a controlled experiment. It should however be noted that LINDDUN's recall results are still better than those of STRIDE (which only has a recall of 0.36)[SWJ13].

For instance, notice that a large number of false negatives come from the spoofing category (S), which is not part of LINDDUN per se. In a way, these false negatives are due to the borrowing of the security threat trees from Microsoft's STRIDE. Without these FNs, the recall would increase by 5 percent points. Also the second largest contributor of false negatives, information disclosure, is a category which is borrowed from STRIDE. This issue deserves further study.

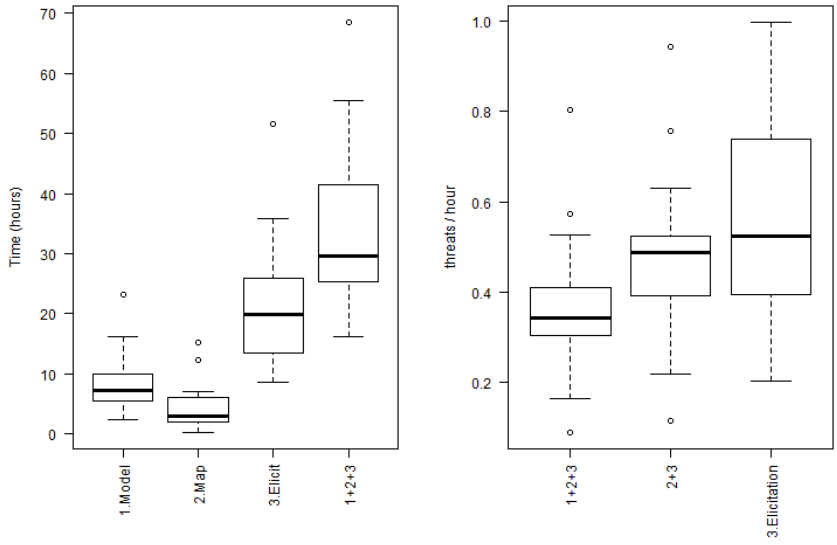
Another relatively large contributor of FNs is unawareness. When looking in more detail at the overlooked unawareness threats, we noticed that they are mainly related to the inaccuracy threats (i.e. outdated or expired information). During the studies, the participants also had problems with grasping these threats, as they are indeed not necessarily privacy related nor system-centric. The unawareness threat tree can thus benefit from an update.

Perceived completeness. Most participants (73%) are convinced they found all important threats, even though they realize that they did not elicit all privacy threats. Half of the participants (58%) think they found between 50% and 75% of the actual threats, while 29% of the participants estimate their level of completeness between 75% and 100%. When comparing these results with the actual recall, we can conclude that the participants tend to overestimate their performance.

5.2.4 RQ3 – Productivity

As shown in Figure 5.4a, the participants spent on average 33 hours per team on the first three steps of LINDDUN (standard deviation is 14.7 hours), of which the majority (21 hours) was spent on the elicitation step, as expected.

To calculate the productivity, we compared the effort time with the actual effort done. Since we are interested in the productivity concerning correct results, we only take the true positives into consideration as effort. On average, the productivity is 0.36 threats per hour (standard deviation is 0.16). with a confidence interval of [0.29, 0.42] threats/h. This means that the productivity is much lower than expected and the null-hypothesis H_0^{Prod} **cannot be rejected**. However, when considering only the elicitation step, the productivity increases to 0.56 threats per hour (standard deviation is 0.22), with a confidence interval of [0.451, 0.663] threats/h.



(a) Effort time: On average, about 33 hours were spent on applying LINDDUN

(b) Productivity: On average, it takes about 2 hours to elicit one correct threat

Figure 5.4: Boxplot of effort time and corresponding productivity

5.2.5 RQ4 – Ease of use

To evaluate the ease of use, we asked all participants to fill in a questionnaire after the assignment was completed. This questionnaire aimed at determining the participants' perception of the LINDDUN method.

Perceived difficulty. As shown in Table 5.4, the difficulty of mastering the LINDDUN method was considered average (3 on a scale from 1 to 5). Overall, applying the method was considered average (44%) to hard (39%). Both the creation of the DFD and the mapping were rated with a medium difficulty. However, the elicitation and documentation of threats was perceived as hard. When asked for a clarification, the participants indicated that the elicitation was hard because the threat tree description was too limited. Also, determining the assumptions that apply to the system-under-analysis resulted in a perceived difficulty of the elicitation step.

Table 5.4: Difficulty and usefulness perceived by LINDDUN adopters. The scale goes from 1 (low) to 5 (high).

Difficulty of activity	Median	Conf. int.
The overall assignment	3	[3.5, 3.5]
Learning the LINDDUN method	3	[3, 3.5]
Applying the LINDDUN method	3.5	[3.5, 3.5]
Creating the DFD	3	[2.5, 3]
Mapping threat categories to DFD elements	3	[2.5, 3]
Eliciting the privacy threats	4	[4, 4]
Documenting the threats using misuse cases	4	[3.5, 4]
Usefulness of activity	Median	Conf. int.
Creating the DFD	4	[3.5, 4]
Mapping threat categories to DFD elements	3	[3, 3.5]
Documenting the threats using misuse cases	3	[2.5, 3]

Table 5.5: Feedback on the catalog of threats. The scale goes from 1 (strongly disagree) to 5 (strongly agree).

Question	Median	Std. dev.
It was useful to have a catalog of threats	4	0.73
The catalog limited my creative thinking	3	1.09
I think the threat catalog contains enough elements in order to do a thorough analysis	4	0.79
The description of the threats in the catalog is clear	2	0.95

Usefulness of each LINDDUN step. As shown in the second half of Table 5.4, creating a DFD was considered useful by the participants (4 on a scale from 1 to 5), while no strong opinions concerning the usefulness of the mapping of DFD elements to LINDDUN threat types were provided (median answer is 3).

When asked why the participants did (not) find it useful to document the threats by means of misuse cases, some indicated that the template helped them find which part of the threat was not yet documented. Others mentioned that describing the LINDDUN (sub)tree(s) to which the threats correspond was annoying. While, in practice the description of the threat trees is not strictly required, it was mandatory for the participants as it helps during the evaluation to trace back which trees and child nodes they consider relevant to the system.

Feedback on the catalog of privacy threat trees. As shown in in Table 5.5, we also asked feedback about the catalog of privacy threat trees that were provided. On average, the participants found the online catalog useful (4), and thought the catalog contained enough elements for a thorough analysis (4). They however did indicate that the threat descriptions in the catalog were not clear (2). Answering the open question about what was missing from the catalog, the participants repeated that a more detailed explanation of each tree node would be useful, and maybe also some examples could be helpful. The extension of the catalog description is therefore a key change in the improved version of LINDDUN (as will be described in Chapter 6).

5.2.6 Threats to Validity

The time participants spent was self-reported rather than measured by the experimenters. This might have led to error-prone results, although we strongly encouraged the participants to keep track of the time as precisely as possible and reminded them often to do so. We also provided them with an easy online time-tracking tool to lower the overhead as much as possible, which was also confirmed by the perceived user-friendliness of the tool (4 on a scale of 1–very user-unfriendly, to 5–very user-friendly). We decided not to create a controlled environment where the participants are tracked more closely, as students in general perceive this as too intrusive.

Also, during the LINDDUN tutorial an example was provided to the participants that had all the LINDDUN steps worked out and documented. As this example is part of the instruction material, it is normal for a novice to reuse it. However, the student attitude can lead to an exact copy of the example, rather than using it as guidance. We therefore eliminated 6 teams from the final results as they clearly copied the example. Nevertheless the threat remains that other teams might have followed a similar approach without us noticing.

Concerning maturation, it is possible that the participants become tired or bored of the rather large task they have to complete. Participants might have also shared results with other teams, although no cases of plagiarism were detected.

Finally, concerning the generalization, the results of this study might be specific to the object and subjects we used. Therefore, the results might not generalize to the analysis of other systems and to real-world privacy analysts.

5.2.7 LINDDUN compared to STRIDE

In a previous study, we have also evaluated STRIDE's completeness, correctness and productivity [SWJ13]. In fact, the operation and execution of LINDDUN's descriptive study were based on the study of STRIDE. As LINDDUN is inspired by STRIDE, it can be useful to compare the results of both studies. Keep in mind that, although the setup is comparable, the two studies were executed independently, thus no statistical conclusions should be drawn.

Table 5.6 summarizes the results of both studies.

Table 5.6: LINDDUN study compared to STRIDE.

	LINDDUN	STRIDE
Precision	0.71	0.81
Recall	0.56	0.36
Productivity	0.5 threat/hour	1 threat/hour

Correctness. The precision of the results in the STRIDE study is 0.81. The precision of LINDDUN is slightly lower with 0.71, although note that the precision of LINDDUN when applied at the requirements phase (see Chapter 4) is 0.85. The precision results are thus comparable.

Completeness. Regarding the completeness, LINDDUN results show a recall of only 0.56, which is lower than expected. STRIDE however even performs less with only 0.36.

Productivity. STRIDE results in 1 threat per hour, which seems a rather low productivity. LINDDUN unfortunately performs even worse productivity-wise with only half a threat per hour. This might be attributed to the nature of the type of threats. In general, analysts are more familiar with security concepts than with privacy, which likely introduces a steeper learning curve.

Baseline. There is a large gap between both studies regarding the number of threats described in the baseline. STRIDE's baseline (214 threats) is in fact a factor of 10 larger than the reference solution of LINDDUN (22 threats). These two numbers should however not be directly compared. In the study of STRIDE each individual threat was counted (i.e. each DFD element was evaluated

separately), while for the LINDDUN study, the concept of reduction was applied. When a threat is applicable to a number of similar DFD elements, they can be combined. This idea of reduction is actually encouraged as the number of threats can quickly grow exponentially, as illustrated by the baseline number of STRIDE. Also, combining similar threats into one description facilitates an easier and clearer navigation throughout and general understanding of the threat elicitation document.

For easy comparison, we also calculated the number of threats in the LINDDUN baseline without applying reduction by multiplying each elicited threat with the number of DFD elements to which it applies. Without reduction, the size of the baseline is clearly proportional to the size of the DFD. When applying LINDDUN to the ReMeS DFD as depicted in Figure 5.1b, the baseline corresponds to 160 threats. Note that even this simplified ReMeS DFD contains more elements than the average DFD size of the STRIDE study (i.e. 25 DFD elements), but yet generates less baseline threats. We believe that the origin of this discrepancy between the LINDDUN and STRIDE baselines is twofold. First, the difference is due to the nature of the application. The STRIDE study analyzed an application that was security-sensitive, while the ReMeS application used in the LINDDUN study had no high privacy demands. Second, the STRIDE threat trees are, in general, more extensive than those of LINDDUN. In fact, the threats borrowed from STRIDE were a large contributor to the LINDDUN baseline. Information disclosure threats of the data flow alone resulted in 15% of the LINDDUN baseline. Also, note that the largest contribution by far in the LINDDUN baseline came from the non-compliance threats. As they apply to the entire system, they were multiplied with all DFD elements, resulting in 40% of the baseline.

It is possible that STRIDE's low completeness rate is due to this unreduced baseline, which likely causes a larger set of missing threats compared to LINDDUN. However, even though LINDDUN reduces its threats, the participants still had to explicitly list all DFD elements to which the elicited threat applied. When one or more DFD elements were missing, they were also listed as a false negative.

We can thus conclude that, overall, LINDDUN and STRIDE perform comparable. LINDDUN is less burdened by an explosion of threats than STRIDE. When considering non-compliance only once (as it suffices to apply it to the system in general), LINDDUN would lead to a very manageable size of threats even without reduction.

5.3 LINDDUN compared to Privacy Experts

The previous studies focused entirely on the characterization of the LINDDUN method, e.g., by analyzing how the participants performed on average with respect to an ideal, optimal execution of the methodology itself. These studies provide quantitative evidence of how the methodology would perform once it is placed in the hands of the analysts. This final study examines the reliability of the LINDDUN outcome. Due to limitations of the methodology, it could be impossible to discover certain important threats, no matter how well the methodology is applied. To this aim, we asked 5 experts to perform the threat analysis of the ReMeS system used in the previous study. Note that this case study is merely an illustration because, given the small number of participants, no statistical relevance can be guaranteed.

Privacy experts. Three individuals are researchers with several years of experience in the field of privacy and hands-on experience with threat modeling: one is a postdoctoral researcher, two are PhD students. They were given the architectural description of the system (same as in study 2) and asked to identify as many privacy threats as they could. The experts could refer to the experimenters in order to ask detailed questions about the ReMeS system, should they had doubts about the purpose of the system, its functionality, or any other technical concern. To facilitate a uniform comparison, the three experts have been asked to document the threats according to a common template. Further, we provided them with the description of 3 example threats (unrelated to the ReMeS system) in order to exemplify the level of detail we expected from them. After having familiarized themselves with the ReMeS system, the three experts worked individually (i.e., without any contact) and used their own expertise in order to carry out the task. None of them were familiar with LINDDUN and, hence, their work is not biased.

At the end of their analysis, which lasted about a week, the results have been collected by the experimenters. Each of the privacy experts had to hand in a report that listed all their elicited threats, which were documented using the provided template. They also individually discussed their results with the experimenters in order to rectify any ambiguity in their reports.

Methodology experts. Two individuals are experts in the use of LINDDUN: one is the author, the other is a postdoctoral researcher in the field of security. We underscore that both individuals have a deep understanding of the ReMeS system. The first expert performed an independent threat analysis of ReMeS and submitted the results to the second expert who reviewed them. After a joint discussion to clarify any disagreement, the two experts documented the identified threats in a consolidated report. Incidentally, this report is the same

Table 5.7: Number of accepted and rejected threats

PRIVACY EXPERT	ACCEPTED	REJECTED	TOTAL
Senior	13	1	14
Junior 1	8	3	11
Junior 2	9	1	10
Total	24	5	29

that has been used as a baseline in the previous descriptive study.

5.3.1 Hypotheses

In this study, we are investigating *reliability* (research question RQ5) by evaluating whether LINDDUN does not miss important threats that would be otherwise discovered by privacy experts.

We therefore introduce a fifth and final research question.

Reliability. The results provided by LINDDUN and by the privacy experts are compared to identify gaps. We want to assess the size of the gaps (how many more threats the experts find).

Our null-hypothesis is:

$$H_0^{Rel} : \mu\{Rel \stackrel{\text{def}}{=} \frac{TP_{linddun}}{TP_{linddun} + FN_{linddun}}\} < 0.80$$

The false negatives due to the methodology are the threats identified by the privacy experts and missed by an optimal execution of LINDDUN. The reader will notice that the definition of reliability is akin to that of recall (see Table 4.1). Hence, for consistency, we used the same threshold of 80% Thus our expectation is that LINDDUN finds at least 80% of the total number of existing threats. If so, LINDDUN supports non-experts "sufficiently" in order to let them reach a privacy-expert level in their analysis results. Moreover, if the expectation is met, privacy experts could rely on LINDDUN as a checklist that avoids gaps in their analysis activity.

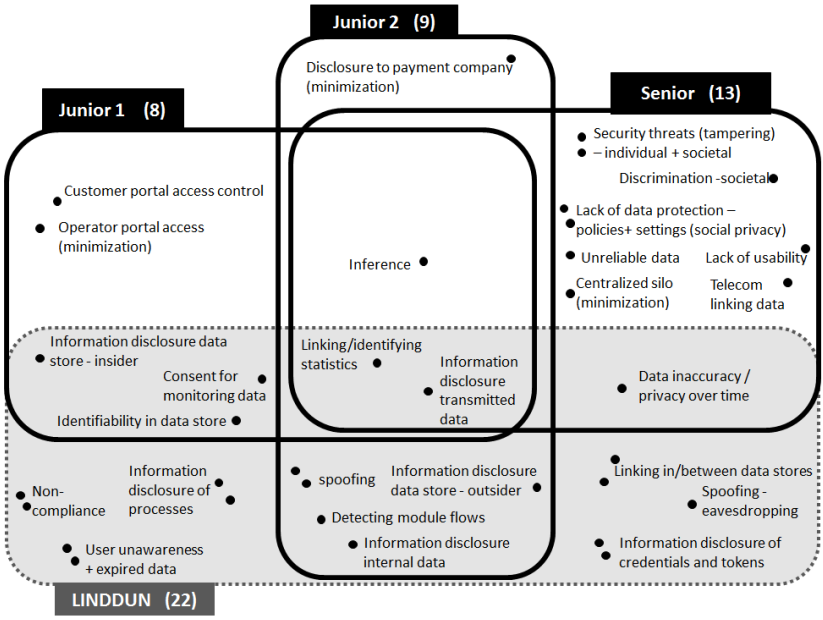


Figure 5.5: Threats identified in study 3: privacy experts (top) and LINDDUN (bottom)

5.3.2 Experts Make (few) Mistakes too

The authors independently analyzed the reports of the three privacy experts and decided which threats should have been discarded. Table 5.7 shows the number of rejected threats per expert. Most of the rejected threats are duplicates of other threats reported by the same expert. For instance, some threats were special cases of more general threats the expert had already identified. In a couple of cases, instead, the threats were rejected because they were unrealistic.

5.3.3 Experimental results

The threats identified by the LINDDUN experts have been already presented and discussed in Section 5.2.1 (reference solution in study 2). Here we focus on the differences with respect to those identified by the privacy experts. Figure 5.5 shows a Venn diagram. An intersection means that the corresponding threats have been identified by two or more parties. In particular, the top part of the figure shows the result of the privacy experts (solid-line sets) and the bottom

parts shows the results obtained by the methodology experts using LINDDUN (dashed-line set).

From a quantitative perspective, the methodology experts have identified 22 threats ($TP_{linddun}$). The privacy experts have identified 13 extra, unique threats ($FN_{linddun}$). This yields to a reliability of 63%, which is much lower than the expectation stated in the null-hypothesis. Therefore, we cannot consider LINDDUN as a very reliable methodology w.r.t. the benchmark we have defined. However, the analysis of the type of threats missed by LINDDUN ($FN_{linddun}$) reveals how to proficiently improve the reliability of the method.

Experts go different directions. In Figure 5.5, the solid-line sets (privacy experts) have significant non-overlapping parts. This means that the privacy experts produced very different results from each other. The main difference is observed concerning the type of identified threats: while the junior experts focused more on the concerns of individual users, the senior expert also took the societal harms into account. There are some differences in the scope of the analysis carried out by the two junior experts. Junior 1 focused mainly on the anonymity issues related to the user access to the system's portal and the storage of user information in the system databases. Instead, Junior 2 focused on the disclosure of user information with respect to the communication channels with 3rd party services.

Acceptable coverage of LINDDUN. Only three threats were identified by all experts: identifying statistics, information disclosure of transmitted data, and data inference. As all experts pointed in this direction, the corresponding threats are important for a comparison with LINDDUN.

Of these threats, two are also spotted by LINDDUN, while the threat related to inference (i.e. deriving additional information based on available, and possibly anonymized data by means of logic reasoning, observing patterns, etc.) is missed. Furthermore, each expert has identified one (different) threat related to minimization of data.

The societal threats identified by the senior expert are clearly also important, but they are considered less critical for LINDDUN, as the methodology focuses mainly on privacy threats to individuals.

Overall, LINDDUN reached an acceptable coverage with respect to the privacy experts.

Identified gaps. This case study identified two areas of improvement for LINDDUN: data minimization and data inference threats. Inference was identified by all three experts (as shown in the intersection of Figure 5.5). Data minimization was also elicited, however the overlap between the three expert threats was less obvious. In general, the concept of data minimization refers to the sharing of the minimal required set of information (as described by data protection legislation), however it is actually much broader as it also relates to minimizing data collection, storage, replication, etc. In particular, Junior 1 describes a threat where an insider (operator) has access to too much data. Junior 2 identified a threat where a 3rd party company receives more information than required for its tasks. The senior expert has described a scenario where data is collected and stored in a centralized silo instead of having local storage. None of these threats are covered by LINDDUN.

All privacy experts considered these concerns as important, which is a key aspect to keep into consideration. We could argue that inference (not considered in LINDDUN) and linkability (considered in LINDDUN) are very related, since unlinkability solutions can prevent inference. However, we agree that the concept itself is missing from the LINDDUN threat trees. Similarly, minimality is not made explicit in LINDDUN. Therefore, it is advisable that LINDDUN would incorporate data minimization (or the lack thereof) and data inference in the catalog of threat trees provided by the methodology. Incidentally, this improvement is well attainable without compromising the structural cohesion of the methodology, as will be discussed in Chapter 6.

LINDDUN also useful for privacy experts. Even with the combined effort of three privacy experts, as many as 11 threats (31% of the total) have been overlooked. These threats, instead, have been identified by a proper execution of LINDDUN. This might even signify the worth of the methodology as such, even for senior analysts. In fact, we would advice also privacy experts to take advantage of LINDDUN as a post-hoc gap analysis tool.

5.3.4 Threats to Validity

As this is only an exploratory case study, we are aware that there are many threats to validity and no strong conclusion can be drawn.

5.4 Related Work

Our empirical evaluation is actually a rather innovative approach as only a small set of previous empirical threat modeling studies in the context of privacy and threat modeling are known to the author.

STRIDE. Scandariato et al. [SWJ13] executed a descriptive study to empirically validate Microsoft's STRIDE. They performed the study twice (2 years in a row) on a group of master students enrolled in a software architecture course. The study investigated the productivity, correctness, and completeness of the STRIDE methodology. With an output of .81 precision rate, and .36 recall, and a productivity of .9 threat per hour, the STRIDE method provides acceptable results, although some improvements (especially related to completeness) are advisable.

Attack trees vs. misuse cases. Opdahl and Sindre [OS09] compared attack trees to misuse cases to determine the preferred (security) threat identification technique. They performed a pair of controlled experiments where the participants (master students enrolled in a course of system analysis and design) used the two techniques individually of two different case studies following the Latin-Squares experimental design. The experiment evaluates both the participants' performance and perception for both methods. Attack trees were identified as most effective, but the authors indicate that the techniques could also be complementary.

Reusing misuse cases. Meland et al.[MTJ10] compared two types of threat model reuse: using existing misuse cases (MUCs) as source for creating your own MUCs and using categorized threats and MUC stubs as a source for creating your own MUCs. They also applied the Latin-Squares experimental design and had 2 groups of participants applying the two techniques on two different cases. No significant difference was found between the two methods and participants indicated that both methods were easy to learn and use and were helpful in identifying threats they would not have identified otherwise.

Engineering of Risk and Security Requirement Challenge. Also, although not specific to privacy or threat modeling, Massacci and Paci [MP12] investigate how successful academic security requirements methods are when they are being applied by master students in computer science and professionals. The authors present a qualitative study where they compare the results of CORAS, Secure

Tropos, Problem Frames, and SI*. The study was conducted during eRISE 2011 (the predecessor of the study that LINDDUN participated in, as is explained in Chapter 4). It consisted of a training phase to get the participants familiar with the security method and the application scenario they were evaluating, the application phase where the participants actual applied the method to the application and concluded with the analysis phase where the organizing team analyzed and evaluated the methods. The organizing team focused mainly on interviews and questionnaires to evaluate the different methods. They analyzed the participants' appreciation of the methods, and determined the most liked conceptual model (SI*), the preferred method concerning analysis capabilities (CORAS), and the method with the best tool support (CORAS). They also discovered that each method led to the identification of different requirements categories. CORAS and Secure Tropos analysis resulted in availability, confidentiality and integrity, while Problem Frames focused on integrity and confidentiality, and SI* mainly focused on integrity and privacy.

5.5 Lessons learned

This section provides a brief overview of the lessons we learned when executing the empirical studies. They are mainly based on the descriptive study in the architecture phase (Section 5.2.1), but are applicable to all studies in general. These lessons do not have any statistical value but originate from our own experience and observations during the study.

5.5.1 Open vs. Restricted Environment

A first decision we made was to provide the participants with a rather free environment. Rather than requesting participants to spent a couple of full working days in an isolated room working on pre-installed computers which are constantly being monitored, we gave the participants much more freedom. As the participants were students, and the study took place during a course, we did not want to restrict the participants too much. We only provided them with a time tracking tool which they had to use whenever they were working on the project. As it was user-friendly, it did not cause any overhead. We also logged all access to the LINDDUN threat tree catalog. This catalog monitoring was also clearly communicated to the participants.

The downside of this freedom is the possible untrustworthiness of the time tracking and access log data. We encountered two groups of participants who deliberately tampered with the access logs of the catalog by having a script

accessing certain threat tree pages hundreds of times within seconds. Also, a number of outliers were discovered related to the timing. It is however not possible to determine if the origin of the outliers is accidental (forgetting to use the tool), deliberate (cheating), or valid data (groups actually spending sufficiently more or less time on the project).

There is no overall environment that suits all studies. A trade-off should always be made. Either a restricted environment is chosen where the study can be closely monitored, however the environment can be perceived as unnatural and too restrictive which might impact the participant's results. On the other hand, an open environment resembles a real-world scenario and feels more natural to the participants making them more willing to participate, however, the study designers have no guaranteed insights on the progress of the study.

The ideal environment varies and mainly depends on the study itself (e.g. a study requiring 15 minutes of participation will be easier to host and monitor in a closed environment) and the required types of output (e.g. highly monitored throughout the process vs. only focused on final results).

5.5.2 Students vs. Practitioners

As LINDDUN is a fairly new and untested approach, the choice of using students as participants in the studies was straight-forward. Only when a methodology is more strongly tested, it will be rolled out in a more industrial environment. Nevertheless, working with students can bring some disadvantages.

Student attitude. Students have the tendency to work with minimal effort when permitted. Although this study was conducted during a graded course, it is still likely that certain students did no give the assignment their best effort. The reports have indeed indicated that some results were inadequate. It is hence important for successful studies to find participants that are highly motivated. It can be valuable to work exclusively with volunteers to ensure a motivated group of participants. Although the architecture study (described in Section 5.1) was part of a course, we still allowed the students to decide whether or not they would participate in the study (although the assignment itself was mandatory). This resulted in 14 teams who opted out of the study. Although this is a rather large groups of participants, it should not be considered problematic for a descriptive study. On the contrary, the dropped out participants were mostly students that did not have the proper motivation we expected for the study. Also, when required, results of participants that clearly lacked motivation should be excluded. We, for example, discarded the results of 11 groups of

participants in the descriptive architecture study as they clearly did not follow the instructions and often merely copied the example threats. When performing a descriptive study, it is important to be vigilant for these types of participants that lack motivation.

Educational trade-offs. Studies executed in an academic environment are often incorporated into a student course. This guarantees a large group of participants with a similar background. As the study has to be integrated into the syllabus, this can create drawbacks. In the first place, the participants are students who need to be taught. We therefore provided a very extensive tutorial which includes a fully worked out case where LINDDUN was applied to an example scenario. Given the student attitude described above, the provided misuse cases from the example were often reused by the participants, sometimes even without any alteration to adapt it to the application to be analyzed. It is hence hard to determine what the outcome would have been when this example scenario had not been provided. On the other hand, it is common practice in software engineering to reuse examples. It however will require further research to determine if analysts would benefit from a set of predefined general LINDDUN misuse cases.

Overall, integrating a study into an (academic) course can be very beneficial as it provides a relatively large set of participants with sufficient motivation, however some concessions might need to be made in order to make the study fit into the course's syllabus.

5.5.3 Practice Makes Perfect

Obviously, the more you execute a methodology the more familiar you become with its process and consequently with its advantages and disadvantages. Even before we carried out the studies, we gained better insights by performing LINDDUN ourselves on several examples and received feedback from colleagues. We therefore already made some minor changes to the methodology before the actual studies that have been described in this and previous chapter. Also during the studies, while observing the participants, several ideas for improvement grew. We therefore decided to implement these changes to enhance LINDDUN even further as is discussed in Chapter 6.

5.6 Conclusion

The results of the descriptive study of LINDDUN during the architectural design phase are comparable to those of LINDDUN at the requirements engineering phase. The correctness is, with its 70% precision rate, encouraging. Also the feedback on the ease of use of the methodology was overall positive. The completeness and productivity rate have however room for improvement. The case study with the privacy experts to explore LINDDUN's reliability also provides promising results. The coverage of LINDDUN is adequate as only two main gaps were identified. In fact, LINDDUN even uncovered some threats which the experts overlooked.

We can thus conclude that LINDDUN is a useful privacy requirements methodology that provides proper support to elicit privacy threats, both during the requirements engineering and architectural design phase. During the studies we did however identify some shortcomings which we tackle in the following chapter.

6

LIND(D)UN: an improved version of LINDDUN

This chapter contains the concluding leg of the three main contributions of this thesis. We used the results of the empirical evaluation, as described in Chapters 4 and 5 to improve the LINDDUN methodology. The chapter first zooms in on the shortcomings of LINDDUN as identified by the empirical studies. For each of these shortcomings, a change to the methodology is suggested. These evidence-based improvements are then extensively discussed in the following sections. The chapter concludes with the expected impact of the changes.

One of the proposed changes involves a repositioning of the security threats. As threats regarding the disclosure of information will no longer be considered part of the approach, we will refer to the improved methodology as LIND(D)UN.

Table 6.1: Main LINDDUN shortcomings and suggested improvements.

PROBLEM	SOURCE	IMPROVEMENT IN LIND(D)UN
1. Gaps in LINDDUN threat trees		
	Study with experts	Added inference and minimization to linkability tree (L_DS, see Section 6.6.2) and identifiability tree (I_DS, see Section 6.6.3)
2. Limited catalog documentation		
	Feedback from participants	Extending and restructuring the catalog (see Section 6.3)
3. Confusing interaction between LINDDUN and STRIDE		
	Observation + feedback + results (high FN rate)	Clear separation where LINDDUN requires STRIDE as prerequisite (see Section 6.4)
4. Low completeness rate		
	Results (FN)	Focus on impact of the mapping table (see Section 6.5)
5. Confusing threat trees		
Unawareness	Observation + results (FP and FN)	Shift responsibility from user to system (see Section 6.6.7)
Linkability and Identifiability	Observation	Adapted and restructured matching threat trees (see Sections 6.6.2 and 6.6.3)
Non-compliance	Observation + feedback	Renamed threat tree nodes + extended description (see Section 6.6.8)

6.1 LINDDUN Shortcomings

This section provides a short overview of the main issues we identified via the empirical studies. For each of the shortcomings we propose a change to the methodology (as summarized in Table 6.1).

A more extensive description of all improvements is provided in the remainder of this chapter.

6.1.1 Identified Gaps

Problem. As shown by the third study with the privacy experts (see Section 5.3), some threats were missing from LINDDUN. The main gap was related to data minimization (or the lack thereof) and inference. The

other missing threats were mainly related to societal harms and are therefore considered less critical for the LINDDUN methodology.

Solution. Although the need for data minimization was implied in the original trees, we decided to make it more concrete by mentioning it explicitly. Similarly, inference was not included in the original trees, as it is actually more a consequence of linkability while the threat trees describe conditions that can lead to linkability. Nevertheless, it is an important threat; either linkability leads to identifiability (i.e. linking data to an identity) or to inference (i.e. linking data based on certain properties to deduce relationships between them and generalize them). We therefore explicitly included inference in the improved LIND(D)UN trees. More information on these improvements can be found in Section 6.2.

6.1.2 Limited Catalog Description

Problem. In both studies, the participants indicated that they found the documentation of the threat trees rather limited. Indeed, the catalog could benefit from a more detailed description of each tree. Also, observations during the studies showed that the participants' overall knowledge of the privacy threat categories was rather limited.

Solution. We have addressed this issues in three ways by 1) extending the description of all threat trees, 2) structuring the description, and 3) adding information about each threat category in general.
First, we updated and extended the description of each leaf node. In addition, an overall description of the entire tree has also been added. Second, the descriptions have been organized according to tree's hierarchy to ease the navigation. Finally, to boost the privacy knowledge of the analyst, we also added a structured overview of each of the threat categories to the catalog. For each threat type (e.g. linkability), a general description is presented, followed by the threat type's consequences, and an overview of privacy concepts that can impact the threat type.

The details on the improved structure of the catalog can be found in Section 6.3.

6.1.3 Confusing Interaction between LINDDUN and STRIDE

Problem. The participants of both descriptive studies indicated, during the study and afterwards during feedback, that it was unclear how far into STRIDE they should have continued their analysis. Indeed, the information disclosure threats of LINDDUN are borrowed from STRIDE and make several references to other STRIDE threats, like spoofing. During the studies we therefore expected the participants to also elicit information disclosure and spoofing threats using STRIDE, which were often missed by the participants (as is reflected by the high number of false negatives for the spoofing category in Figure 5.3 of the empirical study chapter).

Solution. In practice, LINDDUN and STRIDE should both be executed to guarantee a secure and privacy-friendly system. There is however no compelling reason to intertwine both methods. We therefore propose to remove the threats related to the disclosure of information, which leads to a new acronym for the methodology. LIND(D)UN, in isolation, will only focus on the elicitation of privacy-specific threats and refer to STRIDE as a whole for information disclosure and other security-related threats.

The interaction between LINDDUN and STRIDE is described in more detail in Section 6.4.

6.1.4 Low Completeness Rate

Problem. The results of the studies showed a relatively low level of completeness of the threat analysis. When observing the participants during the studies, we noticed that they did not utilize the mapping table [1] to its full potential, which is a support to achieve completeness. When asked about the mapping table during the feedback session, several participants indicated that they did not consider this activity much helpful. This implies that the importance of the mapping table has not been stressed sufficiently in the tutorial.

Solution. Instead of using the table as a guideline, it should actually be used as checklist to determine completeness of the analysis. Each 'checkmark' in the mapping table should be covered by at least one threat described in a misuse

[1] The mapping table is created in the second step of the LINDDUN methodology and indicates for each DFD element to which threat categories it is susceptible. All 'checkmarks' in the table require further investigation by examining the correspond threat tree to determine the precise threats that apply.

case, or assumptions need to be annotated to discard the specific threat category completely. The importance of the mapping table should hence be highlighted in the methodology and in the training material.

More details on the use of the mapping table as checklist can be found in Section 6.5.

6.1.5 Confusing Threat Trees

During the studies, we received quite some questions regarding several threat trees as they were sometimes found difficult to comprehend. This inspired us to make some changes to improve the understanding and navigability of the trees.

Unawareness holds user responsible

Problem. During the execution of the descriptive studies, we received quite some questions from the participants for clarification of the unawareness threats as they were considered hard to grasp. This was also confirmed by the studies' results, as the unawareness threats result in several false negatives. The original tree holds the user responsible for providing too much or incorrect data. It is however not very privacy-friendly to make the user himself the sole responsible for privacy-aware decision making. In addition, these threats cannot be easily mitigated by a software system.

Solution. We therefore refactored the corresponding tree entirely and made the system the responsible party for raising awareness. This shift applies to both branches of the unawareness tree. First, the system should aid the user in deciding on what data he will share. Second, it is not the responsibility of the user to make sure the data that is collected by the system is correct (at least not from a privacy perspective). On the contrary, the system should provide a means for the user to verify the data that have been collected about himself.

A description of the detailed changes can be found in Section 6.6.7.

Linkability and Identifiability not aligned

Problem. Identifiability and linkability are concepts which the studies' participants are not familiar with. We noticed quite some confusion. It is indeed not easy to understand the subtle difference between both types of

threats. The original threat trees, however, do not represent the similarities between both threat types clearly enough, nor do they stress the differences.

Solution. Based on the experiences gained during the studies, we decided that it would make more sense to show the similarities in the threat trees. We have therefore restructured the trees of both threat categories to increase the resemblance. The main difference between the trees now lies in the fact that linkability only links together information, while identifiability is able to retrieve a link to an actual identity. During this restructuring, we also integrated some improvements to make the trees easier to navigate and understand.

Details on the improved threat trees can be found in Sections 6.6.2 and 6.6.3.

Abstractness of Non-compliance

Problem. The non-compliance tree describes rather high-level threats which often required some clarification during the studies. The tree contains a combination of policy- and consent-related threats, as well as threats that concern data management (i.e. updates and deletion of information), which will lead to diverse threats.

Solution. We decided to only slightly update the threat tree. We deliberately keep the compliance tree at a high level of abstraction, as we do not intend to provide a full-fledged methodology for compliance. Data protection legislation is a complex matter that requires the support of a legal expert. We only touch the subject to make the analysts aware of the need of legal analysis. However, we decided to narrow the scope of this tree by removing the threat related to data management. This was an abstract threat that actually encompassed a number of more specific threats that are already spread over other LIND(D)UN trees. We now focus specifically on policies and consents, and also include the concept of 'notice', which aims at providing the user with transparency regarding the collection and processing of his data.

A precise overview of changes in the non-compliance threat tree can be found in Section 6.6.8.

6.2 Improvement 1: Filling the Gaps

Two threats were identified by privacy experts (in the third case study as described in Section 5.3) that were missed by LINDDUN: inference and data minimization.

Actually, including these threats explicitly did not require a large change. *Data minimization* was implicitly already present in the 'linkability of data store' tree as weak data anonymization is mainly caused by the lack of data minimization. In fact, 'weak anonymization' is actually a too strong precondition for linkability, and 'insufficient minimization' covers the threat more precisely. Note that minimization can also be considered from a wider perspective, where one does not only focus on minimizing the information itself, but also tries to minimize central storage, risk, trust, etc. We however decided to limit ourselves to data minimization as we consider this the main privacy concern.

Inference is actually a technique that is applied to linkable data instead of a cause of linkability. In contrast to linkability threats that can lead to identifiability, inference is not limited to linking data that belongs to the same person. Inference will deduce relationships from certain related properties (e.g. people in the same location, people with the same disease, etc.) and try to generalize them. Inference can be used in a rather innocent fashion to determine the best way to organize groceries in a grocery store (e.g. people who buy hamburgers usually buy buns at the same time, hence they are stored close to each other), but can also have a more judgmental nature if it is used to discriminate a certain population (e.g. people living in a certain neighborhood have a higher chance of cancer, hence their health insurance fee is higher than the surrounding cities). Inference can thus lead to societal harm. Linkability and inference are however very related and the impact of inference should not be underestimated. Its importance was also reflected by the reliability study where all three privacy experts described an inference-related threat. We therefore explicitly mention 'inference' together with 'insufficient minimization' in the linkability of data store threat tree (as shown in Figure 6.6). We also added an additional explanation of inference to the catalog documentation [WSJ14b, LIN].

Although considered less critical for LINDDUN, also the societal threats that were initially missed are now (at least partially) included in the updated catalog. Discrimination, for example, is mentioned in the catalog description of inference (part of the linkability of data store tree). The need for data protection policies and default settings has also been made more explicit in the awareness tree.

6.3 Improvement 2: Extending the LINDDUN Catalog

Both descriptive studies have indicated that the LINDDUN privacy threat tree catalog provides much assistance in discovering threats, however, participants indicated that the descriptions that support the threat trees can use some improvement. We agree that the original descriptions were often rather limited. We addressed this issue by providing a detailed description of the threat category in general, as well as extending the information for each tree. In addition, we also structured the tree descriptions according to the tree's hierarchy to facilitate easy look-up of information per subtree and (leaf) node.

6.3.1 General Description

We created a general description for each LIND(D)UN threat type, as we noticed during the studies that the participants were not always familiar with the general concepts of a specific threat type. Figure 6.1 shows an example of such threat description. It is by no means the intention to provide a full overview of the threat type. We merely want to summarize the key concepts and, if available, provide a link to more extensive related work.

We organized the summary of each threat category according to three main items:

In general. This description summarizes the main concepts of the specific threat type. Also links to reference material are added for more inspired reading.

Consequences. We also listed possible consequences of such threats, as it will help to understand the severity of the threat. Sometimes the threat itself does not seem to have a big impact, however, its consequences can be quite privacy invasive (e.g. linkability).

Impacted by. Threats can also be impacted by other privacy concepts, both in a positive and negative way. For example, data minimization will limit linkability threats, while identifiability will increase them.

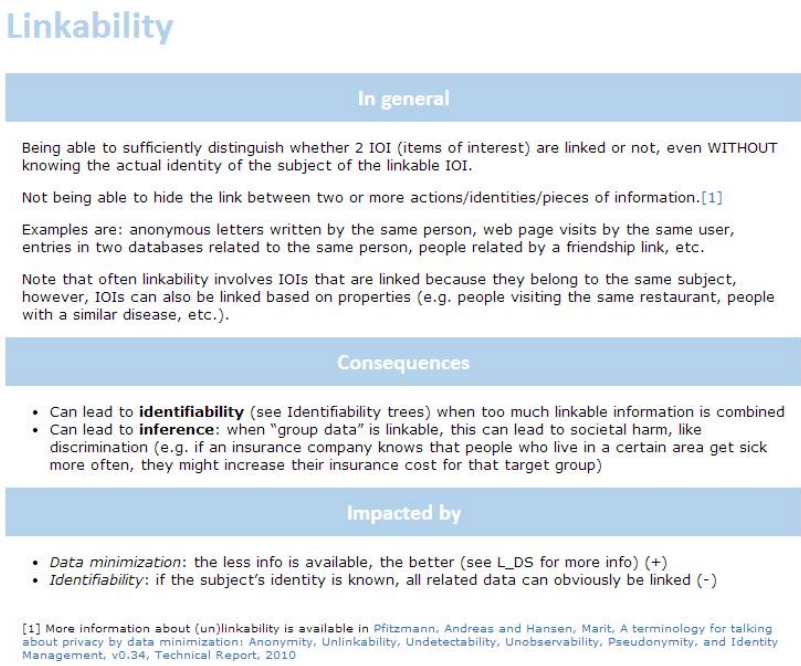


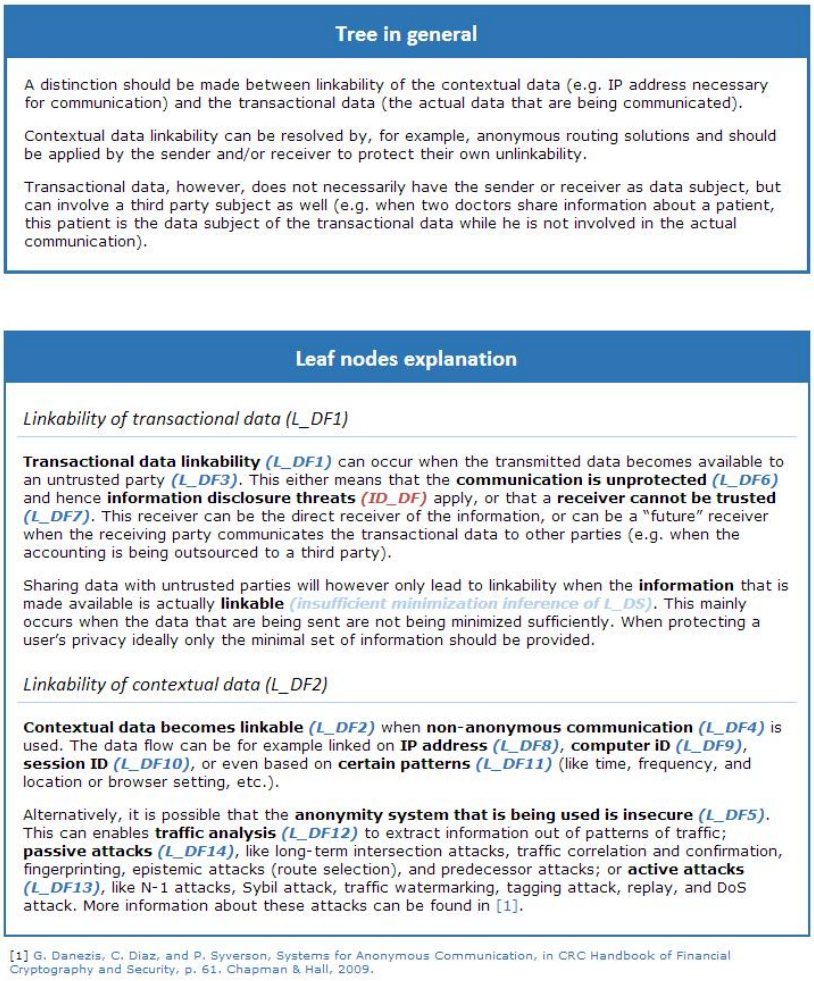
Figure 6.1: Example of improved general threat description: Linkability

6.3.2 Tree Description

For each individual tree, a description is provided, both in general, as well as a detailed overview of each node. The description of the leaf nodes is organized according to the tree's structure. Figure 6.2 shows the catalog description for the linkability of data flow tree.

Tree in general. A short general description is provided that contains a summary of the tree's threats.

Leaf nodes explanation. Each node is described to provide additional information. This specification is structured according to the subtrees to easy navigation within the description. Leaf node labels are highlighted to allow pinpointing of a particular node at glance.



Tree in general

A distinction should be made between linkability of the contextual data (e.g. IP address necessary for communication) and the transactional data (the actual data that are being communicated).

Contextual data linkability can be resolved by, for example, anonymous routing solutions and should be applied by the sender and/or receiver to protect their own unlinkability.

Transactional data, however, does not necessarily have the sender or receiver as data subject, but can involve a third party subject as well (e.g. when two doctors share information about a patient, this patient is the data subject of the transactional data while he is not involved in the actual communication).

Leaf nodes explanation

Linkability of transactional data (L_DF1)

Transactional data linkability ***(L_DF1)*** can occur when the transmitted data becomes available to an untrusted party ***(L_DF3)***. This either means that the **communication is unprotected** ***(L_DF6)*** and hence **information disclosure threats** ***(ID_DF)*** apply, or that a **receiver cannot be trusted** ***(L_DF7)***. This receiver can be the direct receiver of the information, or can be a "future" receiver when the receiving party communicates the transactional data to other parties (e.g. when the accounting is being outsourced to a third party).

Sharing data with untrusted parties will however only lead to linkability when the **information** that is made available is actually **linkable** *(insufficient minimization inference of L_DS)*. This mainly occurs when the data that are being sent are not being minimized sufficiently. When protecting a user's privacy ideally only the minimal set of information should be provided.

Linkability of contextual data (L_DF2)

Contextual data becomes linkable ***(L_DF2)*** when **non-anonymous communication** ***(L_DF4)*** is used. The data flow can be for example linked on **IP address** ***(L_DF8)***, **computer iD** ***(L_DF9)***, **session ID** ***(L_DF10)***, or even based on **certain patterns** ***(L_DF11)*** (like time, frequency, and location or browser setting, etc.).

Alternatively, it is possible that the **anonymity system that is being used is insecure** ***(L_DF5)***. This can enables **traffic analysis** ***(L_DF12)*** to extract information out of patterns of traffic; **passive attacks** ***(L_DF14)***, like long-term intersection attacks, traffic correlation and confirmation, fingerprinting, epistemic attacks (route selection), and predecessor attacks; or **active attacks** ***(L_DF13)***, like N-1 attacks, Sybil attack, traffic watermarking, tagging attack, replay, and DoS attack. More information about these attacks can be found in [1].

[1] G. Danezis, C. Diaz, and P. Syverson, Systems for Anonymous Communication, in CRC Handbook of Financial Cryptography and Security, p. 61. Chapman & Hall, 2009.

Figure 6.2: Example of improved tree description: Identifiability of Data Store

6.4 Improvement 3: Reducing Interaction between LINDDUN and STRIDE

During the studies, there was some confusion among the participants as to what extend the STRIDE threats should be included in the analysis. LINDDUN's disclosure of information threat category reuses STRIDE's information disclosure threats, which refer to other security-related threats. LINDDUN strongly

leverages on STRIDE (with information disclosure threats in particular) to ensure that the privacy-friendly model that results from LINDDUN also takes the data subjects' confidentiality and integrity into account.

In our updated LIND(D)UN methodology, we would however advice against an only partial security analysis using STRIDE based on the trees that are directly referred to by LINDDUN, as security is an asset too important for an incomplete analysis. We thus suggest to have both a privacy analysis using LIND(D)UN, which temporarily discards the information disclosure threats, and a full security analysis, either in parallel or consecutively. We would encourage the use of STRIDE as security threat modeling technique, as it is easily integrable with LIND(D)UN. The DFD that is created during the first step of the one methodology can, in fact, be entirely reused by the other. However, other security analysis techniques can also be used as alternative.

When assured that LIND(D)UN is being used in parallel with STRIDE (or other security analysis techniques), we can concentrate LIND(D)UN to its core by only examining threat categories that are explicitly related to privacy, and hence discard the information disclosure threats.

Because one should however always keep in mind that privacy is also threatened by security threats, we still keep a reference to disclosure of information threats in the LIND(D)UN catalog, with the caveat that the analyst should not limit himself to these threat trees but preferably perform a full security analysis.

6.5 Improvement 4: Increasing Completeness by using the Mapping Table as Checklist

The studies showed that the completeness rate is relatively low and rather obvious threats have been overlooked. We noticed that the participants did not fully utilize the mapping table that was created during the second step of the methodology. They used it to determine which threats they did not have to consider, but they neglected to use it as a checklist to verify that indeed all applicable threats have been investigated and documented. In the questionnaire, the participants of the study also indicated that they did not consider the mapping step as particularly useful (only an average of 3 on a scale from 1 to 5), while it actually is. The use of the mapping table should not be limited to the second step of the methodology, but should be used throughout the elicitation phase as well. We therefore suggest to emphasize its importance throughout the different steps more thoroughly in the methodology and in the training material, as illustrated in Figure 6.3. We believe that using the

mapping table as checklist will improve the completeness rate of LINDDUN (as during the descriptive studies, the recall was considerably low). For each 'checkmark' in the mapping table, all leaf nodes of the corresponding threat tree should be considered. Either they are all covered and documented as threat(s) or appropriate assumptions were made explicit to eliminate irrelevant nodes or (sub)trees.

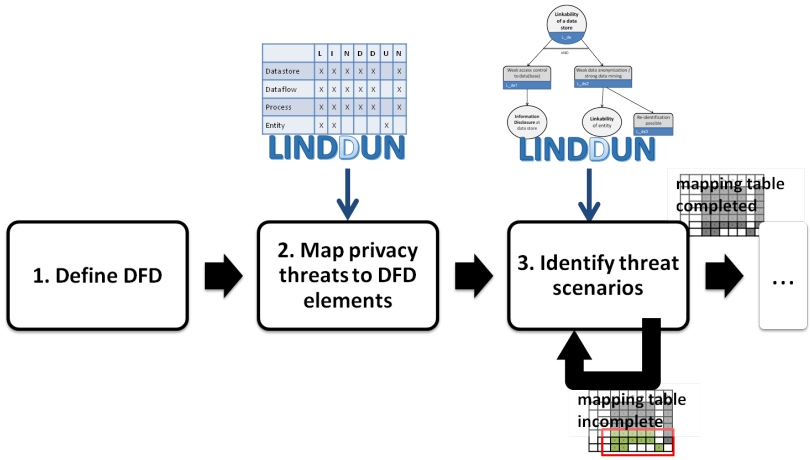


Figure 6.3: Problem-oriented steps of LIND(D)UN methodology with a focus on the mapping table as checklist

When applying STRIDE, a similar technique is used. When you have not covered all 'checkmarks' in the mapping table, you know you are not done yet. Although it should be noted that having all the checkmarks covered, does not imply completeness [Sho14].

6.6 Improvement 5: Enhancing the LINDDUN Threat Trees

Based on both the results and the participants' feedback of the first two studies, we decided to restructure and extend some of the threat trees to eliminate the observed confusion over certain trees. Some of the extensions were also based on the third study with the privacy experts, and aim to increase reliability of the LINDDUN methodology. The entire updated catalog of threat trees can be found in [WSJ14b].

6.6.1 Subtrees

A general change concerns the reference to subtrees. As some rather large branches were repeated in multiple trees, we decided to centralize them by adding them to the tree that is most related and have other trees reference the subtree rather than copy it entirely. This was done to avoid confusion (e.g. why discuss dataflow threats in an entity tree) and to better structure the trees.

6.6.2 Linkability

As linkability is closely related to identifiability, we decided to also show this resemblance in the threat trees. As this similar structure makes trees of both categories easily comparable, the subtle differences between linkability and identifiability become clearer as well. For both threat categories, we made quite some changes to the trees. We restructured some (sub)trees to ease navigation and explicitly added some more threats (mainly to the original leaf nodes) to enhance the understanding of the threats.

Linkability of Entity. As shown in Figure 6.4, we added a more clear distinction between the two subtrees by having linkability based on a linkable login, and linkability based on the metadata of the entity's communication. The former subtree is an extension of the original threat that mentions an insecure flow or store. The original threat was actual quite generic. Based on the changes made to the identifiability of entity tree (which has very similar threats), it was decided to clarify the threat as 'a linkable login is being used which is communicated in an untrusted way.' Concerning linkable logins we made a distinction between a fixed login that is being reused (e.g. username-password) and certificates (which are much more privacy-friendly) that are too specific and therefore can become linkable. A linkable login will only cause a privacy risk when it is being communication in an untrusted fashion. This can occur when the login is sent over an untrusted channel (information disclosure of dataflow threats), stored in an linkable way (linkability of datastore threats), or sent to an untrustworthy receiver.
The latter refers to a subtree of data flow linkability and was originally part of both trees. To avoid duplication (and confusion), we however only kept it as part of the dataflow tree (because the threats specifically focus on communication) and refer to it from the entity tree.

Linkability of Data Flow. Concerning the dataflow, we made an explicit distinction between linking the actual data that are being transferred and

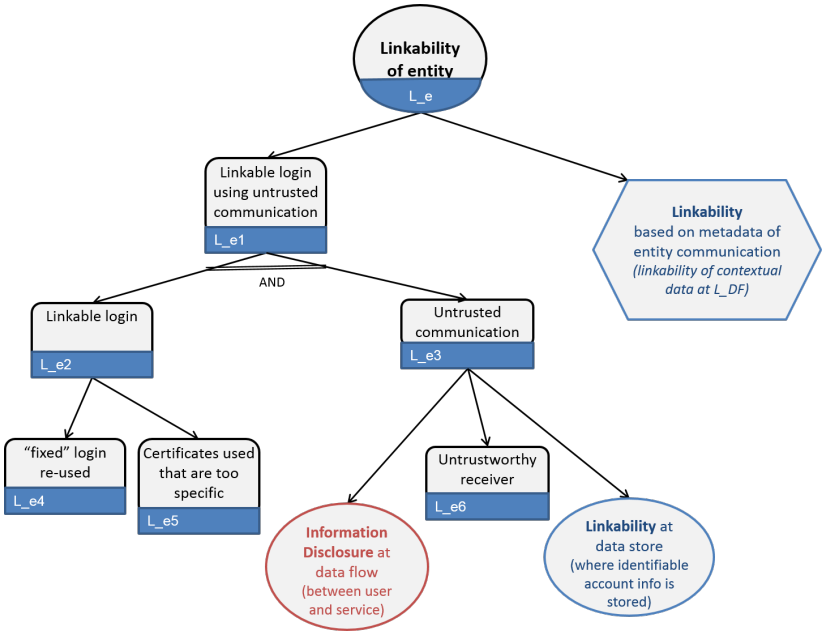


Figure 6.4: Improved Linkability of Entity tree

linking the communication's metadata (see Figure 6.5). The former has been extended, as the original tree only mentions an unprotected dataflow. In fact, an unprotected dataflow will only lead to linkability issues when the data that are being transferred are actually linkable themselves. Also, even when the flow itself is protected but the receiver is untrustworthy (and the data are linkable), the data subject's linkability is also at stake.

The latter has remained unchanged except for the parent node which now clearly states it concerns linkability of contextual data (or metadata).

Linkability of Data store. We only moderately altered the tree as shown in Figure 6.6. As mentioned in Section 6.2, the "weak data anonymization" node was renamed to "insufficient minimization/inference" because weak anonymization was actually a too strong precondition for linkability. This immediately resolves the gaps that were identified.

The linkability of entity reference has been replaced with a regular node concerning linking data to another database, as this is more accurate.

Finally, the re-identification node has been renamed and extended, as for

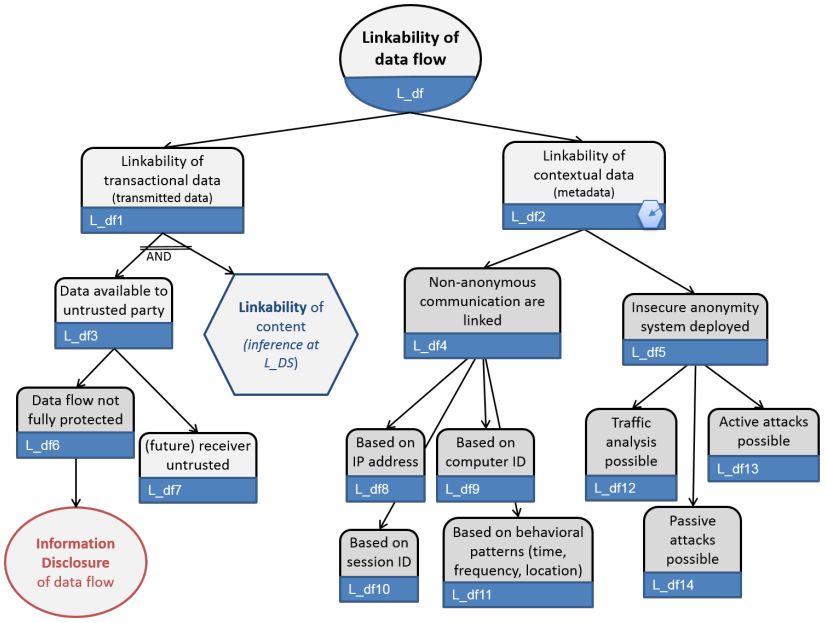


Figure 6.5: Improved Linkability of Dataflow tree

linkability threats, actual re-identification is not required. Merely the combination of excessive data can already lead to linkability based on pseudo-identifiers belonging to the same person (which can lead to identifiability) or based on certain properties to deduce relationships between them and generalize them by using inference.

Linkability of Process. As this is a very rare threat, we did not receive any feedback on it, and decided to leave it unchanged.

6.6.3 Identifiability

As mentioned before, we tried to enhance the structural similarities between linkability and identifiability. The trees for both threat types will thus be broken up in the same way.

Identifiability of Entity. First of all, a link to a subtree was added regarding the identifiability based on metadata (see Figure 6.7). Although these threats

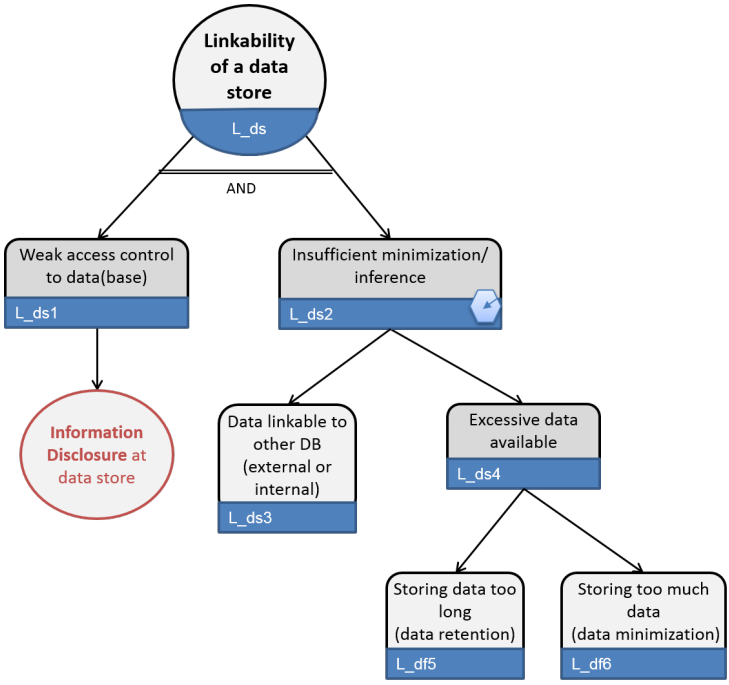


Figure 6.6: Improved Linkability of Data store tree

were available in the original linkability tree, it was missing for identifiability.

Second, we restructured the tree. In the original version, each type of login also referred to possible problems with several nodes referring to the same child nodes, which made it rather difficult to navigate. After evaluation, it turned out that the same problems related to untrusted communication actually apply to all the login types (although some of them where missing in the original tree). We therefore make a distinction in the improved tree between the type of login used and their specific issues (which is one subtree) and the problems concerning untrusted communication that are related to all of them (which is another subtree). The untrusted communication subtree mentions 4 distinct threats: untrustworthy receiver of the login, weak storage of the login by the client, information disclosure of the data flows that transfer the login (this actually refers to a security threat tree), and identifiability of the data store where the identifiable account information is stored.

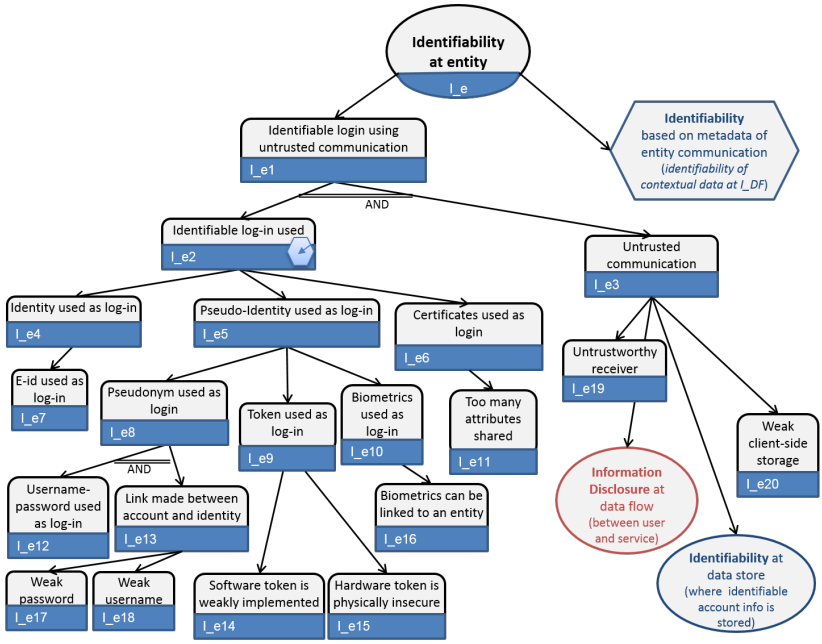


Figure 6.7: Improved Identifiability of Entity tree

Identifiability of Data Flow. The identifiability of data flow tree is very similar to the linkability of data flow tree. The main difference is, as the threat categories themselves suggest, the distinction between merely linking data (for linkability) and actually identifying them (for identifiability). The approaches of both threat trees, and hence its structure, are however the same. We therefore applied the same changes to the identifiability of data flow tree (see Figure 6.8), as to the linkability tree. The right branch is kept unchanged (only to top node has been renamed to explicitly mention contextual data). The left branch has been extended because information disclosure will only lead to identifiability if the content of the message contains identifiable information.

Identifiability of Data Store. As shown in Figure 6.9, the structure of the original tree was mainly kept, only some leaf nodes were extended to clarify threats. Data in the data store do not necessarily become identifiability when the entity is identifiable (as shown in the original tree). Only when an identifiable login is used and this login is linkable to the data in the data store that are being examined, the data store's data become identifiable. We also specified how re-identification can occur: when data are insufficiently minimized (which

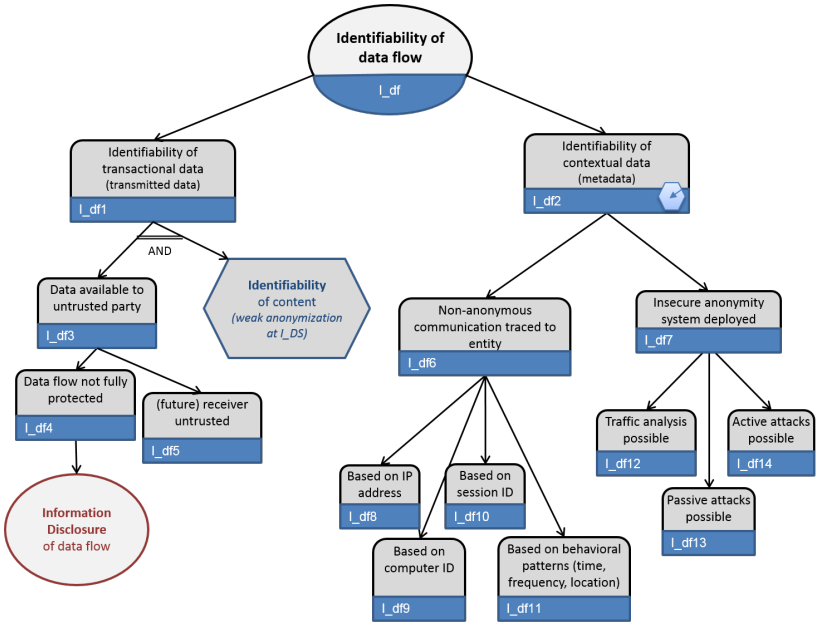


Figure 6.8: Improved Identifiability of Dataflow tree

this node refers to a linkability subtree) and when these linkable data become identifiable (for example because the combination of several pseudo-identifiers leads to an actual identity). The structure of this tree again closely resembles the corresponding linkability tree as they are related.

Identifiability of Process. As this is also a very rare threat, we did not receive any feedback on it, and decided to leave the tree unchanged.

6.6.4 Non-repudiation

Non-repudiation is a rare privacy threat category which only applies in niche applications such as e-voting and whistleblower systems. The software systems that were analyzed during our descriptive studies were therefore not susceptible to non-repudiation threats. Since no feedback was received for this category, we kept the trees in their original state.

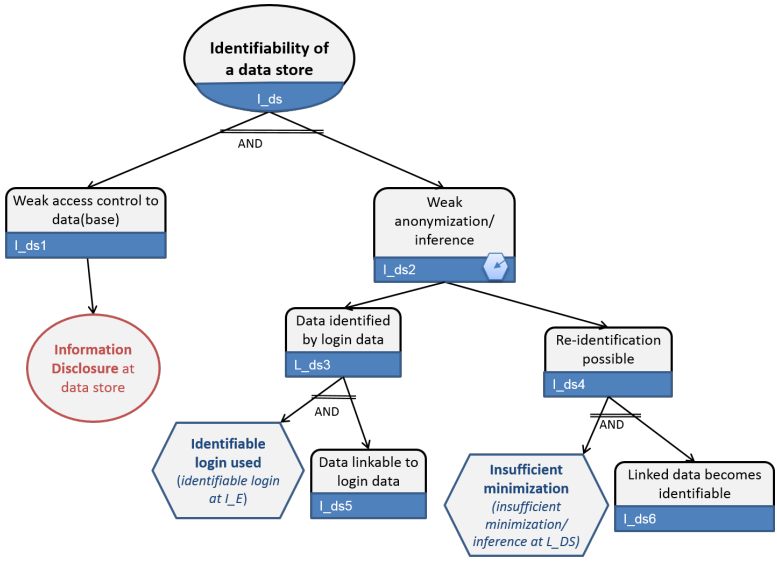


Figure 6.9: Improved Identifiability of Data store tree

6.6.5 Detectability

Similar to non-repudiation, detectability is also a less common threat type. The software application in the second study (i.e. ReMeS) was nevertheless susceptible to a detectability threat, but unfortunately the threat was not identified by the participants. As we thus did not receive feedback on the trees themselves, we did not make any changes to them. We did extend the explanation of the tree nodes (as discussed in Section 6.3) to increase the understanding of the threat category.

6.6.6 Disclosure of information

Because the disclosure of information threats are actually borrowed from Microsoft's STRIDE, which is a well established technique, we decided not to make any changes.

In fact, we actually advice to perform a full security analysis in addition to LIND(D)UN (as explained in Section 6.4). This would eliminate all information disclosure threats from the LIND(D)UN methodology, as they are already covered by the security analysis.

6.6.7 Unwareness

The unawareness threat tree has been changed quite profoundly. During the descriptive studies, we received several questions regarding this tree as it was perceived as confusing. Also the studies' results showed a large set of false negatives in the unawareness category. Indeed, when re-evaluating this threat tree, we agreed that an update of this tree was required.

First of all, the name "content unawareness" was found quite confusing. We received this comment on multiple occasions, even outside the scope of our studies. We therefore decided to drop "content" from the title, and simply call it unawareness, as shown in Figure 6.10.

Second, we shifted the focus of the unawareness threats. While the original tree was very user-centric and put the main responsibilities on the user himself, we moved to a more system-centric awareness as we want to have the service providers implement as much support as possible to improve the user's privacy awareness. Currently, it is unfortunately still often common practice to hold the user responsible for his privacy; e.g. "informed" consents where the user has to inform himself, (hidden) opt-out consents which the user should figure out himself, or the general belief that the user should just know to not overshare information. The system itself should aid the user in privacy decision making by providing more transparency or even nudging the user into a privacy-friendly decision. As LIND(D)UN focuses its analysis on the system itself, this perspective of (un)awareness is more applicable. Our privacy analysis can not impact the entire society, and will therefore focus on the provisions the system itself can take to guide and educate the user concerning his data sharing and nudge the user into a more privacy-aware use of the system.

We therefore altered the entire subtree related to "wrong decisions made based on incorrect data." It is not the responsibility of the user to keep his information up-to-date (especially not from a privacy perspective). Data accuracy is however important from a different angle. A data subject should be able to see what data has been collected, stored, and processed about himself.

Finally, we have extended the "providing too much personal data" threat. There is a subtle difference between this threat and the "insufficient data minimization" threat in the linkability tree. The latter aims at minimizing the data that has already been received, while the former tries to prevent the sharing of excessive information at the source. This is hard to prevent from a technical perspective as the user himself decides what information he will precisely share. Technology can only play a supportive role by, for example, providing feedback concerning the (additional) data the user wants to add, having privacy-friendly default settings, and providing user-friendly privacy tools. The main challenge will be

to educate people about their (online) sharing behavior, however this is clearly out of scope of our methodology.

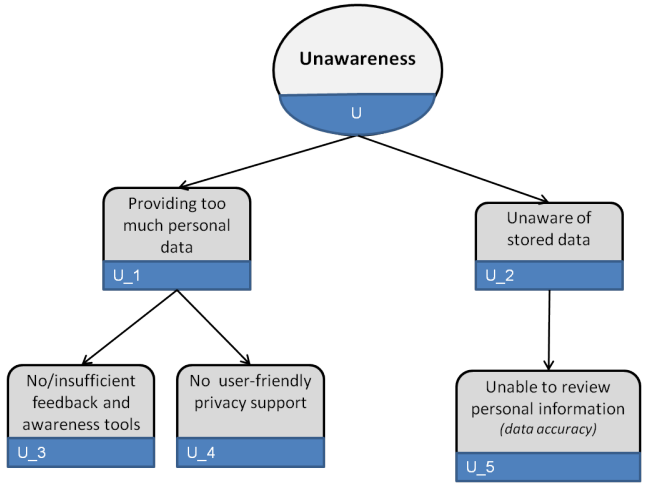


Figure 6.10: Improved Unawareness tree

To summarize, the awareness property focuses on the user's consciousness regarding his own data. The user needs to be aware of the consequences of sharing information. This property does not imply that it is the sole responsibility of the user to ensure his own privacy. On the contrary, we would encourage the service providers to facilitate tools to increase user awareness.

6.6.8 Non-compliance

Non-compliance raised quite some questions from the participants during the studies, as the threat tree nodes refer to rather high-level concerns. We did however only slightly modify the tree, as we intentionally only focus on compliance from a more abstract perspective. Legal compliance is a complicated domain. We do not aim at providing solutions for legal compliance, we only touch the subject as badly implemented data protection legislation and privacy policies can impact the data subject's privacy. To ensure full legal compliance, we suggest the assistance of a legal expert.

We renamed "insufficient consent management" into "insufficient policy management" to make it more general (see Figure 6.11).

The leaf node "insufficient data expire / update" was removed. This threat is actually closely related to the newly introduced data minimization and data

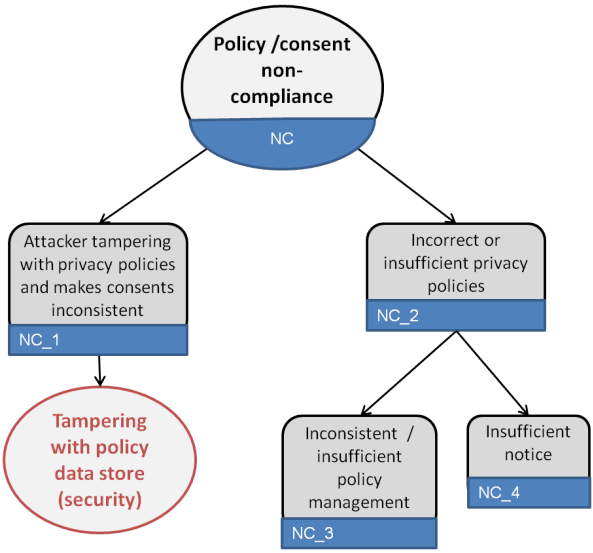


Figure 6.11: Improved Identifiability of Data store tree

retention threats in the linkability tree. We decided to keep the focus of the compliance tree strictly on privacy policies. We did include the caveat in the catalog description that, in order to obtain full legal compliance, these additional regulatory threats should be analyzed, preferably by a legal expert.

Finally, one leaf node, extracted from data protection legislation, was added as it is closely related to privacy policies: "insufficient notice." This threat concerns systems that do not have transparent policies and hence do not sufficiently inform the users regarding data collecting and processing.

We also extended the provided documentation to clarify the concepts in more detail.

6.7 Moving Forward with LINDDUN to the Solution Space

Only the first three core steps of LINDDUN have been empirically validated. This part of the methodology is more problem-oriented as it focuses on the elicitation of threats. The final three steps of LINDDUN are solution-oriented and even though we did not get empirical observations on possible improvements

for these steps, we believe we can suggest some changes based on the feedback we received from researchers who have applied the methodology independently [Bec12, HBQ12].

The original methodology moves from elicited threats to privacy requirements which are then used to select privacy enhancing solutions. In LIND(D)UN, we propose a more light-weight approach to move directly from the elicited threats to a tangible solution, without the burden of eliciting requirements. We suggest to first determine the suitable mitigation strategy for each threat. When an appropriate strategy has been decided, the selection of the relevant privacy enhancing technologies becomes more straightforward. In Figure 6.12, we show the updated structure of the LIND(D)UN methodology. After all threats are identified (step 3), they are prioritized according to their impact (step 4). In order of importance, each threat is mitigated by the appropriate mitigation strategies (step 5). The classification of privacy enhancing technologies according to the mitigation strategies to which they adhere enables a more focused selection of suitable privacy enhancing solutions (step 6).

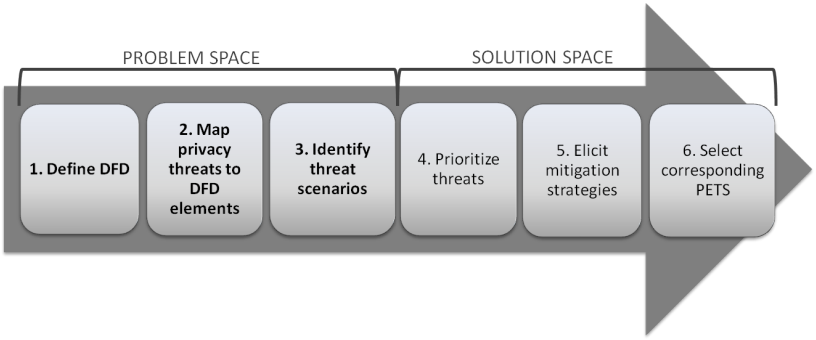


Figure 6.12: LIND(D)UN methodology: an improved version of LINDDUN

6.7.1 From Threats to Mitigation Strategies

The goal of threat modeling is to identify flaws in the system in order to resolve them. Addressing these flaws will require a number of fundamental design decisions. These decisions are often referred to as strategies (or tactics)[BCK05]. A strategy describes the means to achieve a certain objective, or, in case of LIND(D)UN, a means to resolve privacy threats. We will therefore refer to them as '*mitigation strategies*.' They capture a high-level view of common techniques used in practice to prevent privacy threats. As these strategies can be used to classify privacy solutions, they are a more suitable step in the conversion

of privacy threats to appropriate privacy enhancing solutions. We therefore propose to adjust the fifth step of the original LINDDUN methodology "eliciting requirements" to "eliciting mitigation strategies."

In essence, we leverage on a taxonomy of solution strategies that was created in our own previous work [WSDDJ09]. This hierarchical overview provides a structured classification of common mitigation decisions. In addition, we provide a means to select the appropriate strategies by mapping them to the LIND(D)UN threat trees they intend to mitigate.

A Taxonomy of Mitigation Strategies. As stated in [WSDDJ09], obtaining privacy means controlling the consequences of exposing the (possibly indirect) association of individuals to some information/transaction in a given context. Accordingly, in order to obtain privacy, it is important to focus on two major strategies. First, associations between users and their transactions and personal information need to be controlled in order to ensure that the user shares as little information as necessary with the system. This is the *proactive* approach. Second, the damage must be limited by controlling the associations after disclosure. To achieve this, the exposure of these associations needs to be restricted to the minimum. This is the *reactive* approach.

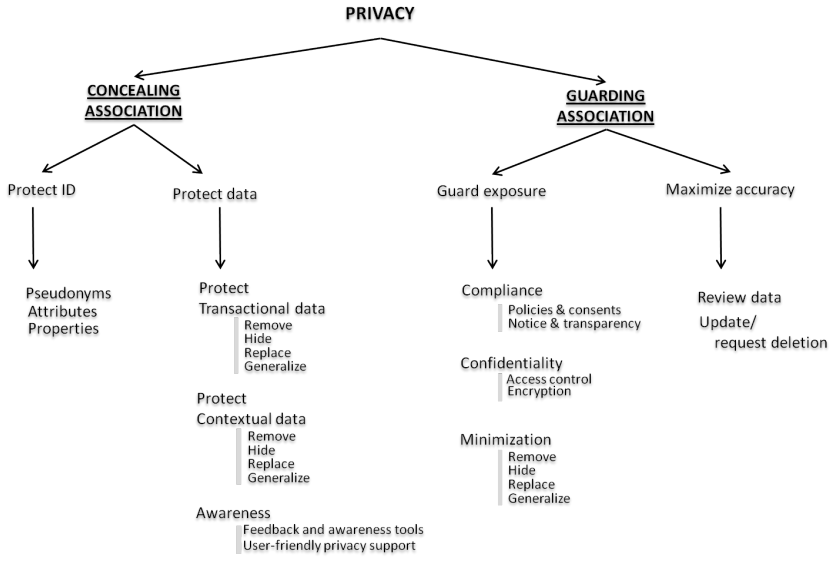


Figure 6.13: Taxonomy of LINDDUN Mitigation Strategies linked to LINDDUN threat categories

Concealing the association can be divided into two sub-strategies: (1) protect the identity of the user during authentication and (2) protect the data that will be communicated to (or throughout) the system.
First, as shown on the left-hand side of Figure 6.13, the identity of the user can be protected by means of pseudonyms (i.e. an alias instead of the real identity), or an attribute (such as a certificate) can be used. Even more anonymity can be achieved by using properties (such as anonymous credentials or other zero-knowledge proofs) as authentication mechanism.
Second, the data that are being communicated have to be protected. We make a distinction between strategies that concern core data protection and those that aim at raising awareness regarding the sharing of information. Data protection is in fact even subdivided into strategies related to transactional data (i.e. the actual data that are being transmitted) and strategies to protect contextual data (i.e. the metadata related to the communication) as both will require different kinds of solutions. Encrypting the information that is being transmitted, for example, will only protect the transactional data, while the contextual data can still reveal the sender based on information required for the communication (e.g. IP address, browser settings, etc.). Strategies to protect data include removing the (sensitive) information, hiding the data, replacing (part of) the information or generalizing it. Note that the strategies related to both types of data are however the same as they only represent high-level approaches used to mitigate the associated threats. The corresponding solutions will however strongly differ (e.g. removing sensitive information before sending the transactional data vs. removing contextual data, such as sender information, by applying onion routing techniques). A final strategy to conceal an association is to make the users more aware of the consequences of sharing. Feedback and awareness tools have been emerging to assist the user, and also user-friendly privacy support can be beneficial to aid the user at managing his privacy settings.

Guarding the association after the data has been shared can be divided into two sub-strategies: (1) guard exposure and (2) maximize accuracy.
Guarding the exposure is the most obvious strategy. All sub-strategies correspond to ways to restrict access and reduce disclosure to the collected data. The strategies are divided into 3 main categories: means to obtain data protection compliance which are related to consents and policies, and notice and transparency; means to ensure confidentiality which correspond to security measures such as access control and data encryption; and means to minimize the collected data. The strategies related to minimization are in fact the same as those related to the protection of transactional data. Some of the solutions might even interlace (e.g. hiding data by encrypting it) however each category will have its dedicated solutions.
Maximizing the accuracy is the final strategy to guard the association. It empowers the subject as it allows inspection and correction of his information.

Table 6.2: Mapping of Mitigation Strategies to Threat Trees

MITIGATION STRATEGY	LINDDUN THREAT TREE
Protect ID	L_e, I_e
Protect data	
Transactional data	L_df1, I_df1
Contextual data	L_df2, I_df2, D_df, NR_df
Awareness	U_1
Guard exposure	
Compliance	NC
Confidentiality	ID_ds, NR_ds, *_p
Minimization	L_ds, I_ds, D_ds
Maximize accuracy	
Review data	U_2
Update/ request deletion	NR_ds3

We make a distinction between two more detailed strategies. The first strategy allows a subject easy access to the collected data about himself in order to review the data. This strategy relates to user awareness. The second strategy extends this access right as it allows the subject to request updates or even deletion of his information. This strategy can be applied to obtain plausible deniability. An example of such deletion is the 'right to be forgotten' which was recently implemented by Google and allowed subjects to request removal of personal information from Google's search index if the links are inadequate, irrelevant or no longer relevant, or excessive in relation to the purposes for which they were processed.

Note that the taxonomy does not aim at providing a complete overview of strategies. It merely shows a common set of mitigation approaches that can be used to group similar solutions. Also, the level of detail of the strategies can be extended when desired.

Selection of Mitigation Strategies. As shown in Table 6.2, the taxonomy can be used to determine the appropriate strategy to mitigate a particular threat. For each strategy, the corresponding threat trees are provided.

First, a distinction can be made based on the type of DFD element that corresponds to a specific threat. Threats related to entities and data flows correspond to the concealment of data branch in the taxonomy (left branch of Figure 6.13). In particular, strategies that aim at *protecting the identity* will mitigate entity-related threats (linkability and identifiability of entity), while mitigation strategies for *data protection* before and during communication will

aim at resolving data flow threats. The linkability and identifiability trees each have a branch that corresponds to transactional data and one related to contextual data protection. Detectability and non-repudiation threats of the data flow are entirely focused on contextual data. The final strategy related to protection of communicated data is awareness, which evidently corresponds to the unawareness threat tree. Note that only the left branch of the unawareness tree maps to this strategy, as this tree is actually divided into threats that require proactive mitigation and those that require reactive solutions.

Second, mitigation strategies that *guard the exposure* of associations correspond to threats related to data that have already been collected and stored. These clearly map to data store related threats, which can be divided in two categories: threats that require confidentiality and threats that require minimization. The main part of data store related privacy threats are mitigated by strategies that have data minimization as common denominator. This is in fact a broad category of mitigation strategies that limit the collected data by, for example, removing redundant data, or generalizing personal information. Data store threats related to non-repudiation and information disclosure will require confidentiality strategies such as access control to the data store and encryption of the stored data[2]. Threats to a process also require strategies to guard exposure. They are all mitigated by confidentiality, as information disclosure is their main threat.

Finally, strategies to *maximize the accuracy* are very specific and correspond to two subtrees. Reviewal of data corresponds to the right branch of the unawareness tree as this requires awareness of the data that have been collected about the subject. As mentioned before, unawareness threats can be found in both branches of the taxonomy of mitigation strategies as a user should be aware of the consequences before he shares information, as well as of what data are in fact collected about him to verify the accuracy. The strategy that maximizes the accuracy by empowering the subject to update and delete collected data (either directly, or indirectly by requesting deletion) is linked to the subtree of non-repudiation that requires deniability by editing the database.

6.7.2 From Strategies to Privacy-enhancing Solutions

A threat can be mitigated by several types of solutions, each with their own benefits and drawbacks. The selection of a fitting solution is thus not straightforward. We therefore leverage on mitigation strategies to enable an easier and more focused selection of the appropriate privacy enhancing

[2]Note that the threats that are mitigated by minimization strategies are indirectly also linked to confidentiality strategies, as hiding data will be often achieved by encryption and access control techniques. Minimization is however much broader than confidentiality

technologies. By first determining a proper strategy to mitigate the threat, we narrow the scope of relevant solutions.

Originally, LINDDUN provided a table of solutions that indicates to which privacy properties each solution contributes. As shown in Table 6.3, we have restructured the solutions and mapped them to the mitigation strategy they adhere most to. The table represents the hierarchical taxonomy of mitigation strategies and links each leaf node of the taxonomy tree to the relevant solutions. Note that, although some solutions correspond to multiple strategies, we only assigned them to their key strategy (e.g. onion routing does not only remove contextual data, but also hides transactional data). In addition, some strategies refer to solutions of related strategies to avoid duplication (e.g. generalization of transactional data refers to encryption solution in the guard exposure branch). We have currently limited ourselves to a restructuring of the original solutions table. An update with state-of-the-art privacy enhancing technologies is part of our future work.

Once the mitigation strategy has been decided, a limited set with designated privacy enhancing techniques can be extracted from the solution table. The more detailed the proposed mitigation strategy is, the more focused the selection of appropriate solutions will be. The table thus facilitates an easy selection of proper privacy enhancing technologies to mitigate a specific threat.

6.8 Expected Impact of the Suggested Improvements

In this section we present our expectations regarding the impact that the suggested improvements might have on the performance indicators stated in Section 4.1.5: correctness, completeness, productivity, ease of use, and reliability. These expectations form the basis for the research hypotheses to be examined in future work in order to validate the proposed improvements.

Improved correctness. By modifying and extending some of the trees and renaming certain nodes, the trees themselves and the reciprocal relationships between the nodes will become more clear. As this will increase the understanding of the individual threats, we believe this will improve the overall correctness of the elicited threats.
Similarly, the extension of the LIND(D)UN catalog description will also have a positive effect on the correctness. Although the main purpose of the extended

Table 6.3: Mapping of Mitigation Strategies to Privacy Enhancing Solutions

Mitigation Strategy					Privacy Enhancing Techniques (PETs)
Concealing association	Protect ID	Pseudonyms			Privacy enhancing identity management system [HBC+04], User-controlled identity management system [CPHH02]
		Attributes			Privacy preserving biometrics [STP09], Private authentication [AF04, ABB+04]
		Properties			Anonymous credentials (single show [BC93], multishow [CL04])
	Protect data	Transactional data	Remove		(see *awareness* to minimize information sharing)
			Hide	Data-flow specific	Multi-party computation (Secure function evaluation) [Yao82, NN01], Anonymous buyer-seller watermarking protocol [DBPP09]
				General	see guard exposure - Confidentiality - encryption
			Replace		/
			Generalize		see *guard exposure - minimization - generalize*
		Contextual data	Remove		Mix-networks (1981) [Cha81], , ISDN-mixes [PPW91], Onion Routing (1996) [GRS96], Tor (2004) [DMS04])
			Hide	General	Crowds (1998) [RR98], Low-latency communication (Freedom Network (1999-2001) [BGS01], Java Anon Proxy (JAP) (2000) [BFK00]
				Undetectability	Steganography [AP98] , Covert communication [MNCM03], Spread spectrum [KM01]
				Non-repudiation	Deniable authentication [Nao02], Off-the-record messaging [BGB04]
			Replace		Single proxy (90s) (Penet pseudonymous remailer (1993-1996), Anonymizer, SafeWeb), anonymous Remailer (Cipherpunk Type 0, Type 1 [Bac], Mixmaster Type 2 (1994) [Mixa], Mixminion Type 3 (2003) [Mixb])
			Generalize	Undetectability	dummy traffic, DC-networks (1985) [Cha85, Cha88]
		Awareness	Feedback and awareness tools		Feedback tools for user privacy awareness [LHDL04, PK09, LBW08]
			User-friendly privacy support		Data removal tools (spyware removal, browser cleaning tools, activity traces eraser, harddisk data eraser)
Guarding association	Guard exposure	Compliance	Policies and Consents		Policy communication (P3P [W3C]), Policy enforcement (XACML [oo], EPAL [IBM])
			Notice and Transparency		/
		Confidentiality	Encryption		Symmetric key & public key encryption [MOV97], Deniable encryption [Nao02], Homomorphic encryption [FG07] , Verifiable encryption [CD98]
			Access control		Context-based access control [GMPT01], Privacy-aware access control [CF08, ACK+09]
		Minimization	Remove		/
			Hide	Receiver privacy	Private information retrieval [CGKS98], Oblivious transfer [Rab81, Cac98])
				Database privacy	Privacy preserving data mining [VBF+04, Pin02], Searchable encryption [ABC+05], Private search [OS05]
				General	see *guard exposure - confidentiality - encryption*
			Replace		/
			Generalize		K-anonymity model [Swe02b, Swe02a], l-Diversity [MGKV06]
	Maximize accuracy	Review data			/
		Update/ request deletion			/

description is the catalog's usability, it will improve the overall understanding of the threats which should impact the correctness of the identified threats.

Improved completeness. We believe that, when using the mapping table as a detailed checklist to determine the remaining threats that still need to be covered, the completeness rate will strongly increase. Many of the overlooked threats during the studies would have shown up as missing when the mapping table would have been properly used.
A second contributor to the improved completeness is the reduced interaction between LIND(D)UN and STRIDE. With a clear separation between security and privacy threats, we eliminate the confusion regarding the boundary between expected LIND(D)UN threats and remaining STRIDE threats. During the empirical studies, we expected the participants to evaluate both information disclosure threats and spoofing threats by applying the STRIDE trees, although they often considered them out of scope. This resulted in a set of overlooked threats (FNs). When providing a clear boundary as suggested by the improvements, no security threats will be expected when applying LIND(D)UN. The results of the second descriptive study (applying LINDDUN at architectural level) clearly show that both information disclosure and spoofing are on average the largest contributors to the number of false negatives. More precisely, 5 of the 10 FNs belong to a STRIDE category. Eliminating them would thus halve the number of original FNs which is a significant improvement. Note however that the security threats are also a large contributor to the number of correct threats (TP), especially information disclosure which has on average even 5 TPs. The overall recall will thus remain unchanged by this improvement, as also the number of TPs will be reduced to only half of its original size.

Improved productivity. We expect an improved productivity of LIND(D)UN, even though none of the proposed improvements directly targeted the methodology's execution time.
The extension of the catalog description was mainly targeting an increased ease of use. We nevertheless believe it is also beneficial for the look-up time of each tree node. The productivity is thus positively impacted by an improved usability of the catalog. Also the enhanced threat trees will not only result in a higher correctness rate, but the productivity will also benefit from it. By providing more structured trees with understandable threats, the relevant threats can be extracted quicker.

Improved ease of use. The extended catalog description will impact the ease of use.

The main usability-related attention point identified by the studies (i.e. lack of threat explanation) has been tackled by adding detailed information about each individual threat tree node and threat tree in general. We have also added a summary of each threat category in general to ease the comprehension. In addition, we have structured the threat tree information, to enhance the navigation and overall ease of use of the catalog itself.

Improved reliability. By explicitly including data minimization and inference, the main missing threats that were highlighted by the privacy experts are now represented in the LIND(D)UN methodology. By including the threats the experts identified but LINDDUN missed, the false negatives of LIND(D)UN will obviously decrease. We thus expect an increased reliability.

6.9 Conclusion

This chapter has provided an overview of LIND(D)UN, which incorporates several improvements made to the original LINDDUN methodology. These improvements have been informed by the results and observations of the empirical validation performed earlier.

As summarized in Table 6.1, the main changes include an injection of missing threats, a restructuring and extension of the catalog description, a repositioning of the security threats, a focus shift of the mapping table throughout the methodology and an update of several threat trees that originally caused some confusion. We also proposed an update of the later steps of the methodology, by introducing a taxonomy of privacy mitigation strategies. We believe that these changes improve the overall performance of LIND(D)UN, however further studies are required to validate this claim.

7

Conclusion

The age of privacy is dead. At least that is what Mark Zuckerberg has been stating for some years now [O'B10]. Unfortunately, if the mindset of today's society will not change, this statement might become reality soon. Most people take the current privacy-violating applications for granted and consider it a given fact that they themselves are the sole responsible for their own privacy. A common belief is that, if you do not want data to become public knowledge, you simply should not share this information at all [BBA+14]. Awareness of the consequences of sharing personal information is crucial, but privacy should not be limited to self-censorship. While the core of this idea, data minimization, is indeed a valid privacy approach, this responsibility should by no means be carried by the user alone. Indeed, the systems that collect, process, and share (personal) information should provide the appropriate measures to guarantee the user's privacy. These measures should not be included as an add-on, but should be incorporated in the development of the software system as early on as possible.

Unfortunately techniques to support this "Privacy by Design" approach are lacking. We therefore propose our own methodology to elicit privacy threats and their corresponding requirements and privacy enhancing solutions.

7.1 Contributions

This thesis makes the following three main contributions.

1. LINDDUN Privacy threat modeling technique. First of all, we presented the LINDDUN methodology, a threat modeling technique that encourages analysts to consider privacy issues in a systematic fashion.[1]

As LINDDUN is a model-based approach, first a data flow diagram is created to represent the information flows in the system based on the system description. This can be either rather high-level when executed during the requirements phase, or more detailed when performed in the software architecture phase. This DFD model is used as starting point and will be used throughout the methodology as guidance. Based on the model, a mapping table is created that indicates for each of the DFD elements which LINDDUN threat categories (i.e. linkability, identifiability, etc.) can be applicable. For each of the matches in the table, the corresponding LINDDUN threat tree is analyzed. If one or more leaf nodes pose a threat to the DFD element that is currently being examined, the elicited threats are documented using a misuse case template. These three problem-oriented steps are the core of LINDDUN. Nevertheless, the methodology also provides more solution-oriented support. If required, the elicited threats are prioritized according to their impact and likelihood. Next, the threats are translated into privacy requirements, after which these requirements are used to select appropriate privacy enhancing solutions to mitigate the uncovered threats.

2. Empirical validation of LINDDUN. During the creation of LINDDUN, we tested the methodology on some small examples, however this was only preliminary testing. LINDDUN, and privacy methodologies in general for that matter, are too abstract and complex to be fully automated. They need to be executed by analysts, and thus not only the methodology itself but also the role of the individuals should be taken into account during validation. We thus empirically evaluated the methodology to understand how LINDDUN is learned and applied. We therefore executed two descriptive studies and a case study to evaluate LINDDUN's correctness, completeness, productivity, ease of use, and reliability. As LINDDUN can be used throughout the entire software development lifecycle, our evaluation took into account the perspective of both requirements engineers and software architects.

[1] This contribution was joined work with Mina Deng.

First, we executed a small descriptive study to evaluate the performance of LINDDUN during the requirements phase. Three teams of seven participants in total applied LINDDUN to two software applications presented by industrial partners. Second, a more extensive descriptive study was executed to validate LINDDUN in the architectural phase. Twenty-seven teams of two participants applied LINDDUN to an industrial-size application. Finally, LINDDUN had to compete against privacy experts. Three privacy experts were asked to elicit privacy threats using their own procedures and these results were compared with those from LINDDUN.

Overall, LINDDUN performed well. A precision rate of 0.7 is encouraging. Also, the general feedback on the ease of use received from the participants was positive. The comparison of expert results with LINDDUN were also promising. Only two categories were missing (related to inference and data minimization) and LINDDUN even found some threats the experts overlooked. The rather low recall of 0.5 and productivity of 0.5 threat per hour are less reassuring.

3. Empirically-founded improvements of LINDDUN. The results of the empirical validation suggest that LINDDUN could benefit from some improvements.

The threat tree catalog was remodeled. The descriptions were extended, and structured according to the trees for easy retrieval. Also a summary of each threat category was added. The threat trees themselves have been updated too. Especially the unawareness tree has been altered to better suit the analysis. Also the linkability and identifiability trees have been restructured to ease navigation and were extended to include the missing expert threats. Concerning the methodology itself, the mapping table has gained more focus throughout the entire process. The integration with STRIDE on the other hand has been put to the background. Although both methodologies can be combined, they do not have to be used intertwined. They share a DFD as starting point, but the remainder of the methodologies can be executed in parallel. Note that this does not mean that security is put to the background. On the contrary, security is crucial for privacy. There is however no need to partially integrate it in LINDDUN, as it should receive a fully-fledged independent analysis by applying STRIDE to its full extent.

7.2 Reflections and Future Work

In this dissertation, we have introduced a privacy threat modeling framework, empirically evaluated it, and improved it based on our findings.

Given its correspondence to STRIDE, LINDDUN is also applicable to the same criticism. This section reflects on the LINDDUN methodology by assessing the disadvantages of the STRIDE methodology and provides a justification for the choices we made when creating LINDDUN. The section concludes by suggesting some future work.

7.2.1 Conformity with STRIDE

One of the criticisms toward LINDDUN is its conformity with STRIDE. Indeed, by closely following an existing technique, LINDDUN will not only inherit the positive features but also similar limitations will arise. This section tackles the most common critiques on the STRIDE and LINDDUN approaches.

Data Flow Diagram (DFD) as System Model. One of the contested features of STRIDE is related to the use of data flow diagrams (DFDs) as system model during the analysis. This representation was chosen due to two factors. First, DFDs are easy to understand. Second, they are data-centric, and thus focus on the flow of information through the system which is often the target of software attacks [Sho08]. Indeed, understanding the information flows is essential for a proper assessment of privacy threats [Inf14]. While DFDs are indeed not a common model in software development, the transition from more mainstream notations (such as UML) to a DFD is rather easy. During our empirical studies, we noticed that the translation of a component-connector diagram into a DFD was relatively straight-forward with an almost one-to-one mapping. Also other UML diagrams, such as context diagrams, show close resemblance with a DFD (although further analysis is required to examine the ease of transition). In addition, research exists that supports semi-automatic derivation of DFD models from code [AAWT07]. This is particularly interesting for threat modeling analysis of existing systems. The use of DFD as system representation should thus not be a restraining factor.

Quantitative and Qualitative Results. Also, as shown in the evaluation of LINDDUN, the quantitative studies show results which are in line with those of STRIDE (as discussed in Section 5.2.7). Although these results are promising, they are unfortunately not yet ideal. As even STRIDE, a well-established technique, does not obtain impeccable results in a quantitative evaluation, chance are low that LINDDUN will either. This result should however not be overly dramatized. The quantitative studies are only part of the methodology's validation. These results are indeed important to get an initial understanding of the soundness of the overall method and to identify possible improvements.

Nevertheless, the value of LINDDUN goes beyond the aspects that have been quantitatively assessed. STRIDE was chosen as basis for LINDDUN given its simplicity and close integration into the development process. LINDDUN also aims at simplicity by providing a structured approach with a rich privacy knowledge base that encourages people to reason about privacy issues in a systematic way. Future evaluation of these qualitative benefits are required, but the feedback which we already received from the requirements community is very optimistic. Indeed, the study of LINDDUN during the requirements phase (as described in Chapter 4) was part of a larger comparative study executed by the University of Trento [MPP]. They performed a qualitative analysis of LINDDUN and five security requirements techniques (CORAS [LSS11], Secure Tropos [MGM03], Security Argumentation [HLMN08], SI* [GMMZ05], and SREP [MFMP08]). Only LINDDUN and SREP were acknowledged by the study's participants as methods that help to identify privacy and security requirements respectively. While the other methods support brainstorming, they do not help to identify security requirements as they depend on the analyst's knowledge in security. LINDDUN and SREP, on the contrary, do not require a priori knowledge as they guide the analysts through the identification of privacy and security requirements.

Maturing Catalog of Threats. Technology keeps evolving and attackers keep coming up with new threats. Critics will claim that threat modeling techniques like STRIDE and LINDDUN that provide a knowledge base struggle with the lack of completeness. Indeed, the LINDDUN threat tree catalog will never be 100% complete, but keep in mind that other techniques cannot guarantee full completeness either. It is precisely this privacy knowledge base that brings added value to the methodology as we do not expect analysts who use LINDDUN to be privacy experts. Also, it is not the case that LINDDUN does not support evolving privacy concerns. The threat trees in the catalog are not fixed. On the contrary, the catalog is an open system that allows updates.

Neutrality w.r.t. Risk Assessment. Microsoft proposed a risk assessment technique called DREAD [Mica] to be used with STRIDE. It was however discontinued as for software-centric threat modeling it seems to add numbers without defining their scales [Sho08]. LINDDUN does not propose its own risk analysis technique. Instead, it describes risk in a general way as the product of the threat's impact and likelihood, and refers the analyst for further details to established techniques like CORAS [FKG+02] and OCTAVE [CMU]. The analysts can thus plug in their assessment method of choice.

Complementarity of LINDDUN and STRIDE. Finally note that the synergy between LINDDUN and STRIDE allows a combined privacy and security analysis with limited overhead as both methods can use the same DFD as starting point and apply a very similar structure. Being complementary to STRIDE is in fact one of LINDDUN's strengths, as it leads to easy adoption in the threat modeling community.

7.2.2 Future Work

This section proposes some future research trajectories related to the LINDDUN methodology.

Evaluation of the Improvements. The implemented changes to the LINDDUN methodology were based on the results of the empirical studies, and we feel rather confident that they will indeed improve the overall quality of LINDDUN. There is currently however no empirical proof. A first future research trajectory can thus include an empirical validation of the improved LINDDUN methodology. A new descriptive study can be performed where the conditions of the previous one are mimicked as precise as possible but where the new and improved LINDDUN is evaluated. These results can then be compared to the original ones to determine whether the changes indeed introduce improvement for our research questions.

Another interesting validation would be to further evaluate LINDDUN in a professional environment. Until now, the participants in the studies were mostly students. This was an obvious choice as LINDDUN was still a new upcoming methodology, however, as it is gaining attention in the privacy requirements community and the proposed improvements have resulted in a more streamlined version of LINDDUN, it will be interesting to examine how professionals in an industrial environment perceive LINDDUN.

Qualitative Evaluation. The results of the quantitative studies in this thesis are promising and the implemented improvements propose even better results. Nevertheless, this promising quantification is not the main advantage of LINDDUN. Indeed, LINDDUN is a methodology that encourages analysts and developers to systematically reflect on privacy issues. Although privacy is a complex concept in a possibly even more complex landscape, LINDDUN provides privacy knowledge, in a format which is easy to use, to deal with privacy. It will be interesting to examine these qualitative features of LINDDUN by comparing the methodology to other privacy methodologies. In this thesis, we

however decided to focus on a quantitative evaluation. First of all, we wanted to examine whether the basis of the LINDDUN methodology is solid and use the results of these studies to introduce improvements in order to make the methodology industry-ready. Secondly, at the time of the studies' design, there were no real competitors for LINDDUN in the privacy research community. LINDDUN's core advantage is its systematic approach which guides the analyst step-by-step through the analysis combined with its rich privacy knowledge base. We therefore want to compare LINDDUN with methods that provide both a structured analysis and sufficient privacy support, but such methods were missing. However, as new privacy requirements techniques are emerging, it would be worth examining these competing methodologies and comparing them to LINDDUN. How do these methodologies perform compared to LINDDUN? Do these methodologies include aspects that LINDDUN missed? All these questions are worth of a controlled experiment.

Catalog Maintenance. As mentioned before, privacy concerns keep evolving and maintenance of the threat tree catalog will be required. Ideally the trees should be updated regularly based on state of the art privacy solutions and emerging threats. We foresee, for example, an update of the trees related to non-repudiation and detectability in the near future. Although we did not receive any feedback regarding these trees from the empirical studies, this does not mean these trees could not use an update. Indeed, the lack of feedback was due to the application examined in the studies, as it did not result in any detectability or non-repudiation threats. When examining the trees in more detail, we however concluded that some of the nodes require some modification. Also the non-compliance tree might benefit from an extension. Research on compliance automation has been emerging to extract rules and obligations from regulatory documents and to provide traceability between such privacy policies and their corresponding implemented rules [BA08, SBSCB06, CHCGE10, YA10, AGR14]. As this is a promising domain, it could be included in an extension of our compliance tree.
In addition, also the solutions table could benefit from an update with state of the art privacy enhancing technologies.

Tool Support. As LINDDUN gains some level of popularity (see Section 3.9), it might be beneficial to add some tool support to the methodology to reach an even bigger audience. The tool could, similar to Microsoft's threat modeling tool [Micb], provide support to automatically generate the mapping table based on a DFD. It would also be worth investigating if the translation from component-connector diagram (or even other models) to a DFD can be supported by the LINDDUN tool. Furthermore, the other LINDDUN methodology steps could

also benefit from tool support. The threat tree catalog could, for example, be made interactive. Also, the mapping table could be incorporated as a checklist that is bound to the documented threats and assumptions. Even the extraction of mitigation strategies and the selection of privacy solutions could be included in a LINDDUN support tool.

As demonstrated in this thesis, LINDDUN is a solid privacy threat modeling methodology. These proposed ideas can contribute to an even more stable and industry-ready privacy threat modeling approach.

Bibliography

[AAWT07] Marwan Abi-Antoun, Daniel Wang, and Peter Torr. Checking threat modeling data flow diagrams for implementation conformance and security. In *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, ASE '07, pages 393–396. ACM, 2007.

[ABB+04] William Aiello, Steven M. Bellovin, Matt Blaze, Ran Canetti, John Ioannidis, Angelos D. Keromytis, and Omer Reingold. Just fast keying: Key agreement in a hostile internet. *ACM Transactions on Information and System Security (TISSEC)*, 7(2):242–273, 2004.

[ABC+05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *Advances in Cryptology (CRYPTO'05)*, volume 3621 of *LNCS*, pages 205–222. Springer, 2005.

[ACK+09] Claudio A. Ardagna, Jan Camenisch, Markulf Kohlweiss, Ronald Leenes, Gregory Neven, Bart Priem, Pierangela Samarati, Dieter Sommer, and Mario Verdicchio. Exploiting cryptography for privacy-enhanced access control: A result of the PRIME project. *Journal of Computer Security*, 2009.

[AER02] A.I. Antón, J.B. Earp, and A. Reese. Analyzing website privacy requirements using a privacy goal taxonomy. In *Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on*, pages 23–31, 2002.

[AF04] Martín Abadia and Cédric Fournet. Private authentication. *Theoretical Computer Science*, 322(3):427–476, September 2004.

[Agr99] Philip E Agre. The architecture of identity: Embedding privacy in market institutions. *Information, Communication & Society*, 2(1):1–25, 1999.

[AGR13] P. Anthonysamy, P. Greenwood, and A. Rashid. Social networking privacy: Understanding the disconnect from policy to controls. *Computer*, 46(6):60–67, 2013.

[AGR14] Pauline Anthonysamy, Phil Greenwood, and Awais Rashid. A method for analysing traceability between privacy policies and privacy controls of online social networks. In *Privacy Technologies and Policy*, volume 8319 of *LNCS*, pages 187–202. Springer, 2014.

[Ale03] Ian Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20(1):58–66, 2003.

[AP98] Ross Anderson and Fabien Petitcolas. On the limits of steganography. *IEEE Journal of Selected Areas in Communications*, 16:474–481, 1998.

[BA08] Travis Breaux and Annie Antón. Analyzing regulatory rules for privacy and security requirements. *IEEE Transactions on Software Engineering*, 34(1):5–20, January 2008.

[Bac] André Bacard. Anonymous.to: Cypherpunk tutorial. http://www.andrebacard.com/remail.html.

[BBA+14] Ero Balsa, Laura Brandimarte, Alessandro Acquisti, Claudia Diaz, and Seda Gürses. Spiny CACTOS: OSN users attitudes and perceptions towards cryptographic access control tools. In *Proceedings of Workshop on Usable Security (USEC 2014)*, pages 1–10, 2014.

[BC93] Stefan Brands and David Chaum. Distance-bounding protocols (extended abstract). In *Advances in Cryptology (EUROCRYPT'93)*, volume 765 of *LNCS*, pages 344–359. Springer, 1993.

[BCK05] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 2005.

[Bec12] Kristian Beckers. Comparing privacy requirements engineering approaches. In *Proceedings of the Seventh International Conference on Availability, Reliability and Security (ARES)*, pages 574–581. IEEE, 2012.

[BF11] Liana B. Baker and Jim Finkle. Sony PlayStation suffers massive data breach, April 2011. Online at http://www.reuters.com/article/2011/04/26/us-sony-stoldendata-idUSTRE73P6WB20110426.

[BFHM14] Kristian Beckers, Stephan Faßbender, Maritta Heisel, and Rene Meis. A Problem-based Approach for Computer Aided Privacy Threat Identification. In *Privacy Technologies and Policy*, volume 8319 of *LNCS*, pages 1–16. Springer, 2014.

[BFK00] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, LNCS, pages 115–129. Springer, 2000.

[BGB04] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use PGP. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 77–84. ACM New York, NY, USA, 2004.

[BGS01] A. Back, I. Goldberg, and A. Shostack. Freedom systems 2.1 security issues and analysis. White paper, Zero Knowledge Systems, Inc., May 2001.

[bpm] BPMN website. http://www.bpmn.org/.

[BS93] Victoria Bellotti and Abigail Sellen. Design for privacy in ubiquitous computing environments. In *Proceedings of the Third Conference on European Conference on Computer-Supported Cooperative Work*, pages 77–92, 1993.

[Cac98] Christian Cachin. On the foundations of oblivious transfer. In *Advances in Cryptology (EUROCRYPT'98)*, pages 361–374. Springer, LNCS 1403, 1998.

[Cav09] A. Cavoukian. Privacy by design: The 7 foundational principles. *Information and Privacy Commissioner of Ontario, Canada*, 2009.

[Cav10] Ann Cavoukian. Privacy by Design : Achieving the gold standard in data protection for the smart grid (2010). Technical report, Information and Privacy Commissioner of Ontario, 2010.

[CD98] J. Camenisch and I. Damgard. Verifiable encryption and applications to group signatures and signature sharing. In *Technical Report RS-98-32, BRICS, Department of Computer Science, University of Aarhus*, Dec. 1998.

[CF08] Barbara Carminati and Elena Ferrari. Privacy-aware collaborative access control in web-based social networks. In *Proceedings of the 22nd IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC2008)*, pages 81–96. Springer, 2008.

[CGKS98] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Journal of the ACM*, pages 41–50, 1998.

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[CHCGE10] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker. A machine learning approach for tracing regulatory codes to product specific requirements. In *Proceedings of the 2010 ACM/IEEE 32nd International Conference on Software Engineering*, volume 1, pages 155–164, May 2010.

[CJMS10] Jeffrey C. Carver, Letizia Jaccheri, Sandro Morasca, and Forrest Shull. A checklist for integrating student empirical studies with research and teaching goals. *Empirical Software Engineering*, 15(1):35–59, 2010.

[CK13] Colette Cuijpers and Bert-Jaap Koops. Smart metering and privacy in europe: Lessons from the dutch case. In *European Data Protection: Coming of Age*, pages 269–293. Springer, 2013.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology (CRYPTO'04)*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004.

[CMU] CMU Software Engineering Institute. OCTAVE. http://www.cert.org/octave/.

[cob] COBIT 5: A business framework for the governance and management of enterprise IT. http://www.isaca.org/COBIT,.

[CPHH02] Sebastian Clauß, Andreas Pfitzmann, Marit Hansen, and Els Van Herreweghen. Privacy-enhancing identity management. The IPTS Report 67, 8-16, September 2002.

[Dan12] George Danezis. An introduction to privacy technologies (presentation), 2012. Online at http://www.cl.cam.ac.uk/teaching/1112/SecurityII/2012-security2-4-danezis.pdf.

[DB92] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work (CSCW '92)*, pages 107–114. ACM, 1992.

[DBPP09] Mina Deng, Tiziano Bianchi, Aalessandro Piva, and Bart Preneel. An efficient buyer-seller watermarking protocol based on composite signal representation. In *Proceedings of 11th ACM workshop on Multimedia and Security*, pages 9–18, 2009.

[DDS09] George Danezis, Claudia Diaz, and Paul Syverson. Systems for anonymous communication. *Handbook of Financial Cryptography and Security, Cryptography and Network Security Series*, pages 341–389, 2009.

[DHG06] Paul De Hert and Serge Gutwirth. Privacy, data protection and law enforcement. opacity of the individual and transparency of power. *Privacy and the criminal law*, pages 61–104, 2006.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[DWS+11] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering Journal*, 16(1):3–32, 2011.

[End95] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[Eur95] European Parliament. 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., 1995.

[Eur02] European Parliament. Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), 2002.

[Eur09] European Parliament. Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009 amending Directive 2002/22/EC on universal service and users' rights relating to electronic communications networks and services, Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector and Regulation (EC) No 2006/2004 on cooperation between national authorities responsible for the enforcement of consumer protection laws, 2009.

[Eur12] European Commission. Commission proposes a comprehensive reform of the data protection rules, 2012. Online at http://ec.europa.eu/justice/newsroom/data-protection/news/120125_en.htm.

[FG07] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for non-specialists. *EURASIP Journal on Information Security*, pages 41–50, 2007.

[fip73] The code of Fair Information Practices. Technical report, U.S. Department of Health, Education and Welfare, Secretary's Advisory Committee on Automated Personal Data Systems, Records, Computers, and the Rights of Citizens viii, 1973.

[FKG+02] Rune Fredriksen, Monica Kristiansen, Bjørn Axel Gran, Ketil Stølen, Tom Arthur Opperud, and Theodosis Dimitrakos. The coras framework for a model-based risk management process. In *Proceedings of the 21st International Conference on Computer Safety, Reliability and Security table of contents*, volume 2434 of *LNCS*, pages 94–105. Springer, 2002.

[fmeon] Procedures for performing a failure mode effect and critical analysis. Technical report, United States Department of Defense, 1980 (revision).

[GÏ3] Seda Gürses. An introduction to multilateral privacy requirements analysis, 2013. http://securitylab.disi.unitn.it/lib/exe/fetch.php?media=research_activities:erise:erise_2013:erise2013-mpra-overview.pdf.

[GGTD11] Seda Gürses, Carmela Gonzalez Troncoso, and Claudia Diaz. Engineering privacy by design. In *Proceedings of the Conference on Computers, Privacy & Data Protection (CPDP 2011)*, page 25, 2011.

[GMMZ05] Paolo Giorgini, Fabio Massacci, John Mylopoulos, and Nicola Zannone. Modeling security requirements through ownership, permission and delegation. In *Proceedings of the 3th IEEE International Conference on Requirements Engineering*, pages 167–176. IEEE, 2005.

[GMPT01] Christos K. Georgiadis, Ioannis Mavridis, George Pangalos, and Roshan K. Thomas. Flexible team-based access control using contexts. In *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, SACMAT '01, pages 21–27. ACM, 2001.

[GRS96] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information. In *Proceedings of Information Hiding: First International Workshop*, volume 1174 of *LNCS*, pages 137–150. Springer, May 1996.

[Gür10] Fahriye Seda Gürses. *Multilateral privacy requirements analysis in online social network services.* PhD thesis, Department of Computer Science, KU Leuven, 2010.

[GZ09] Paolo Guarda and Nicola Zannone. Towards the development of privacy-aware systems. *Information and Software Technology*, 51(2):337–350, 2009.

[haz02] British Standard BS: IEC61882:2002 Hazard and Operability studies (hazop studies)- Application Guide. Technical report, British Standards Institution, 2002.

[HBC+04] Marit Hansen, Peter Berlich, Jan Camenisch, Sebastian Clauß, Andreas Pfitzmann, and Michael Waidner. Privacy-enhancing identity management. *Information Security Technical Report (ISTR)*, 9(1):35–44, 2004.

[HBQ12] Stefan Hofbauer, Kristian Beckers, and Gerald Quirchmayr. Conducting a privacy impact analysis for the analysis of communication records. *Perspectives in Business Informatics Research*, pages 148–161, 2012.

[HIP06] HIPAA administrative simplification: Enforcement; final rule. *Federal Register - Rules and Regulations*, 71(32), 2006.

[HL06] Michael Howard and Steve Lipner. *The Security Development Lifecycle*. Microsoft Press, Redmond, WA, USA, 2006.

[HLMN08] Charles B Haley, Robin Laney, Jonathan D Moffett, and Bashar Nuseibeh. Security requirements engineering: A framework for representation and analysis. *IEEE Transactions on Software Engineering*, 34(1):133–153, 2008.

[HNL04] Jason I. Hong, Jennifer D. Ng, and Scott Lederer. Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *Designing Interactive Systems (DIS2004)*, pages 91–100. ACM Press, 2004.

[hor14] Digital security: Cybersecurity, privacy and trust, 2014. EU Research and Innovation programme, Work package DS-01-2014.

[IBM] IBM. Enterprise Privacy Authorization Language (EPAL 1.2). W3C Member Submission, 10 November 2003.

[Inf14] Information Commissioner's Office (ICO). Conducting privacy impact assessments code of practice (version 1.0). Technical report, UK, 2014.

[Jen05] Carlos Jensen. *Designing for privacy in interactive systems*. PhD thesis, Georgia Institute of Technology, 2005.

[kim] Kimai - open source time tracking. http://www.kimai.org/en/.

[KKG08] Christos Kalloniatis, Evangelia Kavakli, and Stefanos Gritzalis. Addressing privacy requirements in system design: the PriS method. *Requirements Engineering*, 13(3):241–255, 2008.

[KL14] Bert-Jaap Koops and Ronald Leenes. Privacy regulation cannot be hardcoded. a critical comment on the 'privacy by design' provision in data-protection law. *International Review of Law, Computers & Technology*, 28(2):159–171, 2014.

[KLK+14] Benedicto Rodriguez Kuhn, Ashish Lamichhane, Stefan Klein, Michael Raess, Christoph Lengger, Philipp Gormanns, Christian Ohmann, Wolfgang Kuchinke, Ugis Sarkans, Murat Sariyar, et al. Report describing the security architecture and framework (deliverable 5.3). Technical report, Building Data Bridges between Biological and Medical Infrastructures in Europe (BioMedBridges FP7-INFRASTRUCTURES-284209), 2014.

[KLM+97] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. *Aspect-oriented programming*. Springer, 1997.

[KM01] Darko Kirovski and Henrique Malvar. Robust covert communication over a public audio channel using spread spectrum. In *Information Hiding*, volume 2137 of *LNCS*, pages 354–368. Springer, 2001.

[Kre13] David Krebs. Privacy by design: Nice-to-have or a necessary principle of data protection law. 4, 2013.

[Lan] Susan Landau. Privacy: It's all in the use case. Keynote at 2011 Annual Computer Security Applications Conference (ACSAC 2011). http://www.acsac.org/2011/program/keynotes/#dp.

[Lan01] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, UbiComp '01, pages 273–291, 2001.

[LBW08] Heather Richter Lipford, Andrew Besmer, and Jason Watson. Understanding privacy settings in facebook with an audience view. In *UPSEC*. USENIX Association, 2008.

[LHDL04] Scott Lederer, Jason I. Hong, Anind K. Dey, and James A. Landay. Personal privacy through understanding and action: Five pitfalls for designers. *Personal and Ubiquitous Computing*, 8:440–454, 2004.

[LIN] LINDDUN threat tree catalog. https://distrinet.cs.kuleuven.be/software/linddun/catalog.php.

[LSK12] J. Luna, N. Suri, and I. Krontiris. Privacy-by-design based on quantitative threat modeling. In *7th International Conference on Risks and Security of Internet and Systems (CRiSIS 2012)*, pages 1–8, October 2012.

[LSS11] Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. A guided tour of the coras method. In *Model-Driven Risk Analysis*, pages 23–43. Springer, 2011.

[LYM03] Lin Liu, Eric Yu, and John Mylopoulos. Security and privacy requirements analysis within a social setting. *11th IEEE International Conference on Requirements Engineering*, pages 151–161, 2003.

[MF99] J. McDermott and C. Fox. Using abuse case models for security requirements analysis. In *Proceedings of the 15th Annual Computer Security Applications Conference*, ACSAC '99, pages 55–64. IEEE Computer Society, 1999.

[MFMP08] Daniel Mellado, Eduardo Fernández-Medina, and Mario Piattini. Towards security requirements management for software product lines: A security domain requirements engineering process. *Computer Standards & Interfaces*, 30(6):361–371, 2008.

[MGK09] Erika McCallister, Tim Grance, and Karen Kent. Guide to protecting the confidentiality of personally identifiable information (PII) (draft). Technical report, National Institute of Standards and Technology (U.S.), 2009.

[MGKV06] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, page 24, 2006.

[MGM03] Haralambos Mouratidis, Paolo Giorgini, and Gordon Manson. Integrating security and systems engineering: Towards the modelling of secure information systems. In *Advanced Information Systems Engineering*, pages 63–78. Springer, 2003.

[Mica] Microsoft. Improving web application security: Threats and countermeasures. http://msdn.microsoft.com/en-us/library/aa302419.aspx#c03618429_011.

[Micb] Microsoft. Microsoft SDL threat modeling tool. http://www.microsoft.com/en-us/download/details.aspx?id=42518.

[mic08] Privacy guidelines for developing software products and services, version 3.1. Technical report, Microsoft Coorporation, Sept 2008. http://download.microsoft.com/download/0/8/2/082448D8-2AED-45BC-A9A0-094840E9E3A2/Microsoft_and%20Privacy_guidelines_for_developers.doc.

[MIKG13] Haralambos Mouratidis, Shareeful Islam, Christos Kalloniatis, and Stefanos Gritzalis. A framework to support selection of cloud providers based on security and privacy requirements. *Journal of Systems and Software*, pages 2276–2293, 2013.

[Mixa] Mixmaster. Mixmaster homepage. http://mixmaster.sourceforge.net/.

[Mixb] Mixminion. Mixminion official site. http://mixminion.net/.

[ML14] Theodore Mouroutis and Athanasios Lioumpas. Use-cases definition and threat analysis (deliverable d2.1). Technical report, Reliable, resilient and secure IoT for smart city applications (RERUM FP7-ICT-609094), 2014.

[MMZ08] Seiya Miyazaki, Nancy Mead, and Justin Zhan. Computer-aided privacy requirements elicitation technique. *Asia-Pacific Conference on Services Computing (APSCC'08)*, pages 367–372, 2008.

[MNCM03] Ira Moskowitz, Richard E. Newman, Daniel P. Crepeau, and Allen R. Miller. Covert channels and anonymizing networks. In *In Workshop on Privacy in the Electronic Society*, pages 79–88. ACM, 2003.

[MOV97] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. Handbook of applied cryptography, 1997. http://www.cacr.math.uwaterloo.ca/hac/.

[MP12] Fabio Massacci and Federica Paci. How to select a security requirements method? A comparative study with students and practitioners. In *Proceedings of the 17th Nordic conference on Secure IT Systems (NordSec'12)*, pages 89–104. Springer, 2012.

[MPP] Fabio Massacci, Federica Paci, and Pieter Philippaerts. Second report describing the second annual summer school and the annual open competitions (deliverable D13.3. Technical report.

[MTJ10] Per Håkon Meland, Inger Anne Tøndel, and Jostein Jensen. Idea: reusability of threat models – two approaches with an experimental evaluation. In *Proceedings of the International conference on Engineering Secure Software and Systems (ESSoS)*, pages 114–122. Springer, 2010.

[Nao02] Moni Naor. Deniable ring authentication. In *Advances in Cryptology (CRYPTO'02)*, volume 2442 of *LNCS*, pages 481–498. Springer, 2002.

[NIS] NIST. Risk management guide for information technology systems, special publication 800-30. http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf.

[Nis04] Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79:119–158, 2004.

[NM90] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, pages 249–256, 1990.

[nmb13] NMBS data leak was breach of privacy, January 2013. Online at http://www.flanderstoday.eu/business/nmbs-data-leak-was-breach-privacy.

[NN01] Moni Naor and Kobbi Nissim. Communication preserving protocols for secure function evaluation. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing (STOC '01)*, pages 590–599. ACM, 2001.

[Nus01] B. Nuseibeh. Weaving together requirements and architectures. *Computer*, 34(3):115–119, 2001.

[O'B10] Terrence O'Brien. Facebook's Mark Zuckerberg claims privacy is dead, January 2010. Online at http://www.switched.com/2010/01/11/facebooks-mark-zuckerberg-claims-privacy-is-dead/.

[OCS+13] Inah Omoronyia, Luca Cavallaro, Mazeiar Salehie, Liliana Pasquale, and Bashar Nuseibeh. Engineering adaptive privacy: on the role of privacy awareness requirements. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 632–641. IEEE Press, 2013.

[OEC80] OECD. Guidelines on the protection of privacy and transborder flows of personal data, organization for economic cooperation and development, 1980. http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html.

[oo] OASIS (oasis open.org). XACML 3.0 - work in progress, retrieved 09-september-2009. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#CURRENT.

[OS05] Rafail Ostrovsky and William E. Skeith. Private searching on streaming data. In *Advances in Cryptology (CRYPTO'05)*, volume 3621 of *LNCS*, pages 223–240. Springer, 2005.

[OS09] A.L. Opdahl and G. Sindre. Experimental comparison of attack trees and misuse cases for security threat identification. *Information and Software Technology*, 51(5):916–932, 2009.

[OWA] OWASP. Risk rating methodology. http://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology.

[PET] Annual symposium on privacy enhancing technologies (PETs). http://petsymposium.org/.

[PH10] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (version 0.33 april 2010). Technical report, TU Dresden and ULD Kiel, 2010.

[Pin02] B. Pinkas. Cryptographic techniques for privacy preserving data mining. *SIGKDD Explorations*, 4(2):12–19, 2002.

[PK03] Andrew Patrick and Steve Kenny. From privacy legislation to interface design: Implementing information privacy in human-computer interactions. In *Privacy Enhancing Technologies*, volume 2760 of *LNCS*, pages 107–124. Springer, 2003.

[PK09] Sameer Patil and Alfred Kobsa. *Privacy Considerations in Awareness Systems: Designing with Privacy in Mind*, chapter 8, pages 187–206. Human–Computer Interaction Series. Springer, June 2009.

[PPW91] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-mixes: Untraceable communication with very small bandwidth overhead. In *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*, pages 451–463, 1991.

[PW85] Andreas Pfitzmann and Michael Waidner. Networks without user observability – design options. In *Advances in Cryptology (EUROCRYPT'85)*, volume 219 of *LNCS*, pages 245–253. Springer, 1985.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[Roe97] Michael Roe. *Cryptography and Evidence*. PhD thesis, University of Cambridge, Clare College, 1997.

[RP09] Antoinette Rouvroy and Yves Poullet. The right to informational self-determination and the value of self-development: Reassessing the importance of privacy for democracy. In *Reinventing Data Protection?*, pages 45–76. Springer, 2009.

[RR98] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[SBSCB06] Anna Squicciarini, Abhilasha Bhargav-Spantzel, Alexei Czeskis, and Elisa Bertino. Traceable and automatic compliance of privacy policies in federated digital identity management. In *Privacy Enhancing Technologies*, volume 4258 of *LNCS*, pages 78–98. Springer, 2006.

[SC09] Sarah Spiekermann and Lorrie F. Cranor. Engineering privacy. *IEEE Transactions on Software Engineering*, 35(1):67–82, 2009.

[Sch00] Bruce Schneier. *Secrets & Lies: Digital Security in a Networked World*. Wiley, 2000.

[Sch10] Peter Schaar. Privacy by design. *Identity in the Information Society*, 3(2):267–274, 2010.

[sdl08] Microsoft security development lifecycle (SDL) version 3.2. Technical report, Microsoft Coorporation, april 2008. http://www.microsoft.com/Downloads/details.aspx?familyid=2412C443-27F6-4AAC-9883-F55BA5B01814&displaylang=en.

[Sho08] Adam Shostack. Experiences threat modeling at Microsoft. Technical report, Microsoft, 2008.

[Sho14] Adam Shostack. *Threat Modeling. Designing for Security*. Wiley, 2014.

[SO01] Guttorm Sindre and Andreas L Opdahl. Templates for misuse case description. pages 1–13, 2001.

[SO05] Guttorm Sindre and Andreas L Opdahl. Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1):34–44, 2005.

[Soh98] Markus Sohlenkamp. *Supporting Group Awareness in Multi-User Environments Through Perceptualization*. PhD thesis, Universität Paderborn, Germany, 1998.

[Sol06] Daniel J. Solove. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477, 2006.

[STP09] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pages 188–203. IEEE Computer Society, 2009.

[Swe02a] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[Swe02b] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[SWJ13] Riccardo Scandariato, Kim Wuyts, and Wouter Joosen. A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering*, pages 1–18, 2013.

[SZAG12] F. Siddiqui, S. Zeadally, C. Alcaraz, and S. Galvao. Smart grid privacy: Issues and solutions. In *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–5, July 2012.

[Tic00] Walter F. Tichy. Hints for reviewing empirical work in software engineering. *Empirical Software Engineering*, 5(4):309–312, 2000.

[Tso10] Alexia Tsosis. @alexia on Twitter, October 2010.

[Tuc13] Patrick Tucker. Has big data made anonymity impossible?, May 2013. MIT Technology Review, Online at http://www.technologyreview.com/news/514351/has-big-data-made-anonymity-impossible/.

[uml] UML website. http://www.bpmn.org/.

[Uni] University of Trento (Italy). eRISE Challenge 2012 - experimental material. http://securitylab.disi.unitn.it/doku.php?id=erise_2012.

[VBF+04] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 3(1):50–57, Mar. 2004.

[VL09] Axel Van Lamsweerde. *Requirements engineering: from system goals to UML models to software specifications*. Wiley, 2009.

[vRGB+95] Henk van Rossum, Huib Gardeniers, John Borking, Ann Cavoukian, John Brans, Noel Muttupulle, and Nick Magistrale. *Privacy-Enhancing Technologies: The Path to Anonymity*. Information and Privacy Commissioner / Ontario, Canada & Registratiekamer, The Netherlands, Den Haag, August 1995.

[W3C] W3C. Platform for privacy preferences (p3p) project. http://www.w3.org/P3P/.

[WB90] Samuel D Warren and Louis D Brandeis. The right to privacy. *Harvard law review*, pages 193–220, 1890.

[Wes70] Alan F. Westin. *Privacy and Freedom*. Bodley Head, 1970.

[WP90] Michael Waidner and Birgit Pfitzmann. The dining cryptographers in the disco: Unconditional sender and recipient untraceability. In *Advances in Cryptology (EUROCRYPT'89)*, pages 690–690. Springer, LNCS 434, 1990.

[WSDDJ09] Kim Wuyts, Riccardo Scandariato, Bart De Decker, and Wouter Joosen. Linking privacy solutions to developer goals. In *Proceedings of the Fourth International Conference on Availability, Security and Reliability*, pages 847–852. IEEE Computer Society, March 2009.

[WSJ] Kim Wuyts, Riccardo Scandariato, and Wouter Joosen. LINDDUN evaluation - experimental material. https://sites.google.com/site/linddunstudy/.

[WSJ14a] Kim Wuyts, Riccardo Scandariato, and Wouter Joosen. Empirical evaluation of a privacy-focused threat modeling methodology. *The Journal of Systems and Software*, 96:122–138, October 2014.

[WSJ14b] Kim Wuyts, Riccardo Scandariato, and Wouter Joosen. LIND(D)UN privacy threat tree catalog. CW Reports CW675, Department of Computer Science, KU Leuven, September 2014.

[YA10] J.D. Young and A.I. Anton. A method for identifying software requirements based on policy commitments. In *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference (RE)*, pages 47–56, Sept 2010.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations. In *Proceedings of Twenty-third IEEE Symposium on Foundations ofComputer Science*, pages 160–164, November 1982.

[YC02] Eric Yu and Luiz Marcio Cysneiros. Designing for privacy and other competing requirements. In *Proceedings of the 2nd Symposium on Requirements Engineering for Information Security (SREIS)*, pages 15–16, 2002.

[Yu97] Eric S.K. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, pages 226–235, 1997.

List of publications

Journal Papers

Kim Wuyts, Riccardo Scandariato, Wouter Joosen, *Empirical evaluation of a privacy-focused threat modeling methodology*, The Journal of Systems and Software, volume 96, pages 122-138, 2014

Riccardo Scandariato, Kim Wuyts, Wouter Joosen, *A descriptive study of Microsoft's threat modeling technique*, Requirements Engineering, pages 1-18, 2013

Kim Wuyts, Griet Verhenneman, Riccardo Scandariato, Wouter Joosen, Jos Dumortier, *What electronic health records don't know just yet. A privacy analysis for patient communities and health records interaction*, Health and Technology, volume 2, issue 3, pages 159-183, 2012

Kim Wuyts, Riccardo Scandariato, Griet Verhenneman, Wouter Joosen, *Integrating patient consent in e-health access control*, International Journal of Secure Software Engineering, volume 2, issue 2, pages 1-24, 2011

Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, Wouter Joosen, *A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements*, Requirements Engineering, volume 16, issue 1, pages 3-32, 2011

International Conferences and Workshops

Kim Wuyts, Riccardo Scandariato, Bart De Decker, Wouter Joosen, *Linking privacy solutions to developer goals*, Third International Workshop on Secure Software Engineering (SecSE), pages 847-852, Fukuoka, Japan, March 2009

Kim Wuyts, Riccardo Scandariato, Geert Claeys, Wouter Joosen, *Hardening XDS-based architectures*, Third International Conference on Availability, Security and Reliability (ARES), pages 18-25, Barcelona, Spain, March 2008

Technical Reports

Kim Wuyts, Riccardo Scandariato, Wouter Joosen, *LIND(D)UN privacy threat tree catalog*, CW Reports, volume CW675, 41 pages, Department of Computer Science, K.U.Leuven, Leuven, Belgium, September 2014

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
IMINDS - DISTRINET
Celestijnenlaan 200A box 2402
B-3001 Heverlee