Privacy throughout the Data Cycle

Elisa Costante

Printed by Proefschriftmaken.nl - Uitgeverij BOXPress

Cover design by Valerio Vincenzo Guarino.

Privacy throughout the Data Cycle by Elisa Costante.

This work has been partially funded by the Dutch program COMMIT under the THeCS project.

The work in the thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

COMMIT/

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3810-2
NUR: 980
Subject heading: Computer Security.

Privacy throughout the Data Cycle

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 31 maart 2015 om 16:00 uur

door

Elisa Costante

geboren te Avellino, Italië

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. M. van den Brand
1e promotor:	prof.dr. M. Petković
2e promotor:	prof.dr. S. Etalle
copromotor(en):	dr. J.I den Hartog
leden:	prof.dr. G. Canfora (Università degli Studi del Sannio)
	prof.dr. D.W. Chadwick (University of Kent)
	prof.dr.ir. W.M.P. van der Aalst

Al mio picci.

Acknowledgments

The journey of a PhD student is not an easy one. It looks more like a roller coaster ride rather than a calm walk on the beach. While I am writing these words, I can see the finish line and I can't help thinking of who has been with me along the way.

First of all, I want to express my gratitude to the committee members for taking part to my defense and for reviewing my thesis. I thank prof. Mark van den Brand for chairing the committee and prof. David Chadwick for carefully reading my thesis and providing me with a long list of valuable comments. I thank prof. Gerardo Canfora for his insights and for following my academic career since the very first steps of my bachelor years. I also want to thank prof. Wil van der Aalst for the interesting suggestions he provided especially regarding future directions of my research.

Words are not enough to thank my supervisors, firstly for accepting me as one of their pupils and then for following and guiding my steps in academy. Prof. Milan Petković was always encouraging my research and he allowed me to grow as a research scientist. Prof. Sandro Etalle has been my inspiration for many reasons: his charisma and his passion for science gave me the enthusiasm I needed during the hardest moments. Over time he became a dear friend, someone I can rely on when struggling with hard decisions. I can't be more grateful to Jerry den Hartog for everything he did during these years. He was the first face I saw when I arrived to The Netherlands to start this journey. Thank you Jerry, for welcoming me in your house, for guiding me, and for giving me time and advice every time I needed them. I also want to thank Nicola Zannone who, even though was not formally my supervisor, has always been there to listen to my questions when the lights of the other offices were gradually switching off. I thank Mykola Pechenizkiy and Federica Paci, for the

fruitful collaborations we had. I will always be thankful to Damiano Bolzoni and the SecurityMatters' team for allowing me to carry out the most interesting part of my research and for finally welcoming me on board.

The journey would have not been as fun and joyful without the friends I made along the way. I thank all the members of the SEC group, the old and the new ones, everybody with whom I shared a moment of joy, an interesting discussion or a loud laugh. I thank Jolande, for the smile she kept for me every morning, for being the best secretary ever and the closest thing to a mum that The Netherlands gave me. I thank Bruno and Daniel for the uncountable jokes they shared with (and mostly on) me. Meeting you guys has enriched my life (although it might have seriously damaged my self-esteem). I thank Mayla e Antonino for our many and long coffee breaks, for being always available to comfort me and for preparing the most delicious dinners I ever had in town. I thank Antonio for his constant encouragement and for his capability of remembering everything I ever said (a feature that in some occasions has been not that useful though). I thank Alberto for being one of the kindest and most attentive persons I know, capable of many touching gestures.

I thank the dearest friends I made in Eindhoven, for every moment we spent together. Thanks to Luis, Jorge, Andrea, and my sweet-silly Li, for your support and for the many sources of distraction you provided me with during the last years. I can't conclude this section without thanking my old friends, *i ragazzi del bar la buca*, that are always there to listen to my problems, to my complaints or to my funny stories. Thank you guys for making me feel you are close despite the distance. Especially, I thank Valerio for creating the cover of this thesis, working late for several nights. My gratitude "*my friend*" cannot be expressed with words.

My biggest thank goes to my family: to my mum, my dad and my brother for never letting me down, for accepting my decisions wherever they brought and will bring me and for always be there, no matter what.

Finally, I thank Juan Carlos, the one that supported me day by day, that comforted me when I was disheartened, that was full of joy whenever I reached an important accomplishment. The one that I am sure today will be more proud of me than anybody else (excepting my mother, maybe). Honey, I could have not made it without you. Think about this every time you will have to call me *Doctor*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last years we have witnessed an explosion in the use and in the availability of web-based services. From the apps we use on our smartphones, to the interfaces we access for our daily activities: almost everything is based on some kind of web service. Online services are intensively used by businesses to strengthen relationships with their customers and to provide them with tailored solutions. In particular, offering services which meet specific customers' needs has become essential for businesses to maintain and improve competitiveness.

Personalisation represents a key word in this context: to be effective, services are very often automatically tailored to the specific user. Everything on the web seems to be personalised: from the advertisements users receive while browsing, to the order in which articles are displayed on a news website or the results returned by a search engine. The level of detail of the personalization has increased dramatically in the last years. For instance, search results are now enriched with information retrieved from a user's e-mails, calendar and social networks. It is not a coincidence that *big data* is one of the buzzword of today. Big data is more often than not personal data, used to tailor business offerings in a way that was not conceivable just a few years ago.

While the use of personalised services improves the overall user experience and boosts business possibilities, it brings along a number of threats to privacy. To build customized services, providers rely on the collection of large quantities of personal information, like name, surname, bank details, photos, private messages or people's location. Privacy violations can occur in case this personal data is stolen, disclosed to unauthorized parties or used for purposes different from the ones for which per-

mission was granted. To give just one example, a serious privacy violation has been uncovered in conjunction with the NSA case[1] that revealed how governmental agencies are able to access data stored by major technology companies, often without individuals' awareness nor the presence of a warrant.

The explosion in use of personal data, together with the increasing number of privacy violations has raised governments attention towards privacy issues. Nowadays, both the United States and the European Union have legislation that regulates privacy. European Directive 95/46/EC defines *personal data* as "any information relating to an identified or identifiable natural person; 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". In addition, the European Directive on Privacy and Electronic Communications 2002/58/EC defines a *personal data breach* (or *privacy breach*) as a "breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to, personal data". As we see, these are broad concepts. More in general, the European Union, which is traditionally stricter than the US when it comes to privacy, declared privacy as a fundamental right of its citizens as stated in its Charter of the Fundamental Rights [1].

As a result of these (and other) laws, organizations that handle personal data are obliged to implement privacy preserving measures. For example, it is mandatory for them to publish a privacy policy that clearly states what data is collected and how it will be handled. Also, in case of a data breach involving personal information, privacy regulations mandate a timely notification to competent authorities.

Protecting private data is thus important for users and for business alike; in particular, to allow users and businesses to fully benefit from advances in online service provision, it is necessary to have solutions granting secure exchange of private information as well as solutions for preventing or detecting data breaches.

From Collection to Processing

The first step in the treatment of personal data is clearly the collection of data. This can take place in several ways. Signing up for a service or shopping online usually requires the user to provide personal data such as name, e-mail or address. Applications which run on smart phones or on wearable devices, e.g. to provide driving directions or to monitor sport activities, might collect location data as GPS coordinates or health data as the hearth rate. Websites typically track users' activities such as the pages they visit, the time they spend over each page or their IP address.

Once personal data has been collected, privacy violations can occur for several

[1] `http://www.theguardian.com/world/the-nsa-files`

different reasons. For example, because the data minimization principle (as defined by the European Directive 96/46/EC) is not respected, hence the amount and sensitivity of data collected is not justified by the purpose it is collected for (e.g., the Social Security Number –SSN– is required to sign up for a mailing list service); another type of privacy breach occurs when data is collected without users' awareness (e.g., unwanted user tracking), and when data is revealed to unauthorized parties (e.g., as a result of a data breach caused by hackers or by malicious insiders). Data must also exclusively be used for the *purpose* specified at the time of collection, therefore using it for some other reason (e.g. for marketing rather than for research) constitutes a breach.

To prevent privacy violations from happening, several protection solutions have recently come to light. Some of these solutions help the users who are becoming increasingly aware of privacy issues [2] and wish to be granted anonymity while browsing the web[2] or to avoid online tracking[3,4]. Other solutions help organisations to maintain and improve their costumers' control over the data they exchange [3] or help them by recognizing and blocking actions that can lead to a data breach.

In the next section, we discuss the motivations at the basis of the contributions presented in this thesis. To help the discussion, we introduce the notion of a data cycle. A data cycle describes the path usually followed by users' personal data during and after the service provision. In this thesis we present a suite of privacy solutions which help to avoid privacy violations happening throughout the data cycle.

1.1 Motivations

Figure 1.1 illustrates what we call a *data cycle*. The data cycle shows one of the possible paths typically followed by personal data from the moment it leaves the individual's premises until it is stored in remote repositories. Risks of privacy breaches can arise at different points along the data cycle. In the figure we highlight the four points that are the focus of our attention, which are: i) the user, representing the individual to which the personal data refers to and who is also the one who is supposed to have the power to decide to whom and under which conditions her own data may be used; ii) websites on the world wide web offering services to users and relying on the use and exchange of personal data; iii) the web services (behind the websites), which are commonly used for exchanging personal data; and iv) the data repositories where personal data is stored, usually at the premises of service providers.

[2] `https://www.torproject.org/`
[3] `https://duckduckgo.com/`
[4] `https://adblockplus.org`

Figure 1.1: The Data Cycle

We want to stress that the data cycle presented here is just one of the simplest paths personal data may follow and therefore it is limited in its considerations. Nonetheless, it represents a typical path, and we will use it as reference throughout the thesis. As an example of how things can go differently, personal information can reach data repositories via other channels rather than the web, as in the case of health devices that directly communicate medical information (such as *blood pressure* or *glycemic index*) to the health provider repository. The discussions and the solutions reported in this thesis can be applied to this and most other scenarios as well.

User. The first place where privacy risks arise is at the start of the data cycle, when users release their data; this happens for instance when using a website for social networking, when banking online or shopping on the internet. At this stage, users could unintentionally leak their personal data by using websites that misuse their data e.g., by sharing it without the user's consent. To protect privacy at this stage, it is necessary to start by helping users in deciding whether or not they should share their data with a website at all. A first prerequisite towards fulfilling this goal, is to understand the process users follow to assess the trustworthiness of a website, that is how they decide whether to use a website or not.

Websites. Users have to choose whether or not to use a given website, and – in some cases – which website to use out of a set of competing ones. Ideally, the choice should also be based on the privacy guarantees offered by the different sites. Unfortunately, users neglect *en masse* the privacy policies of the websites they use, because they are complex and difficult to understand. To help users in making informed decisions it is important to provide them with a way to evaluate the privacy guarantees provided by the websites they intend to use. Some solutions exist already, provided in the form of browser plugins, which evaluate websites in terms of trust, security or privacy (e.g., WOT[5] and McAfee SiteAdvisor[6]). These systems show a warning in case of websites that have been poorly rated e.g., because they have been reported as hosting malware or perpetrating frauds. These solutions are based on blacklisting and reputation scores computed from community feedback. Unfortunately, these technologies can hardly keep up with the pace of a dynamic environment like the world wide web. This calls for solutions which dynamically evaluate websites, especially w.r.t. to privacy aspects.

Web Services. To provide content to the final users, website often make use of web services, namely software components available on the Internet and offering a given function. Web services are typically built using a service oriented architecture (SOA) where distinct functionalities are delegated to distinct web services combined together to accomplish a complex task. Usually, there is a main website acting as a web service orchestrator, coordinating the different affiliated web services in order to reach a common goal. For instance, a website acting as an online travel agency typically coordinates services for reserving a hotel room, for booking a flight, and for renting a car. Often, the same kind of service is offered by different providers, as in the case of different hotels allowing to reserve a room. Each of these providers needs a certain amount of personal data (e.g., identity card, passport, credit card) to offer its service. For example, while a provider may require only the e-mail address to book a hotel, another might also request the credit card details. In accordance to the data minimization principle, limiting the amount of data released diminishes the privacy risks. Existing literature offers several schemas for creating service compositions satisfying criteria such as minimizing the response time or maximizing the service availability. However, solutions which limit the risk of privacy breaches by e.g., minimizing data collection (e.g., w.r.t. to the amount and sensitivity of personal data) are still lacking.

[5] `https://www.mywot.com/`
[6] `https://www.siteadvisor.com/`

Data Repositories. After it has been used, data is not erased, but it is usually stored in data repositories, often owned by the service providers. Data in these repositories represents an important and delicate asset for any organization and it can be accessed by different actors for different purposes. For example, a provider could grant the access to its employees for after-sale purposes or to third parties for research purposes. At this stage of the data cycle, users' privacy is still at risk: not only there is little guarantee that the provider will not use the data for the wrong purpose, but hackers or malicious insiders could access the repository and steal valuable information. Despite a number of regulatory and technological steps that have been taken to give better privacy guarantees to citizen, data breaches and data misuses are not decreasing: according to the Open Security Foundation[7], over 502 million records were exposed in the first half of 2014. These records include credit card numbers, email addresses, log in credentials and other related personal information. Data breaches occurring at this stage do not only harm users' privacy but also organizations by damaging their reputation and decreasing customers' trust. In addition, organizations can be fined for lacking appropriate security measures when personal data is involved, or they can loose competitive advantage in case Intellectual Property (IP) is stolen. To minimise these risks, Access Control (AC) mechanisms [4] and tools for data leakage detection [5] are commonly used as defense against these threats. However, AC rules are often more relaxed than the privacy policy would prescribe; this in order to guarantee data availability which is critical in emergency situations (e.g. in healthcare domains), or because privacy policies are simply too complex to enforce automatically with an access control system. On top of this, most of the existing data leakage detection solutions are either not effective against unknown threats or too costly to use in terms of a high number of false positives. Developing effective solutions for detecting data leakage caused by known as well as unknown threats, is a problem that still has to be solved in order to protect users' privacy and businesses' critical assets.

1.2 Research Questions

The goal of this thesis is to study how to limit privacy risks that can arise at different points along the data cycle. We do this by zooming in on a specific crucial aspect at each of the steps in the data cycle. In particular, we aim at:

User step understanding how users behave online and how we can increase their risk awareness and support their trust decisions;

Website step evaluating websites with respect to the privacy protection they offer;

[7] `http://datalossdb.org`

Web Service step helping the user in selecting the web service composition which best matches user's privacy preferences; and

Data Repository step devise new ways to detect privacy infringements and data leakages which happen at data repositories.

We now translate these points into the research questions we are going to tackle.

Establishing a trust relationship with an online service is the first step towards the release of users' personal data. Understanding how users establish such trust relationships is a precondition to create tailored solutions which finally help users in their decision making process. This leads to the first of our research questions, which is of empirical nature.

RQ. 1 *What are the factors that a user takes into account before deciding to trust a website and what can be done to avoid misjudgments which could cause privacy losses?*

As we mentioned before, there exists several technological solution who help the user in avoiding questionable websites. These solutions mainly work by blacklisting websites that have been reported as abusive or where malware has been detected. A problem of this approach lies in the difficulty of keeping the blacklist up-to-date. One would rather be able to automatically and dynamically evaluate the privacy of a website. A primary source of information on how seriously a website takes privacy in consideration, is contained in its privacy policy. Unfortunately, privacy policies are long legal documents that users notoriously refuse to read. In this way a large source of valuable information is lost. To further investigate this issue we formulate the following research question:

RQ. 2 *Can we evaluate the privacy level of a website by automatically analysing its natural language privacy policy?*

By answering this question one would at least partly solve a problem *all* users have: that of understanding at a glance how protective (or dangerous) a 30-pages-long privacy policy is. Moving to SOA environments, we can observe that service selection is a key issue. While selection schemas that maximize properties as reliability, latency and response time are widely available, solutions which select service compositions on the basis of the level of privacy they offer are lacking. To deal with this problem we formulated the following research question:

RQ. 3 *How can we identify the service composition which best preserves privacy and best matches a user's preferences?*

Data collected during service provision, usually ends up being stored in large repository owned by the service provider. Data leakage, namely the unauthorized/unwanted transmission of data and information [6], is amongst the greatest privacy risk for data at this stage. Data leakage can be caused by malicious hackers who manage to gain access to the data repositories, or by insiders (careless or malevolent employees) who abuse their right to access sensitive information. If the repository contains sensitive information, e.g., banking or medical information, the privacy risk for end users as well as the costs for the organization storing the data is remarkable. To deal with the issue of data leakage, we formulated the following research question:

RQ. 4 *How can we monitor access to data repositories containing sensitive information in order to detect privacy infringements such as data leakages and misuses?*

In the next section we present the thesis roadmap, where we discuss where the answer to each research question can be found.

1.3 Thesis Roadmap

To answer the research questions introduced in the previous section, in this thesis we provide a suite of privacy protection solutions dealing with privacy risks at different stages along the data cycle.

Understanding the User. In Chapter 2 we address the first question and analyse users' behaviour in order to understand how users decide whether or not to trust a website. To this end, we define a General Trust Perception Model (GTPM) which captures the trust decision making process. The model suggests that users' perceived trust is influenced by user-specific characteristics such as her IT knowledge and her innate disposition to trust, and by the *perceived value* of certain factors of trust (e.g., the website's reliability, privacy and security level). The GTPM and the user study used to validate the model, are based on the contents published in [7] and in [8].

Evaluating Websites. In Chapter 3 and Chapter 4 we reason on how to evaluate websites privacy protection level by analyzing their privacy policies. We argue that, thanks to similarity in their structure and contents, it is possible to automate the analysis of natural language privacy policies. First, we propose a solution to assess a policy's *completeness level*, that is, the degree to which the privacy policy covers the categories extracted from privacy regulations. Policy completeness is assessed by means of a text classifier, based on a machine learning algorithm, which determines whether a text paragraph can be associated to any of the predefined categories. The

solution we propose provides also a structured way to browse a policy content. To prove the feasibility of our approach we test several automatic classifiers, obtained by applying machine learning algorithms to a corpus of pre-annotated privacy policies. The tests demonstrate that our approach has a (surprisingly) high accuracy. The results of this study have been published in [9] and [10] and are discussed in Chapter 3. Second, we present a solution which is able to extract semantic information out of a privacy policy and that evaluates a website on the basis of the amount and sensitivity of the personal data collected. Especially, we present an approach that is able to extract the list of personal data items collected by a service provider according to what it is stated in its privacy policy. In this way, we can immediately communicate to the users the impact of using a service in terms of the amount of personal data they have to reveal. To achieve this goals we use Information Extraction techniques. We validate the effectiveness of our approach over a set of pre-labeled privacy policies; the solution and its evaluation are presented in Chapter 4 which is based on the contents published in [10].

Selecting Web Services. In Chapter 5, we propose a solution for privacy-aware service composition. Service composition is a problem widely studied in literature: it deals with composing several sub-tasks which together solve a bigger problem. We propose a way to let service providers express their privacy policies and users define their privacy preferences; then we create the composition which best preserve privacy and best matches a user's privacy preferences. The solutions we propose has been published in [11] and in [12].

Monitoring Data Usage. In Chapter 6, we move our analysis towards the data repositories. We describe what kind of privacy infringements can happen at this stage and what can be done to detect such infringements. We propose a solution to detect data leakage by monitoring data activities over the data repositories. The solution we propose works by learning normal profiles of database usage, and by flagging any deviation form such profiles as an anomaly. We demonstrated our approach on an experimental dataset consisting of millions of real enterprise transactions. Experimental results prove that our solution is able to detect a wide range of data leakage attacks, and it significantly decreases the number of false positives w.r.t. state-of-the-art comparable approaches. This work has appeared in [13] while its extended journal version, as presented in this thesis, has been submitted for reviews.

Chapter 2

Understanding the User

In this chapter, we focus our attention on understanding how users decide whether or not to trust a website. When a user perceives a website as worthy of trust, she will feel comfortable in exchanging personal information if required to use its services. Misjudgments at this stage represent the very first risk for user privacy: sensitive information can be revealed to fraudulent actors which can abuse it to damage the user. To help the user to make the right decision it is important to understand, in the first place, what process users generally follow to assess the trustworthiness of a website. To this end, in this chapter we provide three main contributions: i) a General Trust Perception Model (GTPM) describing how users make trust decisions and which factors influence such decisions the most; ii) empirical and statistical evidence that users give different importance to different factors of trust according to the type of service offered (e.g., e-banking, e-commerce or e-health); and iii) evidence of the existence of a positive correlation between the user's information technology (IT) knowledge and the importance placed on factors such as security and privacy; this indicates that the role played by factors of trust such as security and privacy is more important for the users with higher IT knowledge.

2.1 Introduction

The number and types of e-services available on the Internet is continuously increasing. Governments and businesses have adopted e-service solutions to enhance citizens's access to information and services, or to improve production and boost economic growth. Nowadays, activities such as tax declaration, prescription renewal, or shopping can be easily accomplished online by using a dedicated website. While simplifying the execution of daily activities, the use of such websites also exposes the user to privacy risks such as identity theft. Users should be aware of the risks they face while using e-services and they should trust only those services that deserve to be trusted. To this end, trust decision support mechanisms should be provided to users. Clearly, to create such mechanisms, it is necessary to know how users make trust decisions, i.e. how do they decide whether to trust a website or not, and when a website is worthy of trust.

Trust is a complicated concept carrying many meanings [14]. The English Oxford dictionary defines trust as the "*firm belief in the reliability, truth, or ability of someone or something*". In sociology and psychology trust is seen as the concept that governs most of the *human-to-human relationships* [15, 16]. In business, trust is identified as one of the key factors for the development and maintaining of long-term cooperative relations [17]. In computer science, trust is essential to the semantic web [18], and the computational value of trust [19] is used to organize agents cooperation.

To deal with trust in the digital world, several attempts to adapt the concept of human-to-human trust to the online context have been made [20]. However, when the object of trust (*trustee*) changes in such a dramatic way (from a concrete person to an abstract object), also the models of trust have to change to adapt to the new context. For example, the signals of trust generally used to establish human-to-human relationships, such as physical aspects or body language, are missing when trust has to be established online. To understand online trust, we need to find analogous (or new) signals of trust and learn in which way they influence users' perceived trust. This understanding will allow the creation of tools to help users making their online trust decisions.

A large body of literature exploring the signals, or *factors*, affecting online trust exists [21]. The main problem of existing solutions is that they typically focus on very specific context. This leads to similar models with many of the same factors being rediscovered many times in different disciplines and contexts. For example, different trust models applicable to e-commerce, e-health, or e-banking have been proposed [22, 23, 24, 25, 26]. Although such models are referred to different application domains, they consider factors of trust that largely overlap.

In this chapter, we propose to unify the work carried on in different areas under a general trust perception model (GTPM) that can be applied to different domains.

The main idea is to have a set of factors of trust that encompasses those already presented in literature and to adapt the model to a specific domain by changing the weight of each factor. To show how one can determine such weight distributions we performed a user study in four important domains: e-commerce, e-banking, e-health and e-portfolio.

The GTPM that we propose describes how users make trust decisions before using a website. Note that the GTPM focuses on the first interaction between a user and a website, thus no direct past experience is assumed. We argue that trust decisions are made on the basis of the *perceived trust*, i.e. the degree to which a user perceives a websites as trustworthy. The perceived trust, in turn, depends on certain characteristics of the user (e.g., her disposition to trust) and on the user's perceived values for the website's factors of trust. Example of factors of trust are reputation, security, privacy, and brand name. The GTPM also addresses the topic of regret. Regret is the feeling that arises when a trust decision turns out to be wrong, e.g. the user is victim of a fraud after using a website. A trust decision can thus be considered good if it minimizes the future regret. To reason about regret, we introduce the following example.

Example 1 *Alice is considering to use an e-commerce site for the first time. After browsing the site for a while, Alice decides to trust it because it looks very professional and easy to use. She does not consider whether the site uses a security protocol, e.g. https during her payment, and she only looks for the presence of a privacy policy to "be sure" that her personal data is protected. Finally, Alice chooses a product, makes a payment, and after couple of days she successfully receives her purchase.*

In this case, Alice took a *positive* trust decision based on the *look&feel* of the website and on the presence of a privacy policy. Let us assume that, after the trust decision has been made, one of the following events takes place: i) the website is victim of a eavesdropping attack and Alice's credit card number is stolen together with a considerable amount of her money; and ii) Alice receives daily advertising emails since the website, as stated in its policy, shared her email with all its partners. In both cases Alice might regret her initial trust decision. In the first case, the trust decision could have been different if Alice would have given more importance to the security factor and looked for a more secure website (e.g. one using https where eavesdropping is more difficult to achieve). In the second case, the decision could have been different if the mere presence of a privacy policy would have not been seen as indicator of high privacy protection. These considerations lead us to argue that regret can be caused by the existence of:

- the *Factor Importance Gap*, namely the gap between *importance* as assigned by the user (does the user think *security* is important before the interaction?)

and its actual importance (considering the outcome of the transaction, how important did *security* actually turn out to be for the user?); and

- the *Trust Indicator Gap*, namely the gap between the *objective value* of a factor of trust (assuming it is quantifiable) and the value a user perceives via a trust indicator. A trust indicator is a visual way of representing the value of a factor of trust. For example, one or more yellow stars are a trust indicator for the factor *reputation*. Some indicators work better than others to communicate the actual value of a factor of trust. For example, a privacy policy as indicator for the *privacy* factor only shows whether a policy exists or does not exist. It cannot show how good the privacy policy is, hence it is not a good indicator to communicate the privacy value to a user. The difference between the actual value of a factor of trust and the perceived value is what we call Trust Indicator Gap.

To meet our general goal of minimizing regret, both gaps need to be reduced. In this chapter, we focus on the *Factor Importance Gap* which we believe has a strict connection with users' knowledge about Information Technology and security and privacy matters, while the analysis of the *Trust Indicator Gap* is left as future work. To study this gap, indeed, a study that creates different indicators for the same factor and verify the impact they have on the users should be carried out. We believe that, given a user, a trust factor and a website, different trust indicators lead the user to have different perceived value for that factor. For example, an icon which adopts the semaphore metaphor (e.g. red for websites with a poor privacy policy, yellow for websites with a medium privacy policy and green for websites with a good privacy policy) is more powerful to communicate the privacy level than a long text of legal terms such as a privacy policy. The validation of this belief, as well as the creation of other improved trust indicators that reduce the *Trust Indicator Gap*, requires studying their impact on users (for example by asking the users to give a numeric value to a factor when they are exposed to different indicators) Given the characteristics of the study, which we believe is closer to the Visualization and User Interface Design field, we left it out of our scope of the thesis. However, in Chapter 3 and Chapter 4 we give directions on the use of different visual indicators for the privacy factor.

The remainder of this chapter is organized as follows. In Section 2.2 we present the related work, in Section 2.3 we show the General Trust Perception Model and the way we obtained it, while in Section 2.4 we describe the user study we performed to test our hypotheses, its design and its validation process. In Section 2.5, 2.6, and 2.7 we discuss the results of the user study, while in Section 2.9 we provide the conclusions. The questionnaire we developed for our user study is fully described in Appendix A.

2.2 Related Work

The definition of the term *trust* has been widely addressed in the literature. Generally, trust can be seen as expectations [27, 28, 29], as vulnerability to the actions of others [30], as probability that a certain beneficial (or at least not harmful) action will be taken by the trustee [16], or as risk [31]. In addition, trust is usually seen as a multidimensional concept [30], i.e. it can be modeled by using a set of factors, and it is often referred as being *context-specific*, i.e. related to a given domain or situation [32].

Despite claims such as "people trust people, not technology" [20], the idea of extending the concept of trust to the digital world gained a lot of attention, leading to the development of computational [19], and human-to-computer [33, 34, 35, 36] models of trust. The human-to-computer models mainly aim at identifying the factors affecting trust, such as usability [35, 37], reliability [34], and security and privacy [37].

The explosive growth of e-services has increased the interest of researchers towards the study of trust between human and online transactional systems such as websites. According to the authors in [38], for such systems to be trusted, they need to be 'designed for trust', i.e. accounting for factors such as privacy, security and reliability from the very beginning. Corritore detected reputation, usability and risk as determinant factors of trust in websites [39]. McKnight, focusing on trust in e-commerce services, asserted that users go through two different stages before using an e-commerce service: an introductory stage, where they explore the website, and an exploratory stage, where they decide whether to make any transaction using it [40]. The quality of a website [41], such as the absence of presentation flaws and the user's knowledge [37], also seems to influence the trust and the willingness to buy in the e-commerce setting. Additionally, misconceptions about security and privacy might lead to a false perception of trust [42]. It is still not clear whether a deeper knowledge, e.g. on security mechanisms, increases [43] or decreases [44] the general trust in e-services. In this chapter, we provide evidence that deeper knowledge increases the impact that privacy and security factors have in determining the trust in a e-service. Studies on trust are generally based on a theoretical *trust perception model* (TPM), accounting for several factors - or antecedents - of trust. An in depth literature review on the antecedents of trust is available in [21]. The general hypothesis at the basis of a TPM is that such antecedents positively or negatively influence the perceived trust. This hypothesis is usually validated by means of a user study that experimentally determines which factors are actually taken into account before trusting. Every trust model applies to a specific domain and category of users. Trust models in the area of human-to-human [30], human-to-aid systems [35, 33], human-to-computer [34], and human-to-e-services [39] relationships can be found in the literature. Specifically to the area of human-to-e-services, trust perception models about e-commerce [45, 46, 47, 48, 49, 41, 50], e-banking [51, 24], and e-health [22, 23, 52] exist.

Table 2.1: The Trust Perception Models in the Literature

Reference	Risk	Reliability & Availability	Third Party Seals	Privacy	User's Knowledge	Security	Disposition to Trust	Quality and Look & Feel	Reputation	Brand Name	Usability
[33]	✓	✓								✓	
[30]	✓						✓			✓	
[35]											✓
[34]										✓	✓
[39]	✓		✓				✓	✓	✓		
[37]		✓		✓	✓	✓			✓		✓
[38]		✓		✓		✓					
[22]		✓					✓	✓		✓	✓
[23]					✓			✓			✓
[52]			✓					✓	✓	✓	
[47]	✓		✓		✓		✓	✓	✓		
[25]	✓	✓				✓	✓	✓		✓	✓
[41]								✓			
[53]			✓			✓					✓
[54]	✓			✓		✓	✓			✓	
[55]			✓					✓			
[50]			✓	✓	✓	✓	✓	✓	✓		
[24]	✓			✓		✓				✓	
[26]				✓		✓	✓				
[56]				✓		✓		✓	✓		
Rate %)	35	25	30	35	20	45	40	50	30	40	35

In Table 2.1 we give an overview of the existent TPMs: each row represents a TPM and each column one factor of trust; a tick indicates that the factor is part of the model and that its influence on perceived trust has been experimentally validated by the related work. Note that authors may use different terms to refer to the same factor (e.g. disposition to trust or dispositional trust): in the table we try to unify

this naming. The last row of the table indicates the percentage of analyzed TPMs accounting for the corresponding factor. Factors of trust can be of two types: user-related, if relative to trustor's characteristics (e.g. *user's knowledge* and *disposition to trust*) or website-related if indicative of a property of the trustee. Following we give a brief explanation of the meaning of each factor.

User related factors.

User's Knowledge (UK) refers to the expertise the user has w.r.t. the IT and Security field (e.g. his level of knowledge/skills about use of the computer and the web, or matters such as https, privacy policies, etc.).
Disposition to Trust (DT) is a characteristic related to the personality of the user, representing the user's general predisposition to trust the world [46].

Website related factors.

Risk expresses the degree to which the use of the website can bring loss or damages to the user.
Reliability & Availability represents how good the website is in performing its functions and keeping them available for users.
Third Party Seals refers to the presence of trusted third party logos and certificates on the pages of the website.
Privacy refers to the mechanisms used to protect users personal data.
Security refers to the set of mechanisms applied to secure user's transactions and data (e.g., use of https, encrypted storing, vulnerabilities assessment).
Quality and Look&Feel captures characteristics such as good design, attractive user interface, absence of syntactic and semantic errors.
Reputation refers to the way the website is seen and judged by people in general.
Brand Name says how well the brand behind the website is known. This factor encompasses other factors such as *competence*, *integrity* and *benevolence*, characteristics that are often associated to well known brands.
Usability refers to the ease of use of the functionalities provided by the website.

The results of the literature study presented in Table 2.1 show that existing TPMs share a significant number of factors of trust (e.g. half of the analyzed TPMs list *Quality and Look&Feel* as an important aspect). New models keep being developed [26, 56] and the factors overlap with existing models continues, justifying the need for a general model that could be easily adapted to different domains.

2.3 The General Trust Perception Model (GTPM)

In this section, we describe our GTPM, built by unifying the factors of trust used in existing solutions. Since the GTPM is not bound to any specific domain, it is possible to describe user's perceived trust in different context by simply changing the factors weight. The model also describes how users make trust decisions and gives hints on how such decisions can be driven to reduce regret. The GTPM provides the benefits of i) having a single model that can be adapted to different domains, and ii) unifying the work carried out in different areas of research. We will start by providing the definition of trust (and its related concepts) as intended in the remainder of this chapter in Section 2.3.1, while we will describe the GTPM in Section 2.3.2.

2.3.1 Definitions

The term trustor is usually referred to the person establishing a trust relationship. In this chapter, with the term **trustor** we refer to the user, while with the term **trustee** we refer to the website the user is considering to interact with. Note that a *website* can be a very complex object; similarly to [57] we restrict our scope to a specific *transaction* between the user and the website.

Trustworthiness is a property of the trustee indicating how worthy of trust the trustee is. Clearly, an object can be worthy of trust in a specific situation but not in another [58]. However, what changes is not the object (which remains the same) but the trustor that is evaluating the trustworthiness in that specific situation. Therefore we make a clear distinction between the trustworthiness and the (perceived) trust. **Trustworthiness** is a property of the trustee, seen as a vector of "various factors" of trust [59], while the **perceived trust** represents the trustworthiness in the eye of the trustor. Perceived trust depends on the importance the trustor gives to certain factors of trust, and on the trustor's personal characteristics, such as her general disposition to trust and her knowledge. Both trustworthiness and perceived trust depend on a set of factors of trust representing the dimensions of trustworthiness. In the following, with the term **factors of trust** we refer to the *website related factors* presented in Section 2.2. This list of factors can be either extended, if a new factor needs to be introduced, or reduced, by giving zero weight to a factor. We assume that each factor is quantifiable, i.e. it is possible to associate a numerical value to it. Although for some factors it is more natural to have nominal values, e.g. a label like *low*, *medium*, or *high* for the factor *usability*, we assume that some form of conversion to the numeric scale is possible. A **trust indicator** is a way of representing one or more factors of trust. For example the *https pad lock* is a trust indicator for the security factor, while *scores* (e.g. 95%) or *feedback* (e.g. 3 or 5 stars) could be indicators for reputation. Trust indicators help the user in associating a value to a factor of trust: such value can

be directly expressed by the indicator (e.g. 5 stars for reputation) or inferred by the user (e.g. the presence of the *https* lock means *high* security).

Based on the perceived trust, the user makes **trust decisions**, i.e. she chooses whether the perceived trust is enough to start a transaction with the trustee. Unfortunately, trust decisions are often made without spending enough time to gather all the data on trustworthiness [60]. To judge whether the right decision was made, the **a-posteriori trust**, i.e. the trust the user perceives after the transaction occurred, has to be taken into account. A-posteriori trust depends on events, such as frauds, that did (or did not) happen. In case the a-posteriori trust is different from the trust perceived before the transaction, the user can experience feelings of **regret**, i.e. a discrepancy between what she has done and what she feels she should have done [61, 62]. Regret can occur either because a positive trust decision (using the service) is betrayed by the trustee [63], or because a negative trust decision (declining the use of the service) is erroneous (e.g. the website not chosen was actually trustworthy and the benefit of using the service was missed by the user). Computational concepts of regret and 'Regret Management' are introduced in [63] and [64].

2.3.2 The model

Figure 2.1 describes our GTPM, which aims at capturing the trust decision making process, and at understanding how trust decisions can be driven to minimize regret. Let assume that Alice is our user and that, for each factor of trust, the website has a given value representing the *objective value* of that factor, e.g. how secure it is, what is its reputation, what is its level of privacy. It is important to notice there could be a gap between the *objective value* of a factor and the value as *perceived* by Alice. For example, the website may have a very *poor* management of users personal data (it sells them to third parties without anonymizing it), but Alice may perceive the privacy management as very *good* (she saw that a privacy policy is published on the website, and for her this is enough). We refer to the gap existing between the *objective value* and *perceived value* of a factor as the *Trust Indicator gap*. Accordingly, we define *objective* and *perceived* trustworthiness as follows.

Definition 1 (Objective and Perceived Trustworthiness – TW_{ws} and PTW^u_{ws}) *Given a website ws, a user u and a factor fac_i, we refer to the factor's objective value for ws as $a_i \in \mathbb{R}$ and to the corresponding user's perceived value as $a^u_i \in \mathbb{R}$. Let $FAC = (fac_1, fac_2, ..., fac_n)$ be the vector of factors of trust; we define a website's* objective trustworthiness *as a vector of objective values, one for each factor, $TW_{ws} = (a_1, a_2, ..., a_n)$. In addition, given a user u we define the* perceived trustworthiness *of u w.r.t. ws as the vector of perceived values, $PTW^u_{ws} = (a^u_1, a^u_2, ..., a^u_n)$.*

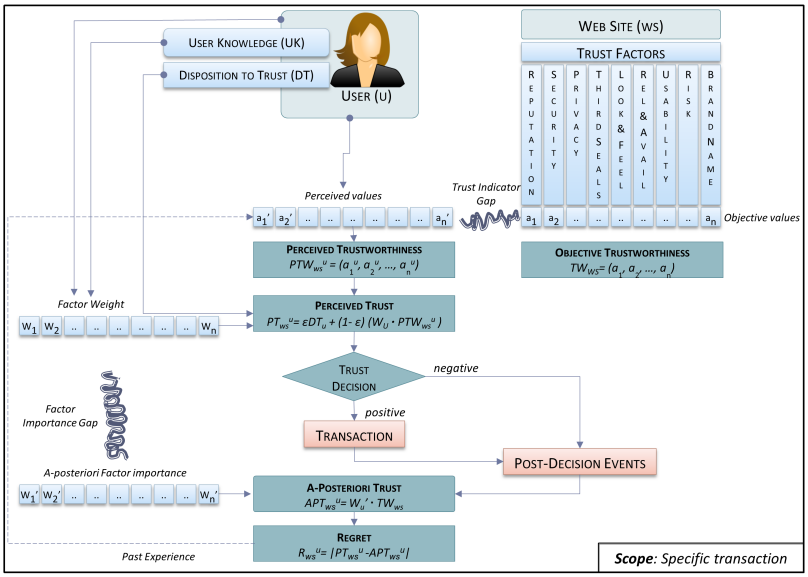


Figure 2.1: General Trust Perception Model (GTPM).

Given a user, we argue that her *perceived trust* depends on two components: i) the perceived trustworthiness; and ii) user's characteristics such as her personal disposition to trust, measurable with specific scales [15, 65], and the vector of factors weight given to each factor of trust. Note that we approximate the vector of factors weight (for the factors of trust discussed in Section 2.2 and for different types of websites) with the results of our user study presented in Section 2.5. Now, we can formalize perceived trust as follows:

Definition 2 (Perceived Trust – PT^u_{ws}) *Let ws be a website, u be a user, DT_u her disposition to trust, $FAC = (fac_1, fac_2, ..., fac_n)$ the vector of factors of trust and $W_u = (w_1, w_2, ..., w_n)$ s.t. $w_i \in \mathbb{R}$ the corresponding vector of factors weight. Given the perceived trustworthiness PTW^u_{ws}, we define the* perceived trust *as $PT^u_{ws} = \varepsilon DT_u + (1-\varepsilon)(W_u \bullet PTW^u_{ws})$ where $\varepsilon \in [0,1]$ is the relative importance of the two components.*

We argue that trust decisions are driven by perceived trust. The user decides whether the perceived trust is enough to lead to a positive trust decision or not. After a trust decision has been made, there are post-decision outcomes that can be positives (e.g. the product is delivered on time), or negatives (e.g. the product has been

delivered but the user realizes her email address is now available to advertising companies). These events enable the user to evaluate the *a-posteriori trust*, i.e. the level of trust the user has after obtaining more information about the *objective* trustworthiness of the website. Let us assume the post-decision outcomes enable the user to compute the objective trustworthiness of the website TW_{ws}, namely the user get to know the objective value a_i for each factor of trust. In addition, thanks to the post-decision outcomes, the user realizes that she did not give the right weight to some factors (e.g. privacy), and that in future she should give more importance to it (e.g. by carefully reading its privacy policy). This create a gap between the importance of a factor before (w_i) and after ($w_i^{`}$) a trust decision has been made. We refer to such gap as the *Factor Importance gap* and we define a-posteriori trust as follows.

Definition 3 (A-posteriori trust – APT^u_{ws}) *Let ws be a website, u be a user, $FAC = (fac_1, fac_2, ..., fac_n)$ be the vector of factors of trust and $W^{`}_u = (w^{`}_1, w^{`}_2, ..., w^{`}_n)$ s.t. $w^{`}_i \in \mathbb{R}$ be the vector of a-posteriori factors weight, namely the weight the user gives to each factor of trust* after *a trust decision has been made. In addition, let be TW_{ws} the objective trustworthiness of the website that the user got to know thanks to the post-decision outcomes. We define the* a-posteriori trust *as the product of the a-posteriori factors weight and the objective trustworthiness, $APT^u_{ws} = W^{`}_u \bullet TW_{ws}$.*

Note that in the a-posteriori trust the component relative to the user's disposition to trust DT_u is not accounted for, since it is only meaningful before interactions with an object of trust take place [46]. Regret can be seen as the difference between user's expectations before using a website, and user's considerations after using a website, hence it is defined as follows:

Definition 4 (Regret – R_u) *Let ws be a website, u be a user, PT^u_{ws} be the user's perceived trust and APT^u_{ws} the user's a-posteriori trust. We define regret as distance between the perceived trust and the a-posteriori trust, $R^u_{ws} = |PT^u_{ws} - APT^u_{ws}|$.*

To minimize regret we need to make the user perceived trust resemble the a-posteriori trust as close as possible, i.e. we need to reduce both the *Trust Indicator gap* and the *Factor Importance gap*.

The *Trust Indicator gap* can be reduced by diminishing the difference between a_i and a^u_i, i.e. trust indicators should be able to communicate to the user the objective value of a factor of trust. Let us consider the *https pad lock* as trust indicator for the *security* factor. This indicator just says that the transaction channel is encrypted, meaning that, e.g. a man in the middle attack cannot be easily performed. However, the pad lock does not say that an attacker cannot enter the database of the website and access users' personal data (if those are not well protected) or that the certificate has not been compromised (e.g. the DigiNotar case[1]). A user that gives high importance

[1] http://www.itpro.co.uk/635833/certificate-authority-confirms-hack-after-gmail-attack

to security, but evaluates it by the mere presence/absence of the padlock, can still regret hes decision if her data is stolen. In this case the main problem is that the padlock is insufficient as trust indicator for security since it does not communicate the security objective value. This is a clear case of *Trust Indicator gap*.

On the other hand, let us assume there is a way to precisely measure the level of security offered by a website, e.g. according to the results of penetration testing, and that a new indicator with such *security level* is shown to the user. If the user does not give high importance to security ($w_{security}$ is very low) she will use the website even though it has poor security. If something goes wrong, e.g. her credit card is stolen, the user will still experience regret: in this case the problem is due to the *Factor Importance* gap, i.e. the trustor gives a level of importance to a factor that is not confirmed by the post-decision outcomes.

Since the understanding of the *Factor Importance gap* is the main focus of this chapter, we developed the user study presented in Section 2.4. With this study we want to understand which factors have the higher impact on trust decisions. Also, the study aims at discovering whether it is possible to manipulate the *Factor Importance* by acting on the user's knowledge, e.g. if the higher importance of the security factor corresponds to the higher knowledge (better understanding of security risks).

Understanding what happens after the trust decision is made, and in which situations users experience regret, requires a different user study, with the goal of understanding whether transactions and external events influence a-posteriori trust and a-posteriori factor importance. This study is part of our future work as discussed in Chapter 7.

2.4 The User Study

According to our GTPM, to reach the overall goal of minimizing regret, it is necessary to reduce both the *Trust Indicator gap* and the *Factor Importance gap*. Since the focus of this chapter is on the *Factor Importance gap*, we carried out a user study with the main objectives of: i) understanding how the importance of a factor changes in four different domains namely e-health, e-commerce, e-banking and e-portfolio; and ii) understanding whether the factor's importance can be manipulated by acting on the user's knowledge. Understanding the importance of factors in the domains we analyzed provides interesting insights on which signals are important for the user to trust. This knowledge can help in the development of trust indicators that are more effective. For example, if the study reveals that users give poor importance to security, new security indicators (e.g. more visible, more attractive and effective) can be developed. On the other hand, if a relationship between users knowledge and factors importance exists, this could help in reducing the *Factor Importance gap*:

users do not need to have a bad experience before realizing they do not give the right importance to some factors (e.g. security and privacy) since it is sufficient to act on their knowledge. Thus, we carried on a user study which aims at testing the following hypotheses:

Hypothesis 1 *The importance the same user gives to a factor of trust is different for different application domains.*

Hypothesis 2 *The importance of a factor of trust is correlated to the user's knowledge in the IT security field.*

Hypothesis 3 *The importance associated to privacy and security factors increases if users are trained and informed about privacy and security topics.*

The user study is carried out by means of a questionnaire which is fully described in Appendix A. The questionnaire is composed of ten questions, divided in three sections: one to gather demographic information about the respondents, one to evaluate their knowledge, and the last to measure the weight respondents associate to each factor of trust in different domains. Measuring the factors weight distribution in other domains can be done by adapting question number 9 to reflect those domains.

The validity of the questionnaire has been confirmed by applying the content validity method [66]. We asked a panel of experts from the TAS3 project Consortium[2] to review and rate each item (question) of the survey. The items rated as relevant remained untouched, while the others were deleted or adjusted according to the reviewers' feedback. A pilot study, monitoring five respondents while answering the questionnaire, was also performed. This helped in rephrasing unclear questions, verifying and eliminating the presence of bias, and adding details to terms seen as vague (e.g. quantifying *very often* to mean *at least once a month*). In the remainder of this section, we present the questionnaire, and we discuss the three different parts it is composed of, namely the sample frame, the user's knowledge, and the factors importance.

1. **The Sample Frame** - The population of our survey is represented by users of e-services such as e-banking, e-commerce, e-portfolio and e-health. The "Digital Report 2009" [67] reveals that 89% of Dutch internet users, aged 25-44, use e-banking services; that the typical on-line shopper is high-educated, aged 25-44; that 19% (aged 12-74) use the Internet to look for a job; and that 30%, aged 55-64, surf the web to look for health-related information. People with these characteristics can be considered samples of our population. The first part of the questionnaire contains questions about gender, age, educational

[2] `http://vds1628.sivit.org/tas3/`

level, and job position of the respondents, to verify whether the respondents group reflects the sampling frame of our study.

2. **The User's Knowledge (UK)** - Questions in this part of the questionnaire aim at capturing the user's knowledge in the IT Security field. Users are asked to judge their own knowledge and ability on IT security-related topics, such as computer and internet usage, privacy policies, and https. A four-item scale is used for the answers: *no knowledge*, *limited knowledge*, *good knowledge*, and *expert knowledge*.

3. **Factors Importance** - Another part of the questionnaire is the one aiming at measuring the weight users give to each factor of trust. This is done by verifying how much attention users dedicate to each of them. Respondents were presented with a website usage scenario in each of the settings (e-banking, e-commerce, e-portfolio and e-health) and asked to answer questions assuming it was the first time they used the specific service. Questions were formulated in such a way as to verify the influence each factor has on perceived trust. To test the importance of the factor *Look&Feel*, for example, we ask the user whether the design aspects of the website (e.g. attractive colors, or professional icons) influence his trust in it. Let Q_f be the set of questions used to measure the weight of the factor f. For each question $q \in Q_f$, respondents were allowed to choose amongst four optional (ordinal) answers : *never*, *almost never*, *very often*, and *always*. The value of the answer to the question q is denoted as v_q. To compute the weight w_i of each factor of trust, we averaged the answers given to each question q in Q_f.

Note that the statistics results we discuss in the following sections have been obtained by transforming the ordinal responses provided by the respondents into numeric values using optimal-scaling. Optimal-scaling is a widely-adopted data-driven technique that binds numerical values to ordinal ones by maximizing the correlation between variables. More on this topic can be found in [68, 69].

With the same questionnaire we carried out three different studies, the results of which are described in the remainder of this chapter. The contributions of the study are as follows:

- **First Study**: shows the results we obtained by asking randomly selected students and employees from the TU/e to fill-in the questionnaire. This study, with a relatively large sample frame (335 respondents), is discussed in Section 2.5. These results extend earlier work presented in [8].

- **Second Study**: shows the results of a reliability test, proving the questionnaire we built is reliable, i.e. it gives similar results with a similar sample (but

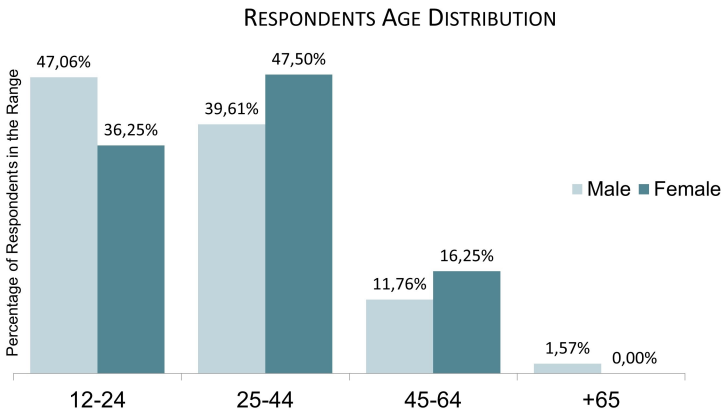


Figure 2.2: Respondents Age Distribution, by Gender.

different respondents) as described in Section 2.6.

- **Third Study**: shows the difference in users' responses when users are trained on the importance of security and privacy, and when they actually use a service (in the first study he service is only described). Results and discussion of this study are provided in Section 2.7.

2.5 First Study: Measuring Factors Importance

For this experiment a web interview methodology was used to ask subjects to take part in our research. The subjects were selected amongst employees and students of the Eindhoven University Technology (TU/e). This sample frame cover the spectrum of knowledge since both people with low and high IT Security expertise, are part of the sample. The TU/e has about 6000 students and 4000 employees. To obtain enough responses for statistically significant results and account for lost e-mails and uncooperative subjects, about 1600 e-mails were sent to e-mail addresses selected from the TU/e internal mailing list. During the sample selection the percentage of students (about 60% of the whole) and staff (about 40% of the whole) has been maintained. The subjects selected by our pseudo-random procedure received an e-mail and a reminder one week later, explaining the scope of our research and inviting them to participate by filling the on-line questionnaire. In the e-mail, they were also informed of the anonymous nature of the questionnaire.

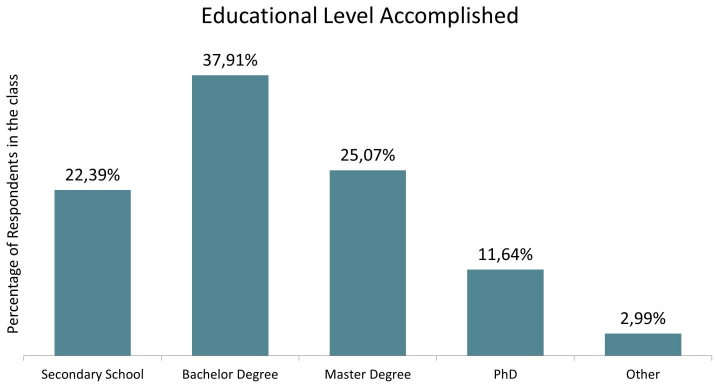


Figure 2.3: Respondents Education Level.

A total of 335 valid responses were collected. Incomplete responses were ignored. As the percentage of those was below the 20% this does not affect our final results [70]. To check the presence of CMV (Common Method Variance), i.e. the "variance attributable to the measurement method rather than to the constructs the measures represents"([71]), Harmans one-factor test was conducted. No single factor with covariance bigger than 50% emerged from the test, indicating that CMV does not constitute a problem for our study.

The respondents group is composed of 76% male and 24% female. Although this is representative for the population at the TU/e, this gender skew might represent a problem for the generalization of the results that should be taken into account. The age distribution, divided for gender, is presented in Figure 2.2: the graph shows the percentage of the males component (respectively females) falling in each of the age categories. In each category men and women are almost equally represented, although there is a lack of respondents aged 65 or older. This is mostly due to the community sampled and to the fact that retirees people are not included. Figure 2.3 shows the education level of our respondents, that is formed for 60% of students and 40% of employees, matching the sample frame.

2.5.1 Factors Weight

The weight of a factor of trust in different settings is presented in Figure 2.4 with a comparative graph. Note that, to improve readability, in the results we discuss in the remainder of the chapter, the factors weight has not been normalized in any way. The complementary analysis with normalized values is presented in [8] and it does not

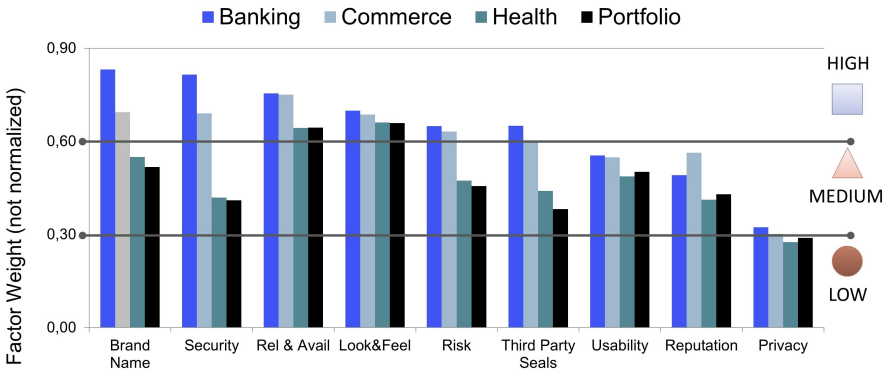


Figure 2.4: Factors Weight Distribution in the Different Domains.

show relevant changes in the factors ranking. The graph is divided into three bands of equal width: *low*, *medium* and *high*. We can notice that the factors weight distribution is similar between e-commerce and e-banking, and between e-portfolio and e-health. This intuition is confirmed by the Mann-Whitney test. The logic behind the Mann-Whitney test is to rank the data for each scenario and see how different the two ranks are. If there is a systematic difference between the two scenarios, then we can say that the ranks will be quite different otherwise we can say the ranks are similar. In our experiment this test suggests that the distribution of the factors weight is not the same across the two groups of scenarios (e-commerce and e-banking versus e-portfolio and e-health). These results confirm Hyp. 1; for e-banking and e-commerce the importance of factors is significantly different from the importance for e-health and e-portfolio.

If we consider the weight each factor has in different domains we can see that the *reliability&availability* and *look&feel* are the only factors in the high band for every scenario, suggesting that frequent error messages and crashes influence the perceived trust. Other highly important factors are *brand name* and *security*, followed by *risk* and *third party seals*, but only for e-banking and e-commerce. The high importance of *brand name* in e-banking seems to support the idea that trust in traditional (offline) banking influences trust in the online banking [51].

The weight given to *security*, *risk* and *third party seals* significantly drops in the e-health and e-portfolio scenarios. Given the sensitiveness of the data collected in such domains, it is a strong recommendation to stress the importance of such factors to the users, e.g. by improving impact of the relative indicators. The motivation

behind the drop of the weight given to the *risk* factor may be due to the fact that in these scenarios the risk (in terms of money loss) is less tangible than in the others. Also, note that e-health and e-portfolio are still not well known services: 15.8% of our respondents had previously used, at least one time, e-health services, while e-portfolio services had been used by only 1.5% of our sample (versus the 96.7% of e-banking and 71.9% of e-commerce). This may also explain why *third party seals* are not important: users probably do not know what kind of seals to expect from these new upcoming services.

The factors *usability*, *reputation* and *privacy* show no considerable differences amongst scenarios; only the reputation is slightly higher in e-commerce probably because it is the domain with the highest presence of automatic reputation systems (e.g. eBay). For the other scenarios it seems that users do not really look for external opinions: 59% of them said they never (or almost never) verify the reputation of an e-banking and 45% does not bother to ask their friends about what kind of experience they had with it. This can be explained by the fact that users do not like to manually collect and evaluate feedback on their own, but not necessary that they would not like to use automatic systems providing them with the reputation of the service (a fact confirmed by the success of reputation systems such as Tripadvisor[3]). Another reason for the low importance given to *reputation* in e-banking can be that users already know the brand and base their on-line trust on this, so they do not have the need to check *reputation*.

The importance of the factor *privacy* is the lowest in each scenario. Note that we measured the weight of privacy by asking whether or not users read the privacy policy stated by a website, i.e. privacy policy has been considered as the privacy indicator. We think the low weight of privacy in this case is a typical example of *Trust Indicator gap*: the fact that few respondents are interested in reading the privacy policy of a website does not necessarily means users are not interested in their privacy but, most probably, that privacy policies, as main current *privacy indicators*, are not enough to capture the real privacy value.

The main take home message of these results is that *reliability&availability* and *look&feel* are the factors that impact trust perception the most in all the domains we considered. To avoid that fraudulent and malicious website manage to be trusted by end users by "appearing" trustworthy, the impact that other factors, such as security and privacy, have on trust perception has to be improved. We believe this can happen by i) educating end-users, e.g. by increasing their knowledge in security risks; and ii) improving trust indicators. The first of the two hypothesis is tested in the following study.

[3] `http://www.tripadvisor.com/`

2.5.2 User's Knowledge

To verify hypothesis 2 it is necessary to first measure the user's knowledge, and check whether there is a correlation with the weight given to specific factors. To measure the user's knowledge (UK) we asked the users to answer questions related to their expertise on IT Security topics, as explained in Section 2.4. Cronbach's alpha test was conducted to calculate the reliability of the scale. The item-total correlation is above 0.5 for each item (values above 0.3 are acceptable) and the Cronbach's alpha value, for all the items we used to build UK, is 0.9, proving the reliability of our scale. UK is the average of the value associated to each given answer. The minimum registered value for UK (ranging in [0,1]) is 0.07, while the maximum is 0.87.

The graph in Figure 2.5 shows the different factors on the x axis, and the correspondent weight on the y axis. For each factor we clustered the responses according to the users knowledge (from the lowest to the highest level of knowledge). The graph intends to show how to an increase of knowledge level corresponds an increase of weight for the factors *brand name*, *security*, *risk* and *privacy*. This graph refers to the e-commerce domain, but the trend is similar in the other domains. This is proved by the correlation analysis between the weight of a factor (dependent variable) and the user's knowledge (independent variable) as shown in Table 2.2. To carry out the correlation analysis we used Spearman's non-parametric correlation test (this test is used when the normal distribution assumption is not verified as in our case). The Spearman's test measures the statistical dependency of two variables by computing a correlation factor (shown in Table 2.2). The sign of the correlation factor indicates the direction of the dependency while the p-value indicates the level of significance (the lowest the p, the more significant is the correlation). The results in Table 2.2 show that there is a significant positive correlation between the user's knowledge and the weight of the factors *security* and *privacy* in all the domains. This suggests that a user with higher IT security knowledge will give more importance to security and privacy while making trust decisions. The positive correlation also exists for *brand name* and *risk* in the e-commerce and e-banking scenario.

2.6 Second Study: Reliability of the Questionnaire

Ideally, each survey question should be clear, unambiguous, mean the same thing to everyone and provide a reliable measurement (of the importance of a factor, the knowledge of a user, etc.). Reliability of measurement, i.e. the extent to which repeatedly measuring the same property produces the same results, can be validated by repeating the same measurement on the same subject at different moments in time. The time between measurements should not be too small to prevent the measurement

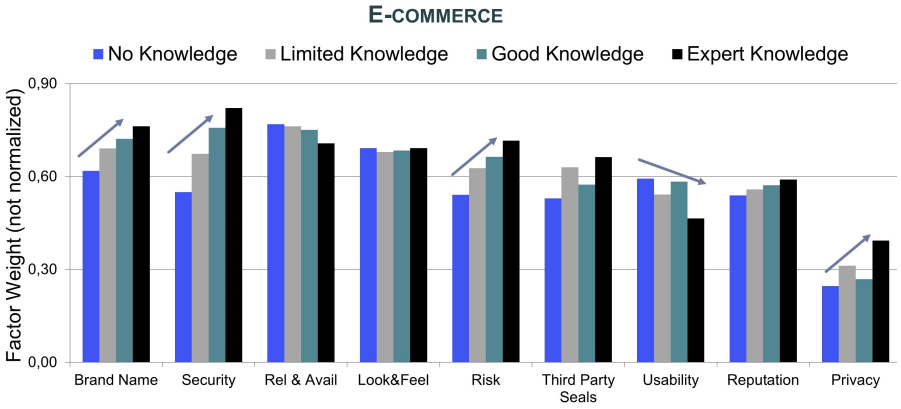


Figure 2.5: Factors Weight in the e-Commerce Domain, by Class of Knowledge.

Table 2.2: Spearman's Correlation Test

	e-banking	e-commerce	e-health	e-portfolio
Security	0.292**	0.278**	0.150**	0.219**
Privacy	0.142**	0.128*	0.183**	0.221*
Brand Name	0.218**	0.146**	0.061	0.097
Risk	0.171**	0.192**	0.045	0.102
Third Party Seals	0.154**	0.102	0.032	0.074
Usability	-0.091	-0.119**	-0.064	-0.076
Reputation	-0.015	0.088	0.099	0.080
Rel. & Avail.	-0.026	-0.092	-0.067	-0.099
Look & Feel	-0.009	-0.013	-0.009	-0.019

* = Correlation is significant with $p < 0.05$
** = Correlation is significant with $p < 0.01$

being influenced by memory, and not too large to avoid changes in the measured quantity which may change over time. Since the respondents in our settings are anonymous, it is difficult to carry out a reliability test in this way. To avoid this problem, but still provide an indication of the reliability of the questionnaire, we performed the test by asking a small group of new respondents to fill in the questionnaire, and by comparing the results we obtained in this way with the ones of the on-line survey.

The new group consisted of 12 Information Technology experts, gathered in a panel for the evaluation of the TAS3 project. Since the panel members were selected based on their high expertise in IT and security, they all fall in the class of *expert* knowledge. We compared the answers of these certified experts with the answers given by the users classified as *expert* in the online study. Figure 2.6 reports the results of the comparison, based on the 7 valid responses collected from the expert panel. For each scenario we present the difference amongst the weight given to the factors by the two groups. The factors with a red circle are the ones for which no statistically significant similarity between the responses on the two groups was found (based on the Mann-Whitney test for independent samples).

Most of the factors show similarity, i.e. there is no significant difference in mean scores. Exceptions are *look&feel* in the e-banking and *third party seals* in both e-banking and e-commerce, to which the panel experts give considerably less importance and *reliability and availability* in e-health to which the experts give considerably higher importance. The results also suggest that the TAS3 experts gave approximately the same answers for all the domains and that their variance is far less than the general public. Finally the panel members give a high importance to *security* in all setting. As the panel members are all IT experts facing security problems on a daily basis, security cannot be less than a top priority to them. Moreover, since the experts from the panel will have even more expertise than the average *expert* class user from the survey, this higher importance of security confirms the trend expressed in Table 2.2 (higher knowledge corresponds to higher importance of security). In all, we can conclude that this study supports the questionnaire as a reliable measurement tool.

2.7 Third Study: The impact of User Coaching

The results presented in Section 2.5 give us two main conclusions: i) the importance of trust factors is different in different domains, and ii) higher user's knowledge corresponds to higher importance associated to *security* and *privacy*. In the study described in this section, we demonstrate that if users are coached about the importance of security and privacy, then the weight of those factors will increase. To verify this, we asked to a new group of respondents to fill-in the questionnaire after they have been trained and they have used a specific website. While in the online survey respondents were only provided with a description of the scenario, in this setting we ask them to use a website (specifically offering e-portfolio services) before completing the survey. The respondents were recruited during a series of workshops which have been made in the context of the TAS3 project. During the workshop the participants were coached, guided, and monitored during their activities. At the end of the live session

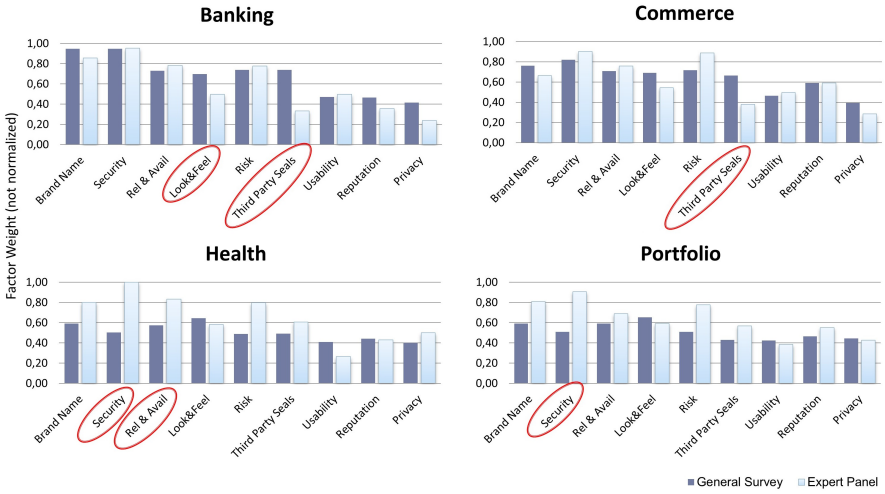


Figure 2.6: Factors Weight Comparison: Survey versus Expert Panel.

(they were asked to perform some tasks on the website) they were provided with the questionnaire which was slightly modified to fit the context. An additional question was also added to the questionnaire giving the following main changes:

- The questions regarding the factor importance are only related to the e-portfolio scenario;
- The questions are rephrased to reflect the fact that respondents fill in the questionnaire after experiencing a live session of an e-portfolio website ('*did you*' instead of '*would you*');
- A new question for the privacy factor has been added. Since we believe the question used before ("Would you read the privacy policy stated by the website?") does not assess the importance of privacy but the validity of privacy policies as privacy indicators, we rephrased it by asking whether "the way the website manages your personal data influence your trust in it". The previous question is maintained so we can compare the weight of privacy in the two cases.

A total number of 17 completed surveys were collected during the workshops. Although this number of responses is not enough to provide statistically relevant results, they can be used to confirm or deny the results and our intuitions coming from

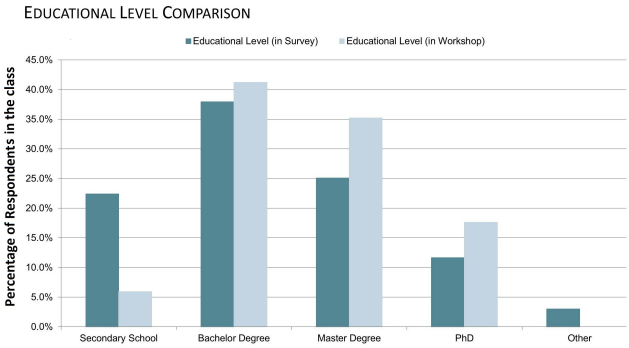


Figure 2.7: Educational Level Comparison: Survey versus Workshop.

the analysis of the on-line survey. Figure 2.7 and Figure 2.8 show the differences between the participants to the survey and the respondents to the workshops. Note that the first group of participants are on average older (the workshop focused on graduated professionals while the survey population included many students), and with a higher educational level (50% of the workshops' respondents has at least a Master degree). The respondents frame consist for 75% of employees, and for 25% of employers. Applying the Mann-Whitney test for independent samples does not show any relevant difference between the average knowledge in the two groups. However, as shown in Figure 2.9, the factors weight changes in a significant way for *security*, *reputation*, *usability* and *reliability and availability* – the red circle in the figure means the average is statistically significantly different according to the Mann-Whitney test. The fact that *security* and *reputation* are more important is likely due to the training that the participants received; special focus is given during the whole workshop to the secure management of personal data and a reputation system is in place in the demo service. During the development of the e-portfolio service a large effort was made to make it easy to use, which may contribute to the higher importance given to the *usability* factor. The others factors, when compared to the results obtained with the survey, show only small variation, confirming the results of the survey and indicating that these results are generalizable to other populations.

Another relevant result obtained during this study is the one presented in Figure 2.10: here we compare the importance associated to the privacy factor, when computed in two different ways: i) as the importance given to privacy policies, or ii) as the importance associated with the way personal data is managed. The results clearly confirm our intuition: the importance of privacy measured in the survey was consistently lower, not because users do not consider privacy important, but because

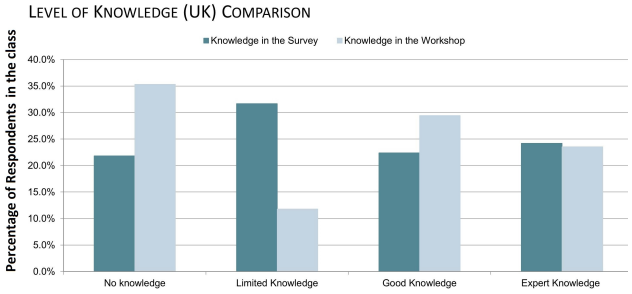


Figure 2.8: Knowledge Level Comparison: Survey versus Workshop.

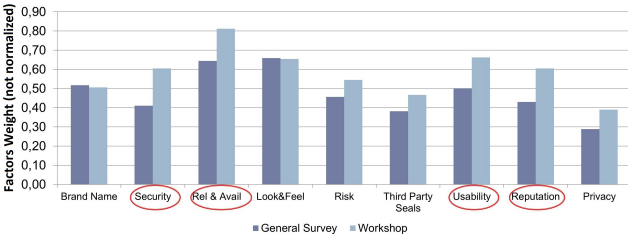


Figure 2.9: Factors Weight Comparison: Survey versus Workshop.

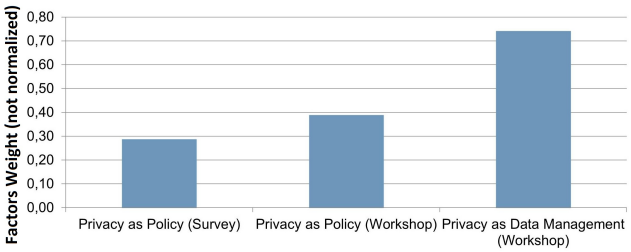


Figure 2.10: Privacy Factor's Weight for Different Questions.

they do not consider it worthwhile to read privacy policies. The main message of this result is that users do care about privacy (the privacy factor is the second most important factor if its weight is computed using the second question), but privacy policies are not the right trust indicator.

2.8 Limitations

In this section we discuss the limitations of the research presented in this chapter. In our GTPM we assume the *past experience* of a user does not affect the very first interaction with a website but it only impacts future trust decisions with the same website. This is not entirely true since the disposition to trust of a user may be affected by the outcomes of her current and previous transactions, with current and previously used websites. Hence, this assumption constitutes a limitation of our approach and extending the model by including past experience should be considered. Another limitation can be found in the reliability test discussed in Section 2.6, where we used 12 IT experts to carry out the test. We have to note that such a small sample might not be representative of our population. The small size of the sample is due to the difficulties we had in getting participants from other university or organisation. To overcome this limitation it is necessary to repeat the study over a larger sample of respondents.

2.9 Conclusions

In this chapter we presented a general trust perception model (GTPM) describing the way users make trust decisions and what can be done to help the user in making trust decisions that minimize regret. We suggest that to minimize regret both the *Trust Indicators gap* and the *Factor Importance gap* need to be reduced. With our user study we demonstrate that factors of trust have different importance in different scenarios, and that there is a positive correlation between user's knowledge and the weight given to *security* and *privacy*. Thus the *Factor Importance gap* can be reduced by educating the user on the importance security and privacy have in making trust decisions. We have also looked at the *Trust Indicators gap* and we suggest that improving the way trust indicators communicate the value of trustworthiness can reduce such gaps. This can be especially true for the privacy factor as our study shows that users do care about privacy as "data management policies" but they do not read privacy policies which show to be not effective as trust indicator tool. This is exactly the topic of the next two chapters of the thesis in which we present solutions for improving the privacy indicators.

Chapter 3

Evaluating Websites: Privacy Policy Completeness

The results of our users' behavior analysis presented in Chapter 2 point out that factors such as privacy and security do not play a key role when it comes to making online trust decisions. On the other hand, we also uncovered that, if trained and made aware of the risks they face, users do care more about privacy. However, the privacy indicators they are provided with, such as privacy policies, have shown to be not effective to guide users' decisions. This is due to the fact that policies are very long documents difficult to understand. In this chapter, we present a solution to assist the user in making informed decisions by automatically analyzing natural language policies to assess their quality in terms of completeness (i.e. the degree of coverage of privacy categories extracted from privacy regulations) and to provide a structured way to browse the policy content.

3.1 Introduction

Websites are obliged by law to publish privacy policies in which they state how personal data will be handled. As a key communication channel to the user, privacy policies should provide complete information on the way personal data is collected, used, stored or shared. However, this information is often 'hidden' in free text full of technical terms, that users typically refuse to read [8, 72]. This leads to situations

in which users unconditionally accept the privacy policy (a step often necessary to use a website or an application), with no clue of what conditions they have agreed to, e.g. whether the service provider will share their data with third parties or whether the privacy policy contains information about sharing practices at all. To allow users to truly make informed decisions, the usability of privacy policies needs to be improved.

To this end, we propose a system that automatically evaluates the level of completeness of a privacy policy. The privacy policy completeness is defined as the degree of coverage of privacy categories. Privacy categories are defined according to privacy regulations, such as the EU 95/46/EC and the EU 2006/24/EC, and guidelines as those provided by the OECD (Organization for Economic Cooperation and Development). Examples of privacy categories include *Data Collection*, *Data Sharing* and *Retention Time*. A high level of completeness of a privacy policy alleviates the (user's) concerns on data disclosure [73], and forms a better tool for communication. By measuring the level of completeness, we give the user information on whether the privacy categories of her interest are covered or not by a policy.

To measure completeness, we use text categorization and machine learning techniques to check which paragraphs of the natural language privacy policy belong to which category. The completeness level (or grade) is then computed according to which of the categories important for the user are covered by the policy. The user can then inspect the policy in a structured way, selecting to look at only those paragraphs belonging to the categories she is interested in.

Providing initial information on whether or not important categories are covered by a policy can help the user in making initial assumption on how the website takes into account her (vision) of privacy. Moreover, by being able to detect which paragraph of a policy is related to which privacy category, we open the way for the application of more sophisticated natural language techniques (e.g. Information Extraction) that can be applied to explore the semantic value of a policy.

The remaining part of the chapter is organized as follows: in Section 3.2 we analyze the related work, and the position of our contribution. In Section 3.3 we discuss the goals, the methodology and the implementation of the completeness analyzer, while in Section 3.4 we present the results regarding the accuracy of our approach. Finally, in Section 3.6 we discuss the conclusions.

3.2 Related Work

The difficulty users have in understanding privacy policies, often resulting from complex and ambiguous language, has been observed in several studies [74, 75]. Existing approaches which aim at improving privacy protection by the means of privacy policies can be distinguished according to which of the two actors – the service provider

(website) or the user– is the target of the solution.

Approaches such as those described in [76, 77], in [78, 79] and in [80] aim at facilitating the service providers in authoring privacy policies. Especially, SPARCLE [76, 77] is a framework that intends to assist an organization in the writing, auditing and enforcement of privacy policies. Its main goal is to help organizations to create understandable privacy policies, compliant with privacy principles. The framework takes privacy policies written in constrained natural language (CNL), checks them for compliance with privacy principles, and translates them into a machine readable and enforceable format, e.g. EPAL [81] or XACML [82]. The use of specific patterns in the sentences and CNL, i.e. a subset of a natural language with a restricted grammar and/or lexicon [83], makes it possible to parse such policies. A similar approach is also used in [80], where privacy authors can type (CNL) English sentences, and a machine processable policy is automatically generated from the input text.

PPMLP (Privacy Policy Modeling Language Processor) [78, 79], as SPARCLE, aims to help organizations in generating privacy policies, making such policies compliant with the privacy principle extracted from the Australian National Privacy Principles. The privacy policy authors specify a meta-privacy policy, that is then translated in a template of rules for the enforcement. Such a meta-policy is then analyzed by the system that suggests new rules, to allow the compliance with the privacy principles. Once the meta-policy is ready, it is translated both in EPAL, used for its enforcement, and in natural language (using static matching rules), for the presentation to the user. Within the system a PPC (Privacy Policy Checker) is used to check whether the policy is enforced: the PPC is plugged into the website, and has access to the data the application stores about a user. In this context the PPC is trusted by both the user and the website. When the user performs a transaction, the PPC analyzes it, to verify that the policy is enforced, and assigns a compliance rate to the website. Such grade is based on whether the enforcement of the policy takes place, and on the weight the user gives to each of the different privacy principles. However, the PPC is not able to check compliance if personal data is subsequently transferred to another site, especially if the user is no longer connected.

The availability of machine-readable privacy policy makes it easier to create automatic tools to evaluate the quality of a policy. However, the applicability of such tools strongly depends on the adoption rate of structured languages to express privacy policies. Solutions such as P3P (Platform for Privacy Preferences) [3] deal with this problem by proposing both a structured language for expressing policies, and a tool for matching policies with users' privacy preferences. By using P3P to manage privacy policies, websites can express them in a XML-based machine-readable format, and users can automatically check those policies against their preferences by the means of P3P-enabled browsers [3]. In this way, users do not need to read the privacy policies of every web site they visit [84] to asses whether it is compliant withe their

privacy preferences or not. Privacy Bird [85] and Privacy Finder [86] are examples of P3P user agents, able to compare P3P policies with users' preferences. A limitation of the P3P, shared by SPARCLE and PPMLP, is that it needs server-side adoption, which is not easily obtained: according to the results presented in [87], only 20% of the websites amongst the E-Commerce Top 300 is P3P enabled.

The PrimeLife Privacy Dashboard [88] is a recent browser extension which is addressed to the user and aims at evaluating the quality of a website. To this end, it collects information about the website such as whether it has a P3P policy, whether it collects cookies, and whether it is certified by trust seals. The dashboard provides then a visual 'privacy quality' rating of the website: the presence of a P3P version of the privacy policy will increase the quality rating, while the presence of external or flash cookies will decrease it. In addition, the dashboard allows the user to set preferences and check what data has been exchanged with a website; also, if available, it translates the website P3P policy into plain English. The low adoption of P3P limits the effectiveness of this approach: a website may have a good privacy policy, but it may be get a low rate just because of the lack of a P3P version of the policy.

As opposed to existing solutions, our approach is a pure client side solution and only requires the existence of a plain text privacy policy. As such, it can be easily adopted by privacy-concerned users as no special support from the server side is needed. The trade-off is that it does not cover enforcement of the privacy policy at the server side.

3.3 The Privacy Completeness Analyzer

The completeness analyzer is the framework we devised to parse and analyze privacy policies in order to evaluate their completeness grade. The completeness grade is computed according to the category coverage, namely whether a given category is covered by a policy or not, and to the weight the user gives to each category. We can formally express these concepts as follows:

Definition 5 (Category Coverage – $cover(P, \lambda_i)$**)** *Given a privacy policy* P *and a sequence of privacy categories* $\lambda = \langle \lambda_1, \lambda_2, ..., \lambda_n \rangle$*, we define* $cover(P, \lambda_i)$ *as a boolean function which takes value* 1 *if the category* λ_i *is covered by* P*, the value* 0 *otherwise.*

Definition 6 (Privacy Policy Completeness Grade – $G(P)$**)** *Let* P *be a privacy policy,* $\Lambda = \langle \lambda_1, \lambda_2, ..., \lambda_n \rangle$ *be a sequence of privacy categories and* $\langle \psi_1, \psi_2, ..., \psi_n \rangle$ *where* $\psi_i \in [0, 1]$ *be the corresponding sequence of category weight as assigned by the user. Given the category coverage* $cover(P, \lambda_i)$ *for all the privacy categories*

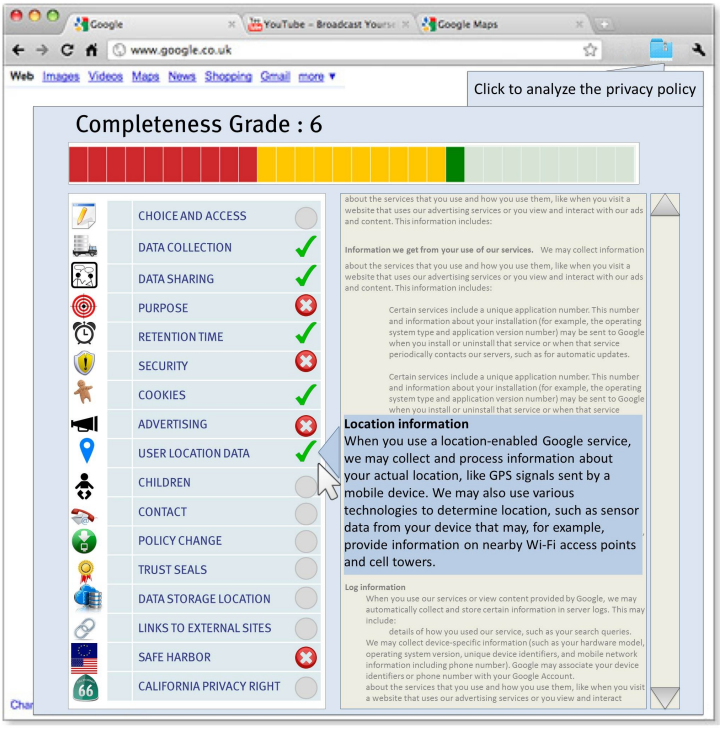


Figure 3.1: The User Interface.

$\lambda_i \in \Lambda$, *we define the completeness grade* $G(P)$ *as follows:*

$$G(P) \;=\; N \cdot \sum_{i=1}^{n} \psi_i \cdot cover(P, \lambda_i)$$

Note that $N = 10/\sum_{i=1}^{n} \psi_i$ *is a normalization factor which scales results to* $[0, 10]$. *If all weights* ψ_i *are* 0*, i.e. the user does not care about any of the categories, the grade of any policy is defined to be* 10.

Figure 3.1 shows an example of a graphical interface that can be used to present the results of the completeness analyzer to the user. The overall completeness grade $G(P)$ is presented using the traffic light metaphor. Then, for each privacy category, we show whether it is covered (green tick; $cover(P, \lambda_i) = 1$, $\psi_i \neq 0$), not covered (red cross; $cover(P, \lambda_i) = 0$, $\psi_i \neq 0$) or not considered relevant by the user (gray

circle; $\psi_i = 0$). The user can also select each of the covered categories and browse the paragraphs of the policy that belong to that category: in the figure we show the user while she is browsing the category *User Location Data*.

3.3.1 Methods

To assess whether the category λ_i is covered by a privacy policy, we apply text classification and machine learning techniques. Text classification is the automatic activity of labeling natural language text with thematic categories from a predefined set, while machine learning is an inductive process to automatically build a text classifier by learning from a set of pre-classified documents [89].

In our context, the pre-classified documents are paragraphs from manually labeled privacy policies. This set of policy paragraphs (a.k.a. corpus) is used to train a classifier, to correctly assign a privacy category to the paragraphs of an unlabeled policy.

Figure 3.2 depicts the process we used to build and evaluate our text classifier. The first step of the process regards the definition of privacy categories, which are extracted from privacy regulations and common practices as discussed in Section 3.3.2. Once privacy categories have been defined, we can use them to annotate the corpus by manually labeling each paragraph of the corpus with one of the privacy categories. The corpus we used is described in Section 3.3.3. The labeled corpus is then transformed into a suitable representation for text classification by applying standard text formatting operations such as tokenization, stemming and vectorization [90]. At this point, the corpus can be split into two different sets: a training and a testing set. The training set is used as input to the learning phase, while the testing set is kept apart so that a completely fresh set is available for the final evaluation of classifiers.

The learning phase starts by pre-processing the training set, namely optimizing it for an effective learning as discussed in Section 3.3.4. During the learning phase, different classifiers are built by using different machine learning algorithms as explained in Section 3.3.5. The performance of the classifiers, measured with the metrics presented in Section 3.4.1, is evaluated by using the *k-fold cross-validation* technique [91] which is discussed in Section 3.4.2. Finally, we test different settings and configurations, and several optimization techniques in order to find the classifier with the best performance . The results of our tests are presented in Section 3.4.3.

3.3.2 Privacy Category Definition

To define the privacy categories we considered different privacy regulations and directives, such as the EU 95/46/EC and the EU 2006/24/EC Directive on the protection of individuals, the guidelines issued by the Organization for Economic Cooperation

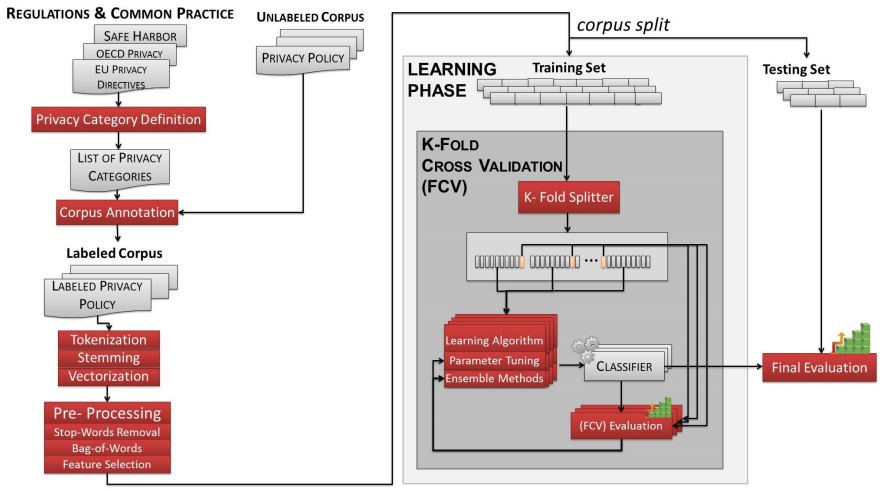


Figure 3.2: Process to Build and Evaluate a Text Classifier.

and Development (OECD), the Fair Information Practice Principles published by the U.S. Federal Trade Commission (FTC), and the Safe Harbor Framework [92]. We also considered regulations concerning the privacy of specific groups of people, like the Children's Online Privacy Protection Act from which we derived the category *Children*. A more comprehensive survey of current regulations on privacy protection can be found in [93]. The principles mentioned above are not all mandatory, however they provide guidance for authoring privacy policies. Organizations such as TRUSTe[1], eTrust[2], and Webtrust[3] require compliance with such directives to release their trust seal to an organization. We classify privacy categories in two types: *Core Categories* and *Additional Categories* that are described as follows.

Core Categories. The following categories have been directly adapted from Directive 95/46/EC and 2002/58/EC 2002, and from the OECD guidelines:

- **Choice and Access** provides information about the user's privacy choices, such as opt-in/opt-out options, and user's rights to access, modify and/or delete the information collected by the website.

[1] `http://www.truste.com/`
[2] `http://www.etrust.org/`
[3] `http://www.webtrust.org/`

- **Data Collection** explains how and what kind of personal data may be collected by the website.
- **Data Sharing** explains whether, and under which conditions, the website will share user's information.
- **Purpose** explains for which purposes the data will be collected and used.
- **Retention Time** explains for how long personal data is retained by the website.
- **Security** explains whether security technologies, e.g. the use of SSL or access control policies, are applied by the website.

Additional Categories. The following categories are related to topics that, although not mandatory, are usually addressed by privacy policies, e.g., how users' data is used for advertising or how policy changes are communicated to the users.

- **Cookies** explains whether the website makes use of cookies, and the information the cookies store.
- **Advertising** explains how the user's data is used for advertisements, and whether it is controlled by the website itself or by third parties.
- **User Location Data** explains how the website manages user's location information.
- **Children** explains the website's policy regarding the collection and use of personal information of children.
- **Contact** provides website's contact information, such as the registered office, or the address users can use for further questions or complaints.
- **Policy Change** explains how updates to the privacy policy are managed, and whether and how the users will be informed of such changes.
- **Trust Seals** explains whether the website has been awarded with trust seals (e.g. the TRUSTe's seal), signifying the website's policy and practices are compliant with trusted third party's requirements.
- **Data Storage Location** explains where the personal data is transferred to, stored and processed (e.g. whether it goes from Europe to USA, where a different privacy legislation applies).
- **Links to External Sites** warns the users about the fact that the current privacy policy does not cover third party websites reachable with external links.

- **Safe Harbor** explains the website participation in, and self-compliance with, the U.S.-EU/Swiss Safe Harbor Framework.

- **California Privacy Rights** is specific to California residents, and defines whether and how users have the possibility to request the records of the personal information disclosed to third parties (e.g. in the last year).

Note that users can decide what categories are relevant to them, therefore users who are not from California will probably not be interested in the *California Privacy Rights*, as well as the *Safe Harbor* category could be (automatically) discarded for websites that are not located in the USA. Discarded categories are not taken into account while computing the final completeness grade of a policy; therefore a European website lacking the *Safe Harbor* category, can still potentially reach the maximum grade available. Also, notice that our approach allows the definition of new categories, e.g. as consequence of changing in legislation like the upcoming EU Data Protection Regulation. The impact of adding a new category to our model is discussed in Section 3.4.3.

3.3.3 The Corpus

The corpus is an initial set of paragraphs extracted from privacy policies, labeled with the categories described before and used during the learning phase to build the classifier. The corpus is divided in two subsets: the training set ($\sim 70\%$ of the corpus), used to train and validate the classifier, and the testing set (the remaining $\sim 30\%$), used during the final evaluation of the classifier. The two subsets are strictly separated, i.e. the testing set cannot be used during the learning phase.

By using machine learning techniques, one can only train the patterns present in the corpus. Therefore, the corpus must be large enough to contain all the patterns of interest, and to cover all the categories with a sufficient number of samples. Our corpus has been extracted from 64 privacy policies of the most commonly used websites (e.g. Google, Amazon, FoxNews) and each paragraph of such policies has been manually annotated. This resulted in 1049 annotated paragraphs, 772 of which are used as training set, and 277 as testing set.

Clearly, reliability of the annotated corpus is a prerequisite for training a high quality classifier. To verify the reliability of the annotations, a basic agreement test was performed. An agreement test aims at measuring the level of agreement amongst different annotators: a high level of agreement shows objectivity of the annotation process, and indicates a high quality of the corpus. Since the labeling of our corpus was performed by a single annotator, to carry out the agreement test, a second annotator was asked to independently label a selected subset composed of 102 para-

graphs, extracted from the training set. The subset was selected in such a way to have a distribution of paragraphs over categories close to the one of the super set.

The results of the agreement test show that the two annotators agreed in 91% of the cases. Three of the nine items upon which they disagreed were marked as *ambiguous* by the second annotator who assigned two labels to the same paragraph: the second label assigned was always matching the first annotator's decision. If we discard these cases, the agreement level raise up to 94%. These results allow us to state that the corpus annotation is sufficiently reliable. However, the results also suggest that in some cases even for human classifier is not easy to label a paragraph: this is something that needs to be kept in mind when analyzing the results of the automated classifiers.

3.3.4 Preprocessing

In general data preprocessing consists of applying a set of techniques, such as cleaning, normalization, transformation, feature extraction and selection, to the ('raw') corpus to eliminate noise that could decrease the quality of the learnt classifier [94]. Using the resulting ('polished') corpus as input, usually significantly increases the effectiveness of the machine learning algorithms and/or the quality of the resulting classifiers.

Here we apply cleaning (stop-words removal) and transformation (bag-of-words representation) techniques. A stop-word is a very common word (e.g. conjunctions or articles) that is of very little help in the further analysis of the documents. The stop-words removal, aiming at reducing the noise produced by very common words, has been done using a common English stop-words list[4]. The bag-of-words transformation represents a document as a vector of words. The weight of each word is given by the number of its occurrences. Instead of the simple word frequency, we apply the tf-idf (term frequency - inverse document frequency) weighting [95], because it has the advantage of decreasing the weight of words appearing very often in the document while increasing the weight of rare words.

Feature selection [94] is an optimization technique that can have positive impact on the performance, enabling learning algorithms to operate faster and more effectively by removing irrelevant and redundant information. During the feature selection, a correlation score between a word and each category is computed; only the words with the highest scores are selected. The impact feature selection has on performance will be discussed in Section 3.4.3.

[4] A collection of stop-words lists in many languages can be found at: `http://www.ranks.nl/stopwords`

3.3.5 Learning Algorithms

The problem of constructing a text classifier has been widely treated in literature, resulting in the existence of many machine learning algorithms. The goal of these algorithms is to build a function that, given a document, returns a value, usually called *categorization status value* or CSV_i which represents the evidence that the document should be assigned the class C_i. The CSV_i score for classifying documents can represent either a probability value or a measure of vector closeness depending on which learning algorithm is used [89]. Machine learning algorithms can be divided based on several characteristics [96] including supervised versus unsupervised approaches, and the use of linear versus non-linear combination of features. In text classification a feature usually corresponds to a word. As we work with an annotated corpus we use the supervised learning.

Since the algorithm selection problem, i.e. the problem of selecting the algorithm that performs best for a given task, is still unsolved [97], we test the performance of different algorithms to determine which algorithm achieves the best results for our task and whether such results are satisfactory. The algorithms we selected are the following: Naïve Bayes (NB), Linear Support Vector Machine (LSVM), and Ridge Regression (Ridge) as 'linear' algorithms and the k-Nearest Neighbor (k-NN), the Decision Tree (DT) and the Support Vector Machine (SVM) with non-linear Kernel as 'non-linear' algorithms. In addition, we applied several ensemble learning methods which attempt to build more effective classifiers by combining the outcome of different algorithms. Following, we briefly describe the main characteristics of the algorithms we selected and the reasons behind their selection. For a more in-depth explanation of the different algorithms please refer to [98].

- the Naïve Bayes based family is a group of closely related probabilistic methods where CSV_i is the probability that a document belongs to a category C_i. We selected it because of its computational efficiency, important for real time applications [99]. Its most commonly used variants are the multivariate Bernoulli event model, and the multinomial event model. We selected the latter variant, because it generally outperforms the former [100].

- The LSVM [101] is one of the most popular machine learning algorithms, based on the representation of training items as points in space. The mapping is then done in such a way that items belonging to different categories are divided by a clear linear gap. The classifier obtained with this algorithm labels new items into a category based on which side of the gap they fall on. It has been selected because it usually behaves good in text classification tasks [102].

- The Ridge Regression classifier [103] is a variant of the least squares regression model, selected because it is computationally efficient, and because it solves

the problem of possible not-unique solutions [104]. Moreover, just like all the others linear classifiers, it is supposed to do well for text classification.

- In the k-NN classifier [105], the k represents the number of closest neighbors (training items) considered. The k-NN has a minimal training stage and an intensive (and timing consuming) testing stage. The basic idea is that the k-NN classifier assigns to a new item the category owned by the majority of its k nearest neighbors. The k-NN classifier is known as one of the top-performing approaches in text categorization tasks [106], therefore, we want to check how it performs in our context.

- A decision tree [107] is a structure where each internal node represents a test on the value of a feature, each branch represents the result of the test, and each leaf is a class label. Decision tree based approaches are particularly appealing for text learning, because their performance compares favorably with other learning techniques in this setting [108]. There are several classic decision tree algorithms, here we use the CART version [109].

- Non-linear SVM [110] algorithms are useful when the gap between different categories cannot be linearly modeled and, thus, a more complex function is needed. There are several alternatives to the linear kernel used in the linear SVM [111]. In our experiments we use the RBF kernel, because, for text classification tasks, it outperforms several other variants [112].

- Ensemble learning [113] is an optimization technique that generates a classifier as a combination of others. To create ensembles of different types of classifiers, (e.g. SVM, Ridge Regression and Naïve Bayes) it is advised to use methods such as the *voting committee* [114], and the *stack generalization* [115]. The voting committee method combines the results of different classifiers, into a voting committee: the category that has been selected by the most of the classifiers is chosen to label the item. On the other hand, the main idea in the stack generalization method is to learn a meta-level (or level-1) classifier based on the output of base-level (or level-0) classifiers [116]. In our experiments we test the voting committee and two variants of the stacking method: the *probability variant* (Prob.), and the *prediction variant* (Pred.) [116].

3.4 Evaluation

Evaluating the performance and the effectiveness of a text classifier built by using a specific machine learning algorithm, is particularly important to understand how suitable the classifier is for a given task. In this section, we describe how we measure the

effectiveness of a classifier and what method we use for its evaluation. We also discuss the results obtained by applying certain optimization techniques to the classifiers (e.g., feature selection). Additionally, we provide an estimation of expected performance over new samples by measuring the accuracy over the testing set. Finally, we test the impact that adding a new category to the system has over performance.

3.4.1 Metrics

The effectiveness of a classifier is usually computed in terms of the information retrieval notions of *precision* and *recall* [89, 106]. Intuitively, the *precision* represents the rate of items classified as belonging to a class which are actually from that class, while the *recall* represents the rate of items from a class which are indeed labeled as belonging to that class. The F_β score of a classifier combines precision and recall into a single value that can be used to compare the performance of different classifiers.

Let consider a corpus of documents D divided into $D_1, \ldots, D_N$ according to the category label, and a classifier $L : D \rightarrow \{1, \ldots N\}$. Given a category label $i \in 1, \ldots, N$, we can define the true positive (TP), i.e. the number of items in D_i correctly classified as belonging to the category i, the false positive FP, i.e. the number of items not belonging to D_i to which it has been assigned a category i , the true negative(TN), i.e. the number of items which do not belong to D_i and to which it has been assigned a category $j \neq i$, and the false negatives (FN), i.e. the number of items which do belong to D_i but have been assigned to the category $j \neq i$. In formulas:

$$
\begin{aligned}
TP_i &= |\{d \in D | L(d) = i \wedge d \in D_i\}| \\
FP_i &= |\{d \in D | L(d) = i \wedge d \notin D_i\}| \\
TN_i &= |\{d \in D | L(d) \neq i \wedge d \notin D_i\}| \\
FN_i &= |\{d \in D | L(d) \neq i \wedge d \in D_i\}|
\end{aligned}
$$

In addition, if we use $|D|$ to denotes the size of the set of documents D, we can define the *precision*, *recall* and F_β as follows:

$$
\begin{aligned}
Precision &= \sum_{i=1}^{N} \frac{|D_i|}{|D|} \cdot \frac{TP_i}{TP_i + FP_i} \\
Recall &= \sum_{i=1}^{N} \frac{|D_i|}{|D|} \cdot \frac{TP_i}{TP_i + FN_i} \\
F_\beta &= (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}
\end{aligned}
$$

Precision and recall are a weighted average over the different categories. The β in F_β determines the relative importance of precision and recall where smaller numbers, e.g. $1/2, 1/3, \ldots$ indicate a higher importance of precision, while large numbers, e.g. $2, 3, \ldots$, are used if recall is more important. In the following, we use the F_1 score ($\beta = 1$, namely the same importance is given to precision and recall) to measure the effectiveness of a classifier. Intuitively, the higher the F_1 score, the lower the false predictions ($FP + FN$), the better the effectiveness of a classifier.

3.4.2 Cross Validation

In the experiments described below, we make use of the *k-fold cross validation* technique [91] to estimate the performance (in terms of F_1 score) of different classifiers. Typically, the k-fold cross validation consists of partitioning the training set into k equally (or nearly equally) sized folds (a fold is a set of samples from the training set). After the partitioning, k iterations of training and testing are performed, in such a way that in each iteration a different fold of the data is used for testing, while the remaining k - 1 folds are used for learning. We use the 10-fold cross validation because this value of k is particularly attractive [91].

In our experiments we apply the $n \times 10$-fold cross validation [117], i.e. the 10-fold cross validation is repeated n-times, every time using a different random selection of folds from the training set. Note that during the cross validation we only use the samples in the training set, the samples in the testing set are still kept separated: they will be only used to test the classifiers once they are ready. In this way the results are obtained over $n \times 10$ different training and testing sets. The F_1 score and the error are then estimated respectively as average of the values obtained at each repetition. Note that the cross-validation method does not produce a classifier, but it helps to estimate the average performance of classifiers learned from sub-sets of the training set. The performance (F_1 score) obtained from cross-validation is used as an

Table 3.1: Parameter Tuning

Classifier	Parameter(s)	Parameter Domain	Best Value
k-NN	k	$1, 3, 5, \ldots, 49$	19
SVM (loose search)	$log_2(\gamma)$	$-15, -11, -7, -5, -3, -1$	-5
	$log_2(C)$	$-3, -1, 1, \ldots, 15$	7
SVM (fine search)	$log_2(\gamma)$	$-8, -7, -6, -4, -2$	-4
	$log_2(C)$	$5, 6, 7, 8, 9, 10, 11$	5
LSVM	$log_2(C)$	$-3, -1, 1, \ldots, 15$	-1
	regularization method	$l1, l2$	$l2$
Decision Tree	max_depth	$6, 8, 10, 12, 14, 16$	16
	min_split	$2, 3, 4, 5, 6$	5

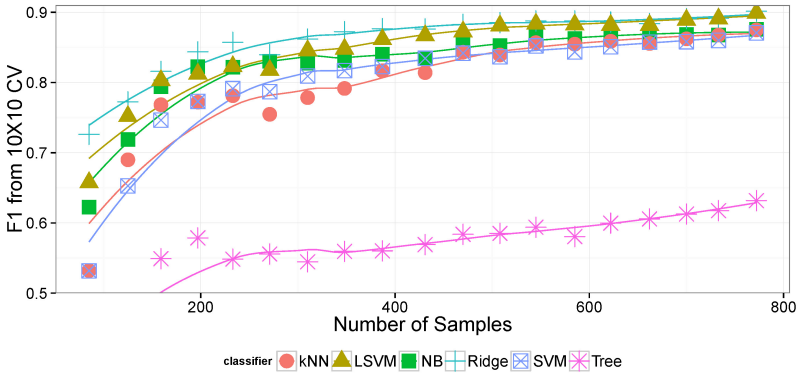
informal estimation of the performance of a final classifier, built by training on the whole training set[5].

3.4.3 Experiments

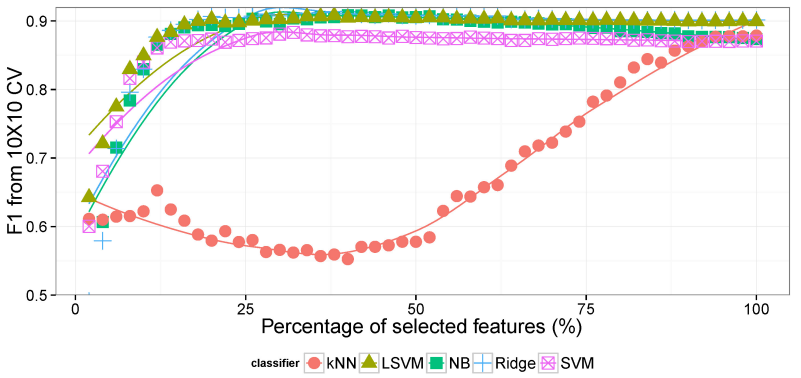
In this section we describe the experiments we carried out to evaluate the performance of the classifiers, and the impact of different configuration settings and optimization techniques. We also discuss the results of the experiments and the implications of these results. Note that in the first experiments (1-5) the privacy category *Purpose* is left out, while it is introduced in the last experiments (6): this allows us to discuss the impact that adding a new category has on our model in terms of cost (manual labeling) and performance (accuracy of the model).

Parameter Tuning. The machine learning algorithms we used in our process (except for Ridge Regression and Naïve Bayes) depend on certain parameters (e.g., the number of clusters k in the k-NN algorithm). The value of these parameters has to be tuned in such a way that performance are maximized. Parameter tuning is usually done empirically, by running the algorithm with different values for a parameter, and by selecting the value that leads to the best F_1 score. When an algorithm depends on multiple parameters, a grid search is used for the tuning [118]. Table 3.1 shows the parameter(s) required by each of the algorithms we tested, together with the domain we explored during the tuning, and the value which leads to the best performance. For readability reasons, for some of the parameters we show the $\log_2$ of the value. In the following experiments, unless stated differently, for each algorithm we set the parameter(s) to their *Best value*, as shown in last column of the table.

[5] `http://cseweb.ucsd.edu/~elkan/250B/classifiereval.pdf`



(a) Sample Size



(b) Feature Selection

Figure 3.3: Impact of Sample Size and Feature Selection.

Impact of the Sample Size. Choosing the size of the training set is the result of a compromise between the effectiveness achievable by a classifier and the effort required to build the training set. In case the training set is too small, this may significantly degrade the effectiveness of a classifier. On the other hand, diminishing returns in improvements to the classifier make that increasing the training set is, at a certain point, no longer worth while. In this experiment, we test how having a training set with an increasing size will influence the F_1 scores of the classifiers. In addition, we

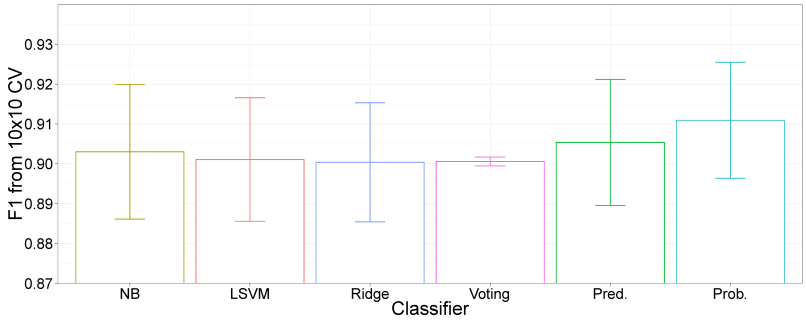
validate that the size of our training set is adequate for our task and that increasing its size would only lead to marginal benefits. For this experiment, we divided the training set into 18 incremental subsets. The first subset contains 10% of the whole training set, while each following subset contains 5% more items than the former. Intuitively, the last subset will be equal to the training set. Figure 3.3a shows the F_1 score of the six classifiers we are comparing. The F_1 score is computed for each of the 18 subsets by using the 10×10-fold cross validation. We can notice that the F_1 score is below 80% if we choose a training set with less than 200 paragraphs, while all the classifiers (except for the decision tree) have stable performance with a training set with more than 700 samples. This means that after 700 samples, an increase in the size of the training set only leads to marginal improvements, hence we can argue that the size of our training set is adequate for our settings. Figure 3.3a also provides a first idea about the general performance of the classifiers we are testing. Let us consider the F_1 score obtained over the complete training set. In this case the classifiers created with the Ridge and LSVM algorithms are the ones with the best performance with an F_1 score of, respectively, 90.0% and 89.9%. On the other hand, the classifiers obtained by adopting Naïve Bayes, k-NN and SVM are less effective, with an F_1 score of, in order, 87.5%, 87.4% and 87.1%. The decision tree algorithm leads to the creation of a classifier which performs very poorly, with an F_1 score of 63.1%. Given these results, we discard this classifier from our further experiments.

Impact of Feature Selection. In this experiment, we evaluate the performance of the different classifiers when applying the χ^2 feature selection method [119], a very common method used to test the independence of two events. More specifically, in feature selection this test is used to verify whether the occurrence of a specific term (*feature*) and the occurrence of a specific class are independent. In case of dependency, the feature is selected for the text classification. Since this method leads to a variation in the number of samples, we change the parameters k (for k-NN) and γ (for the SVM) to the values $k = \#\, features/2\,\#\, classes$ and $\gamma = 1/\,\#\, features$. Figure 3.3b displays the impact feature selection has on the performance. The results indicates that Ridge and LSVM reach their best performance at 20% of feature selected, while Naïve Bayes reaches such peak with slightly more features (30%), and start to decay with more than approximately 60% of features selected. The k-NN classifier performs poorly until almost all features are selected. For the next experiments a 40% feature selection is applied, given the fact that for that value, all classifiers (except for k-NN) have already reached their best performances, and NB performance is not yet decreasing.

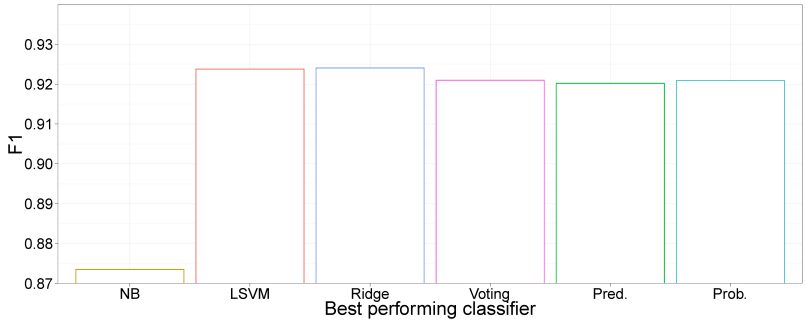
Performance of Ensemble methods. In this experiment we compare the performance of the best single classifiers (Ridge, LSVM and NB) with the ones obtained by applying ensemble methods, such as the voting committee, and the probability (Prob.) and prediction (Pred.) variants of the stacking. Figure 3.4a shows the average mean and error obtained by applying a 10×10-fold cross validation and a 40% feature selection. SVM and k-NN do not appear in the figure because of their low performances with 40% feature selection ($\sim 55\%$ for k-NN, and $\sim 87\%$ for SVM as shown in Figure 3.3b). We can notice that the probability variant (Prob.) of the stacking ensemble outperforms the best single classifier (NB) with a F_1 score that is about 1% higher (91.1% versus 90.3%). Since all the classifiers have high performance (between 90-91%) it is difficult to prefer one over the other. The voting ensemble method is the one presenting less variance (the error bar is the shortest), something that is usually very appreciated when choosing a classifier. However, ensemble methods are also more computationally expensive than single classifiers, simply because they use several of them, and not necessarily in parallel. Therefore, if the gain in performance is not relevant, single classifiers can be preferred over the ensemble ones since they will require less computational effort.

Performance over the testing set. The evaluation over the testing set aims at providing an estimation of the classifier performance over new samples. This test has been done at the very end, once all the classifiers have been fully specified. The test checks whether the classifiers suffer from the over-fitting problem, namely the fact that a classifier is too specialized towards the training set and, therefore, it does not correctly classify new samples (represented by the samples in the testing set). In case of over-fitting, performance over the testing set would be significantly worse than performance over the training set. Because no information about the testing set was used during the training of the classifiers, the results of this experiment should be indicative of the performance in real scenarios. Figure 3.4b presents the results of the test, showing that all the classifiers (except for Naïve Bayes) have very similar F_1 score ($\sim 92\%$), even better than the score achieved during the validation phase, thus the over-fitting problem seems to be absent. On the contrary, Naïve Bayes performs worse over the testing set ($\sim 87\%$) than over the training set ($\sim 92\%$), suggesting that it actually suffers from the over-fitting problem and it should be therefore discarded.

Impact of adding a new category (Purpose). In this test we check the impact that adding a new category has on our model, in terms of costs and performance. To perform this test we added 50 new paragraphs to the corpus (35 to the training set and 15 to the testing set) regarding the category *Purpose*. We chose to add the category *Purpose* at the very end, mostly because of the difficulty we encountered during its



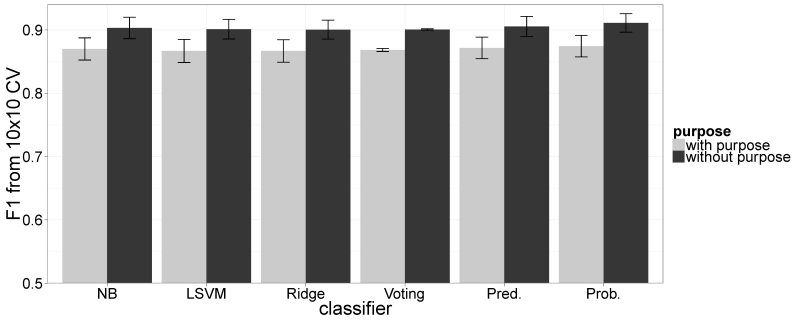
(a) Comparison with Ensemble Methods (Cross Validation)



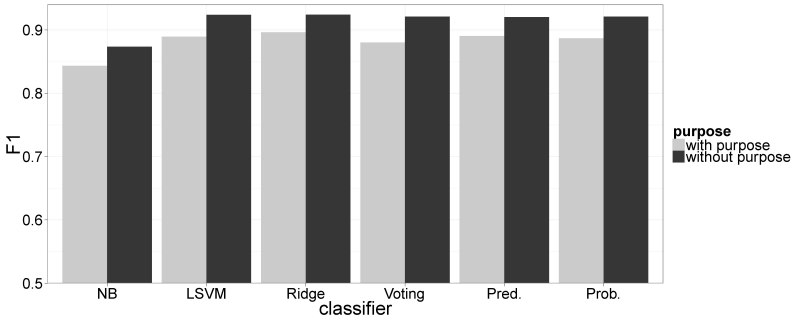
(b) Evaluation over the Testing Set

Figure 3.4: General Estimated Performance.

manual labeling. Since paragraphs regarding *Purpose* are difficult to isolate (they are often interleaved with other categories such as *Data Collection* or *Retention*) the performances the system has when *Purpose* is included, are likely to be a lower bound for our system. Figure 3.5a shows the results of the cross validation test when *Purpose* is included and when it is left out. We can notice how the best classifiers continue to be the best even when the new category is added, suggesting that we can choose the best classifier independently from the categories. However, it is also possible to notice an overall decrease in performance of about 3.6% when *Purpose* is included, confirming that *Purpose* is indeed a category difficult to handle. Figure 3.5b shows the results obtained over the testing set and compares them with those obtained when *Purpose* is not included. The trend is again confirmed, with the NB classifier performing poorly and the others showing similar F_1 score, between 89-90%. In overall these results



(a) Cross Validation



(b) Evaluation over the Testing Set

Figure 3.5: Impact of Adding a New Category (Purpose).

show that, if required, new categories can be added to the systems. The costs in terms of time depend on the number of new samples which needs to be labeled, while the costs in terms of performance seems to be limited to a decrease of about 3%. Note that the classifiers need to be trained again since the corpus changed, but no parameter tuning or optimization technique needs to be re-applied.

As general conclusions, we can say that LSVM, Ridge, Voting, Pred. and Prob. perform equally good in terms of F_1 score. However, in case we use one of the ensemble methods, we need to account for an increase of computational costs (using ensemble methods costs approximately 10 times more than using a single classifier). Amongst the single classifiers, we also have to point out that Ridge and LSVM cost from 10 to 20 times more than the Naïve Bayes, so the latter may represent a reasonable fallback option when costs are very important (e.g. on mobile devices) and an F_1 score of

87% is considered sufficient. However, recall that the Naïve Bayes classifier suffers from the over-fitting problem, so it can eventually lead to lower performance. As final remark, note that the maximum effectiveness we reach with an automatic classifier ($\sim 92\%$), can be considered adequate to our task, since it approximates the results obtainable with a human classifier. As seen in the agreement test, when different human judges are asked to classify a privacy policy, the objectivity (that can be seen as a measurement of effectiveness) is close to the 92% level.

3.5 Limitations

The solution we discussed in this chapter shows high accuracy in determining whether a text paragraph can be associated to any of the predefined privacy categories. Nevertheless, the solution presents some limitations that we discuss in this section. The first clear limitation is the fact that our solution currently only works for privacy policies that are written in the English language. Though the framework itself is general, to get it to work with other languages it is necessary to re-train the classifier, namely the whole process (privacy labeling and classifier learning) has to be executed again for every new language. Although this task might be time consuming, it only needs to be done once: when the classifier is ready it can be used to automatically evaluate new privacy policy in that specific language. Another limitation of the solution we proposed is that we assume a one-to-one mapping between a policy paragraph and a privacy category. As a matter of fact, a single paragraph in the privacy policy might address (even if only partially) more than one category. To solve this problem it would be necessary to re-label our corpus by adding the possibility of associating multiple categories to each paragraph. More about this is discussed in Chapter 7, when we provide directions for future work.

3.6 Conclusions

In this chapter, we presented a solution to evaluate privacy policy completeness by applying machine learning and text classification techniques. The completeness grade is an important factor to evaluate the overall quality of a policy: the more complete the policy, the more the information provided to the user. To prove the feasibility of our approach, we tested several automatic text classifiers obtained by applying machine learning over a corpus of pre-annotated privacy policies. The results show that it is possible to create automatic classification with a high accuracy ($\sim 92\%$), similar to the one obtainable with a human classifier.

Based on the results we obtained with our experiments, we can provide some

guidelines about which classifier is better to use for our completeness analyzer. The classifiers performing best are LSVM and Ridge Regression with an accuracy of more than 92%. The ensemble classifiers have similar performance but they require a lot more computation. If computational costs matter, Naïve Bayes represent a possible choice: although its accuracy is lower than the other classifiers (about 87%), its computational costs are 10 to 20 times better than LSVM and Ridge.

Note that simplifying a privacy policy with a grade of completeness may lead to a lack of precision that can have legal implications [120]. For this reason, the privacy policy *as-is* has to remain the main source of information. That is why, although we provide a completeness grade, we also provide a structured way to browse the actual text of the privacy policy, to let the user access the complete information. Finally, a high completeness level indicates the policy covers the most important privacy categories, but no assumptions over their semantic value can be made. For this reason, a semantic analysis of the contents of a policy, able to assess whether what the policy states about a category is bad or good (e.g. '*sharing*' versus '*not sharing*'), is necessary. This semantic analysis will be the focus of the next chapter.

Chapter 4

Evaluating Websites: Privacy Policy Data Collection

In Chapter 3, we proposed a solution to automatically evaluate the privacy completeness of a policy written in natural language. The policy completeness is defined as the degree of coverage of privacy categories which are of interest to a user; it also evaluates the policy compliance with regulations and guidelines. A limitation of the solution we propose lies in its inability to deal with the semantic content of a policy, e.g. by establishing whether such content is good or bad for the user's privacy. For example, w.r.t. the Data Collection *category, the privacy completeness grade says whether the policy describes Data Collection procedures or not. However, it does not say whether the* amount *or the* sensitivity *of the data collected are excessive for the type of service provided (e.g. the social security number is required to register to a blog). In this chapter, we propose a solution which is able to extract semantic information out of a privacy policy. Especially, we extract the list of all personal data items collected by a website during its usage as stated in its policy. In addition, we propose a grade to evaluate the privacy costs of using a website, in terms of the amount of personal data items collected and their sensitivity.*

4.1 Introduction

To provide highly personalized and free services to their customers, companies collect a large volume of personal data. However, public institutions and users are concerned that the amount of data collected for a given purpose can easily become excessive and unjustified. For instance, requiring a user's credit card details is unnecessary if no payment is involved in the service requested by the user. To avoid excessive data collection, the European Union defined the data minimization principle according to what data collection must be "adequate, relevant and not excessive in relation to the purposes for which data is collected" (European Directive 96/46/EC).

Organizations collecting personal data are obliged to comply with the data minimization principle. Also, they have to state in their privacy policy what data will be collected and for what purpose it will be used. Although statistics show that users are deeply concerned about privacy [121], it is also true that users do not read privacy policies [7]. This leads to situations where the presence of privacy policies serves more as liability disclaimer for service providers than as a tool to protect privacy [122].

In this chapter, we describe a solution which exploits the valuable information contained in privacy policies. The main contribution of this chapter is the definition of a Information Extraction framework and of a set of patterns that can be used to extract the details of the data collected by a service provider according to what it is stated in its privacy policy. In this way, we can immediately communicate to the users the impact of using a service in terms of the amount of personal data they have to reveal. In addition, we allow the user to express a level of sensitivity to associate to each data item, e.g., to express the fact that *credit card details* are more sensitive than an *email address*. By knowing the data items collected by a website, and how sensitive each item is for a given user, we can compute the privacy *cost* of a website, expressing the price users have to pay in terms of personal data they need to reveal.

The solution we present is based on the use of Information Extraction (IE) techniques. We argue that by presenting the user with the list of data items collected by a website and with the privacy cost, we can increase users privacy awareness and helping them to take more informed decisions without the need of reading long privacy policies. For example, by looking at the list of data collected, users can judge whether the information required is justified by the service offered (e.g., in case credit card details are required to register to a mailing list). An example of how this information can be presented to the user, e.g., by using a browser extension, is presented in Figure 4.1.

The remaining of this chapter is organized as follows: we first discuss related work in Section 4.2, we introduce the privacy cost metric in Section 4.3 and the Information Extraction methods we employed in Section 4.4. In Section 4.5 we present

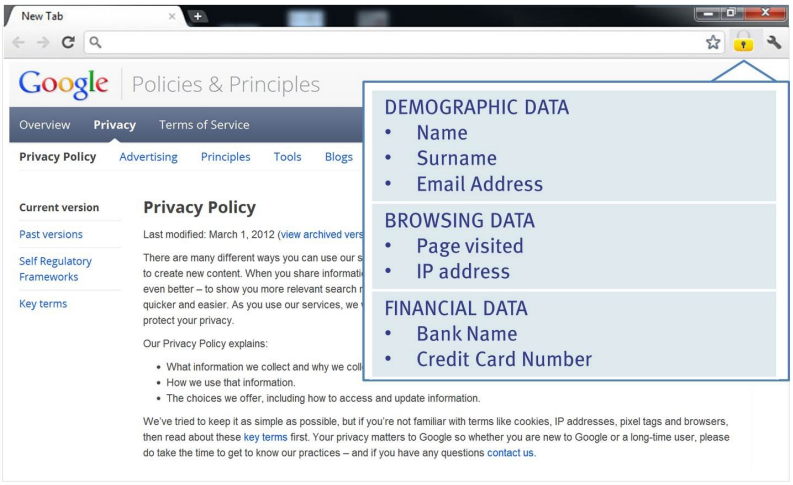


Figure 4.1: An Example of GUI as Browser Extension.

the process we adopted to build our IE system, while in Section 4.6 we discuss the effectiveness of such a system. Finally, in Section 4.8 we address conclusions and future work. Note that in this chapter we focus on the analysis of the policy's contents which refer to data collection. However, the same approach, if proved effective, can be applied to extract other information, such as the sharing practices or the retention period of personal data.

4.2 Related Work

Privacy policies represent, in most cases, the users only source of information about how their personal data will be handled and they are widely recognized as a powerful tool to drive users decisions. However, users typically do not read them, hence a valuable source of information is lost. In addition, understanding privacy policies, often written in a complex language, is not an easy task for end-users [74, 75].

There have been various attempts to improving privacy policy understandability by making them machine readable. Some of these approaches, such as SPARCLE [76, 77] and PPMLP [78, 79] have been discussed in Section 3.2. One of the most famous attempt is represented by the Platform for Privacy Preferences (P3P) [3] which enables websites to express policies in a structured format so that users can automatically match their preferences against it. In this way, users can decide to

interact only with those websites which comply with their preferences. However, P3P is not largely adopted by service providers. Thus, solutions such as the Privacy Bird [85], the Privacy Finder [86] and the Primelife Dashboard [88] that evaluate websites based on the contents of their P3P policy, are not widely applicable. UPP (User Privacy Policy) [123] is an approach similar to P3P, but mainly focused on social network websites. The mechanism defines a specific language that allows a user to define policies to protect his resources (e.g. pictures or videos). Other users (his *friends*) can access such resources only if they guarantee the enforcement of the user's policies.

Solutions based on privacy policies written in a machine readable language can only be successful if a large part of service providers adopts them. On the other hand, every website is obliged by law to publish a privacy policy written in natural languages. We believe it is important to exploits this source of information and apply natural language processing techniques to extract valuable information from the policies.

There are many studies in literature that evaluate the quality of natural language privacy policies. The evaluation criteria may refer to the policy readability [124] as well as to its content, structure, navigation, and accessibility [125]. In [126] the authors propose a privacy policy ranking that assigns different weights to different privacy categories such as the personal data collected, the usage of cookies or the guidelines followed to author privacy policies. In [127] the authors rank the privacy protection offered by a website by also looking at aspect related to its privacy policy such as the accessibility of the policy as well as the types of tracking technologies used.

A shortcoming of all these solutions is that the evaluation of the policy has to be manually computed. As far as we know, beside the completeness analyser we discussed in the previous chapter, the only work that applies automatic techniques to evaluate privacy policy is the one proposed in [128]. Here the authors start from the assumptions that privacy policy are often made ambiguous to trick the users. For example, the use of words such as *'occasionally'* or *'from time to time'* gives the service providers permission to send as many advertising emails as they want. To discover ambiguities that might represent a danger for the user, they propose a solution based on Latent Semantic Analysis to automatically analyze privacy policies.

4.3 Privacy Cost

The goal of the solution we discuss in this chapter is to automatically extract the list of personal data items collected by a website. This list can be shown to the users to help them in judging whether the privacy cost they have to pay for using a certain service

is adequate. Intuitively, the privacy cost w.r.t to data collected depends on the amount of data collected and on the sensitivity of each data item. The sensitivity of a data item, represents the user's perception of the harm the misuse of such data can cause to her. For example, a user might consider that her credit card details, if misused, can cause more harm than her e-mail address. Given the persona data collected by a website and their corresponding sensitivity, we can define the *privacy cost* as follows.

Definition 7 (Privacy Cost – υ) *Let R be the exhaustive set of personal data items, $d_i \in R$ be a data item and $\sigma_i \in [1, 10]$ be the sensitivity of d_i. Given a website, let $R_1 \subseteq R$ be the set of personal data collected by the website. We define the privacy cost υ of using the website as : $\upsilon = \frac{1}{\sum_{d_i \in R} \sigma_i} \sum_{d_i \in R_1} \sigma_i$*

The privacy cost value ranges from 0 to 1; the value 1 is reached in case a website collects all the personal data as defined in R. The list of personal data R can be drafted by following governmental guidelines which define as personal information every piece of data that refers to the racial or ethnic origin of an individual, as well as his/her political opinions, religious, mental health or sexual life. The sensitivity of each data item should be defined by each user during the preference settings. To alleviate the users from this task, a default sensitivity vector can be defined by following data protection guidelines or by by applying collaborative filtering to group of users. In the following, we describe how the set of personal data collected by the website, namely R_1, can be automatically extracted from its privacy policy by applying Information Extraction techniques.

4.4 Methodologies and Tools

In the solution we present in this chapter, we use *Information Extraction* (IE) to extract the list of data collected by a website, by analyzing what is stated in its privacy policy. IE is a technique for analyzing natural language texts and extract relevant pieces of information [129]. The analysis takes raw text as input and produces fixed-format, unambiguous data as output [130]. An IE system applies IE techniques to a specific scenario or domain. To build an IE system, an in-depth understanding of the contents of texts is needed [131]. IE systems have the advantage of accounting for the semantic contents of the text, rather than only for the presence/absence of given key words as it happens e.g. in Information Retrieval. Taking semantics into account gives the potential for systems that are accurate enough to really help users, reducing the time they need to spend reading texts such as privacy policies [130].

Privacy policies, due to their legal nature, show strong formality and fixed patterns. Because of this, applying IE techniques to privacy policy text may lead to high accuracy, as confirmed by the results presented in Section 4.6. The fact that IE only

extracts information on/in a-priori selected subjects/format fits well with our idea of showing to the user only specific information, i.e. the list of data collected.

The general architecture of an IE system may be defined as"*a cascade of transducers or modules that, at each step, add structure to the documents and, sometimes, filter relevant information, by means of applying rules*" [131, 132]. Figure 4.2 describes the architecture of our IE system, showing the cascade of modules we used. As first step, the raw privacy policy is given as input to the *Tokenizer* that splits the text into very simple tokens. To each token it is associated a type, for example, *number*, *punctuation* or *word*. The *Sentence Splitter* divides the text into sentences, while the *POS Tagger* annotates each word with the related part-of-speech (POS) tag. A POS tag specifies whether a word is a verb, a noun or another lexical category[1]. The *Named Entity (NE)* recognition module seeks the text for concepts of the application domain, while the *Annotation Patterns Engine* provides the means to define extraction rules, needed to detect the information wanted, according to specific syntactic-semantic patterns. Note that, as described in Figure 4.2, at each stage a new *Annotation Set* is added to the document. *Annotations* can be seen as metadata attached to part of the text giving specific information. For example POS tagger annotations serve to say that the word '*and*' is a *conjunction*. Every annotation has a type (e.g. *conjunction*) and a value (e.g. *'and'*), and belongs to an *Annotation Set* (e.g. POS Annotation Set) that groups together similar annotations. Annotations Sets that are output of earlier modules, can be used as input of later stages.

The structure shown in Figure 4.2 is a basic cascade of modules. Such a cascade can be extended by adding other modules for language processing. For example, coreference and anaphora resolution [133] could be added to verify whether two noun phrases refer to the same entity [134], e.g. '*email address*' and '*it*' in the sentence "*We store your **email address**, **it** will be used to inform you that your order has been shipped*". The use of ontologies, which associate a concept with the terms expressing it [129], may also be used to enrich the domain knowledge. Finally, syntactic parsing could help to find the connection between POS tags, e.g. to discovery that an adjective and a noun combine to be a 'Noun Phrase'. A POS tagger has the advantage of being really fast, but it does not give enough information. The syntactic parser can overcome this limitation, but it would typically need more time to analyse the text. The decision of whether or not to add one or more of these modules to our system is a trade off between the increased accuracy they can grant and the added computational costs they will bring. Since we aim at building a system with a good response time, in our system we will use the modules as presented in Figure 4.2. The benefits and costs of adding other modules are discussed in Section 4.8, as part of our future

[1]The complete list of POS tags we used is available at `http://gate.ac.uk/sale/tao/splitap7.html#x37-729000G`

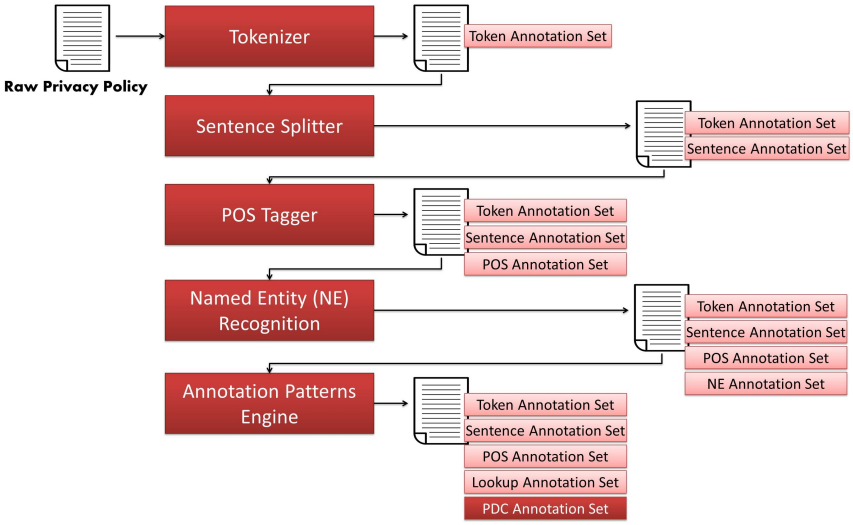


Figure 4.2: The Pipeline of our IE System.

work. To develop our system we used the GATE (General Architecture for Text Engineering) framework [135, 136], and the ANNIE (A Nearly-New Information Extraction System) [137] suite. These tools, amongst other functionalities, make available multiple implementations of the modules depicted in Figure 4.2.

4.5 The Process

The process we followed to build and evaluate our IE system is described in Figure 4.3. In this section, we give details for activities of this process such as the definition of Named Entity (Section 4.5.1), the selection, annotation and splitting of the Corpus (Section 4.5.2), and the creation of extraction rules (Section 4.5.3). The evaluation and the results are discussed in Section 4.6.

4.5.1 Named Entities

Information Extraction uses Named Entities (NEs) to represent important concepts within a certain domain[2]. For example, in the domain of novels, the *author*, the *title*,

[2]A discussion about the difficulties of defining *Named Entity* is available at `http://webknox.com/blog/2010/09/named-entity-definition/`

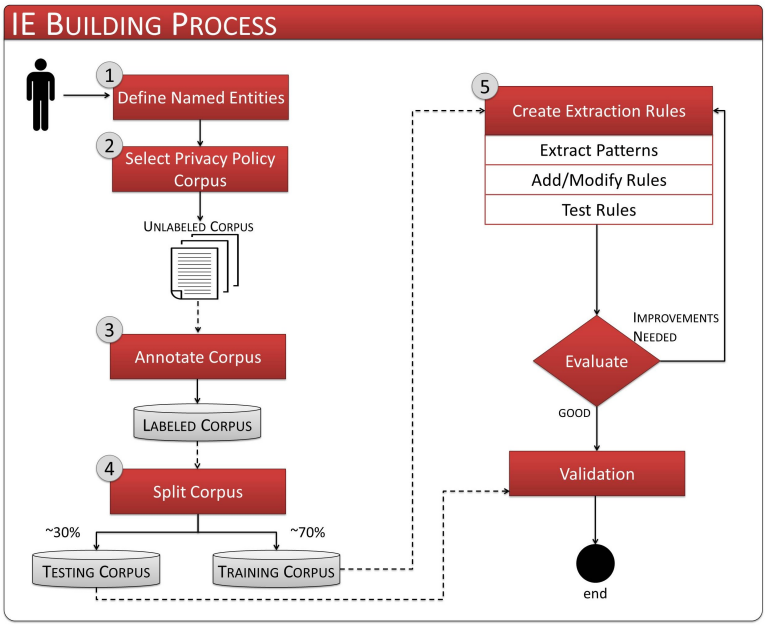


Figure 4.3: IE Building Process.

and the *characters* are important concepts that can be modeled as NEs. Defining NEs requires a deep knowledge of both the application domain and the type of information to be extracted. Since we aim at extracting the list of personal data collected by a website, we need entities modeling concepts of this scenario. To select our named entities we analyzed several privacy policies. In this way, we discovered that privacy policies statements about data collection can be expressed via three patterns:

1. The policy states the website may require user's personal data for some purposes, e.g. registration or shipping;

2. The policy states the user may provide his personal data to the website, e.g. by filling in profile information;

3. The policy states the website may use automatic tools to get users' personal data, e.g. cookies to track user behavior.

The discovery of these patterns led us to define the NEs presented in Figure 4.4 as part of our system. The Data Provider represents the data owner, generally the

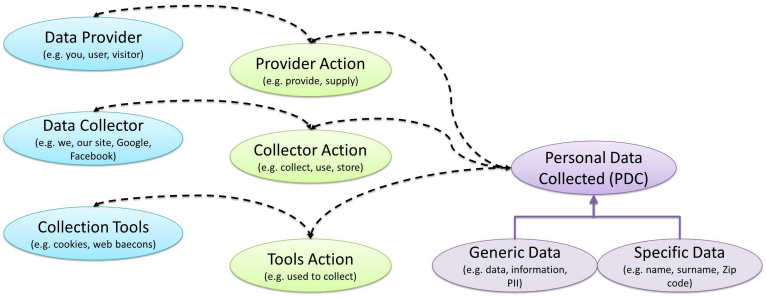


Figure 4.4: The Named Entities (NEs) in our System.

user, that provides personal information; the Data Collector refers to the website (or the company behind it) that collects the personal information; while the Collection Tools is used to describe web technologies, such as cookies or web beacons, used to track users' online activities. The NE mentioned so far can be seen as *subjects* of the *actions* leading to data collection. The corresponding actions are respectively modeled as i) Provider Action, including verbs such as *provide* or *supply*; ii) Collector Action, with verbs such as *request, collect, use, store*; and iii) Tools Action including verbs describing tracking activities, such as *track* or *monitor*. The core entity of our domain is the Personal Data Collected (or PDC), which we divide into two subentities: Generic Data to refer to general concepts of personal information such as *contacts information*, *personal identifiable information*, *browsing information*; and Specific Data to refer to personal data items such as *name*, *surname*, or *nationality*. Note that the list of personal data items it can be much greater than indicated in our examples, since it should cover any data that can be used to directly or indirectly identify a person (e.g., information such as IP addresses and house selling prices). The list of PDC, once extracted from the text, will be presented to the user.

Once the list of NEs has been defined, it is necessary to populate it by adding instances to the concepts, e.g. to list *name*, *surname* and *address* as instances of the entity *Specific Data*, and *you* and *user* as instances of *Data Provider*. The population process is performed by creating files which list instances corresponding to each NE (the so called gazetteer lists). Note that, as suggested in [138, 122], while creating the gazetteer list for the NE Specific Data we considered any personal data item as defined by the European Union, hence every piece of information that can identify an individual such as *name*, *surname* or *birthdate*, as well as tracking information such as the *current GPS location* and the *clickstream*.

4.5.2 Corpus

A corpus is a set of documents forming the core of an IE system. Its importance is twofold: its analysis drives the acquisition of knowledge necessary to extracting patterns and creating rules, and, once annotated, it is used to test the accuracy of the IE system. According to EU directives (e.g. EU 95/46/EC and the EU 2006/24/EC), service providers have to specify privacy policies that address specific topics of interest for the user, such as what data they collect, how long they retain the data, whether they share it with third parties. Each topic is usually discussed in specific paragraphs of the policy, leading to documents with share a fixed structure and similar patterns.

Our corpus is composed by privacy policies retrieved from websites of different application domains such as e-commerce (e.g., eBay, Paypal, Amazon), search engine (e.g., Google, DuckDuckGo), social networking (e.g., Facebook, LinkedIn), and news and communities (e.g., WordPress, FileTube, and TripAdvisor). Having policies coming from different domains is especially useful to enrich the gazetteer lists. For example, by analyzing social network policies, new instances of *Specific Data*, such as *profile*, *friends list* or *profile picture*, can be accounted for. Each privacy policy has been read by the authors, which isolated the paragraphs dealing with *data collection* practices. At this point, we created two corpus: *corpus A* composed of 128 paragraphs extracted from different privacy policies; and *corpus B* composed of 12 complete privacy policies. The use of corpus A is useful to focus on the modeling of recurrent patterns used to describe data collection practices. On the other hand, the use of corpus B, especially during the testing phase, is useful to analyze the feasibility of our approach. For example, by executing our process over a complete privacy policy (which is composed of many paragraphs) will allow us to estimate the response time and to verify whether our approach can be applied at runtime on thin clients such as browsers. In addition, we can test that extending the analysis to other sections of the policy does not decrease the accuracy of the system, e.g. by introducing too many false positives.

Once the corpus has been selected and the named entities defined, the corpus annotation task can take place. During the corpus annotation, a human annotator reads every document, and tags each occurrence of NE instances. For example, in a document with the sentence "*We collect your personal information such as your name, surname and gender when you register to our services*", '*we*' will be annotated as Data Collector, '*collect*' as Collector Action and so on, as described in Figure 4.5. Note that instances of Specific Data (SP) and Generic Data(GD) are annotated as PDC only if the text makes clear that such data will be retained by the website. For example, in a sentence like "*If you do not want to receive e-mail from us, please adjust your preferences*", the term *e-mail* will not be annotated as PDC.

Once the annotation task is completed, each sub-corpus is divided into a train-

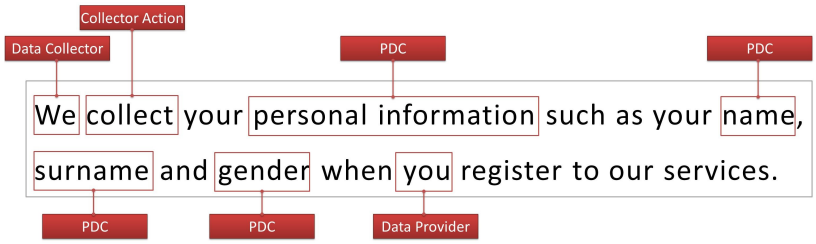


Figure 4.5: Example of Text Annotation of NEs.

ing (70% of the documents of the sub-corpus) and a testing set (the remaining 30%). After the splitting, corpus A contains 87 paragraphs in the training set, and 34 in the testing set, while corpus B contains 9 complete policies in the training set, and 3 in the testing set. The training sets are used to develop the extraction rules, while the testing sets are kept apart until the very end, when they are used to measure the system's accuracy. The training set of B is especially useful to verify whether patterns extracted by analyzing A do not occurs in other sections of a complete policy as well. In the following, we refer to the manual annotations as *Standard Gold Set*, and to the annotations resulting by running our IE system as *Response Set*. To evaluate the accuracy of the system, PDC annotations of the *Response Set* will be compared to PDC annotations of the *Standard Gold Set*. Intuitively, the more similar the two sets are, the better the accuracy of the system.

4.5.3 Extraction Rules

Extraction rules are used to detect specific regularities and patterns in the text. To define such rules, specific declarative languages can be used [139, 140, 141]. To develop our IE system we used the Jape language [141]. Jape provides mechanisms to recognize regular expressions in annotations made over a document. For example, it allows to create rules that annotate as a *Person* a word that starts with a capital letter, and that is preceded by the word *Mr*, *Mrs* or *Miss*. Jape's extraction rules consist of two components: a Left Hand Side (LHS), representing the condition, expressed in form of pattern, and a Right Hand Side (RHS), representing the conclusion, i.e. the action to take once the pattern matches. The main idea behind the development of our IE system is that privacy policies share a set of fixed patterns. Table 4.1 presents the patterns we detected during the analysis of the privacy policies of our corpus. In the table, each pattern is expressed as a sequence of *NE annotations*, but also *Token* and *POS annotations* (like *MODAL*, *SUCH_AS*, *TO*, *INCLUDE*), created by earlier steps of the process (see Figure 4.2). For each pattern in the figure, we present one or more

Table 4.1: *Collection* Patterns in our Corpus.

	Pattern
n.1	**DATA_COLLECTOR** *MODAL* **COLLECTOR_ACTION DATA_PROVIDER** *TO* **PROVIDER_ACTION** *your* **GENERIC_DATA** (*SUCH_AS*)? (**SPECIFIC_DATA**)*.
Ex. 1	*We may ask you to provide your personal data such as name, surname, and gender.*
Ex. 2	*The website may request you to provide your financial information.*
n.2	**DATA_COLLECTOR** *MODAL* **COLLECTOR_ACTION** *your* **GENERIC_DATA** (*SUCH_AS*)? (**SPECIFIC_DATA**)*.
Ex. 1	*We may collect your personal data such as name, surname, and gender.*
Ex. 2	*Google may gather your contact information.*
n.3	**DATA_PROVIDER** *MODAL* **PROVIDER_ACTION** *us with your* **GENERIC_DATA** (*SUCH_AS*)? (**SPECIFIC_DATA**)*.
Ex. 1	*You must provide us with your personal data including your name, surname, and gender.*
Ex. 2	*The user should supply his personal information.*
n.4	*The* **GENERIC_DATA DATA_COLLECTOR COLLECTOR_ACTION** *MODAL INCLUDE your* (**SPECIFIC_DATA**)*.
Ex. 1	The personal data we collect may include your name, surname, and gender.
n.5	**COLLECTION_TOOLS** *MODAL BE_USED TO* **TOOLS_ACTION** *your* **GENERIC_DATA** (*SUCH_AS*)? (**SPECIFIC_DATA**)*.
Ex. 1	*Cookies may be used to track your browsing data such as your IP address, browser type and pages visited.*
Ex. 2	*Web beacons may be used to monitor your browsing activities.*

?: zero or one repetition
*: zero or more repetition
italic: terms not relevant for the pattern
ITALIC: temporary annotations
BOLD: named entities

examples of matching sentences. For instance, the pattern n.1 of the table will match every sentence where the website is asking the user to provide personal information.

Once patterns have been defined (LHS), it is necessary to create the output (RHS). Figure 4.6 describes how a pattern can be translated in a Jape rule, by referring to the first pattern of Table 4.1. The LHS describes the pattern while, when a match occurs, the RHS is called to annotate *specific* and *generic data* as PDC. Note that in Jape the symbols ?, * and + means respectively zero or one, zero or more, and one or more occurrences. For each pattern of the table, one or more rules have been created. We created rules accounting for positive and negative (‘*we collect*’ versus ‘*we do*

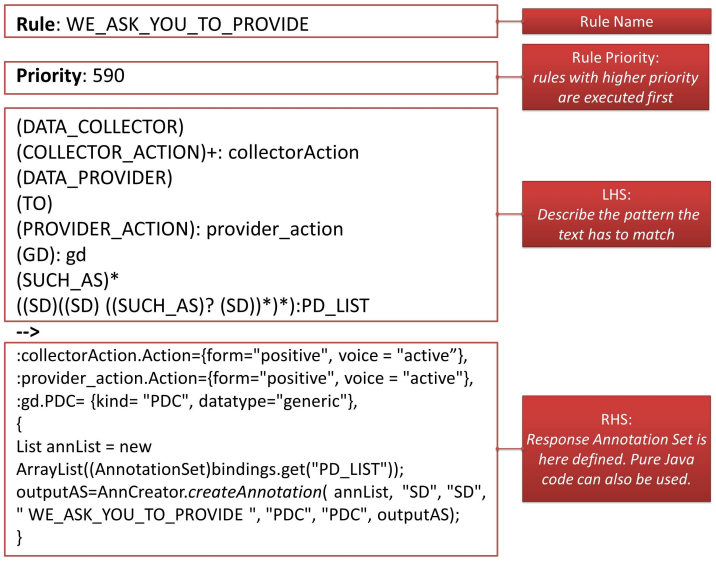


Figure 4.6: Example of JAPE Extraction Rule.

not collect'), and for active and passive ('*we collect your data*' versus '*your data is collected*') forms of the patterns. Clearly, a negative pattern indicates the specified personal data is not collected, so it is not added to the *Response Set*. Moreover, rules have been created in such a way that tokens not relevant for the pattern are ignored (e.g. in a sentences like '*we ask you to provide your personal data*', the presence of the token *your* does not stop the rule to match).

4.6 Evaluation and Results

In this section we discuss the methodology used to evaluate the accuracy of the IE we developed, and the results we obtained. We evaluate our IE system by measuring the recall, the precision, and the F_1 score obtained by comparing the *Response Set* to the *Standard Gold Set*. The GATE framework automatize the computations of such metrics. In addition, GATE allows to compute recall and precision using *strict* or *lenient* modes. Such modes differ for the way partial matches are treated. A partial match occurs when the result of the *Response Set* does not completely match the one of the *Standard Gold Set*. For example, in case the manual annotation says that "*e-mail address**es*** = *PDC*" but the *Response Set* only recognizes "*e-mail address*=

Table 4.2: System Accuracy for Corpus A and B.

	Time(sec. per set)		**Precision**		**Recall**		**F_1 score**	
	A	B	A	B	A	B	A	B
Training Set	3.22	9.25	76%	75%	90%	83%	83%	79%
Testing Set	1.28	3.25	78%	80%	81%	87%	**80%**	**83%**

PDC". In such a case, the *strict* mode would consider the result as a mistake, while the *lenient* mode would treat it as a correct result. In our context, partial matches cane be considered correct, so we use the *lenient* mode when measuring precision and recall.

The process of creating the extraction rules of an IE system is iterative and it involves several tests over the training set. Hence, the evaluation of the accuracy takes place during the whole development phase. The results of this evaluation serve to indicate whether the creation of new rules, or the modification of existing ones, is needed.

Besides rules, gazetteer lists are also modified during the development process: privacy policies are analyzed and, when found, new NEs instances are added to the lists. For example, when the *Facebook* policy is analyzed, the terms *profile* and *profile picture* are added to the *Specific Data* gazetteer list and *Facebook* is added to the *Data Collector* list. We noticed that, when the last policies of the corpus are reached, only few new NEs instances are discovered. This suggests that, although gazetteer lists can still be extended, they have already reached a good level of completeness. However, adopting techniques such as collaborative filtering, or applying smart heuristics to extract new *Data Collectors* once encountered (e.g. at the beginning of the parsing of a new website) could further improve our system.

The IE system can be considered complete when when the accuracy of the system (over the training set) is satisfactory and no new patterns are detected. Table 4.2 shows precision, recall and F_1 score obtained by running the best configuration of our IE system over the training and testing sets of the corpus A and B. This configuration uses the Annie implementation of the modules shown in Figure 4.2. The overall accuracy we reached, captured by the F_1 score obtained over the testing set, is 80% for corpus A, and 83% for corpus B. To avoid biased results, the accuracy is evaluated considering the results over the *testing* sets of corpus A and B, since the extraction rules have been created without any knowledge of the documents in such sets.

We believe these results are very promising. If we consider that average users do not read privacy policies, we increase their knowledge on data collection from 0 to 80%. Also, the results we are discussing are obtained by using a very basic pipeline,

and a limited set of rules. This means the accuracy can still be improved, e.g. by adding new modules (such as co-reference or ontologies) or by extending the rule set. Finally, note that the results presented in Table 4.2 do not take repetition into consideration, i.e., if a personal data item (e.g. *name*, *e-mail*) appears in two sentences, but it is only detected once, we have one correct result and one wrong result, leading to an accuracy of 50%. In our context, the fact that at least one occurrence of the same data item is correctly identified suffices to have 100% accuracy (for that specific item). By taking this into account, the accuracy of our system would raise from 79% to 92% over the training set of corpus B.

The main bottleneck for our system is the accuracy of the POS tagger module. Most of the errors in the *Response Set* are due to erroneous results provided by this module in our experimental tests. The POS tagger is in charge of distinguishing whether a word represents a verb, a noun or another lexical category. That means it should be able to say that the word '*place*' represents a noun in the sentence "*we collect information about the place you took your photo*", while it represents a verb in the sentence "*we store your mail address when you place an order*". Since our rules are based on the POS tagger annotations (e.g. *provide* is considered a PROVIDER_ACTION if and only if it has been annotated as a *verb* by the POS tagger), an error of the tagger will result in an error of our system. For example, let us consider the following sentence: "*LinkedIn* ***requests*** *other information from you during the registration process, (e.g. gender, location, etc.)*". If '*requests*' is not recognized as a verb by the POS tagger, no extraction rule will fire, and the words *gender* and *location* will not be annotated as PDC. That means the accuracy of our system strongly depends on the accuracy of the POS tagger, and that it can be improved by decreasing the tagger error rate.

Since multiple implementations of the POS tagger exist, we tested how the accuracy of the system changes according to which tagger we use. The results of this comparison, running the system over the testing sets, are shown in Table 4.3. We used three different POS tagger implementation: the OpenNLP, the LingPipe[3], and the Annie POS tagger. As we can observe, the use of the LingPipe tagger leads to the worst performance over the corpus A, with a difference of 5% when compared with the best results, obtained by using Annie tagger. The results confirm that the choice of the POS tagger has a considerable impact on the performance of the system. Although Annie is the best performing tagger in our setting, it still shows several errors. A possible way to improve the POS performances, and with it our system's accuracy, is by training a POS tagger over the privacy policy domain.

Finally, note that our solution, which was not optimized for time performances, can be executed in near real-time. As reported in the *Time* column of Table 4.2, the

[3]For the LingPipe the *pos-en-general-brown model* was used.

Table 4.3: Performance Comparison of Different POS Taggers.

POS Tagger	**Time(sec)**		**Precision**		**Recall**		F_1 **score**	
	A	B	A	B	A	B	A	B
OpenNLP	1.55	4.50	77%	70%	80%	83%	78%	77%
LingPipe	2.45	3.95	79%	75%	68%	84%	73%	79%
Annie	1.02	3.25	78%	80%	81%	87%	**80%**	**83%**

system needs 9.25 seconds to process 9 complete privacy policies (training set of B), therefore the analysis of a single policy requires around one second. This means our approach is applicable to real scenarios, where providing real time responses to the user is an important requirement.

4.7 Limitations

The framework we proposed in this chapter shows promising results w.r.t. automatic extraction of semantic information from privacy policies. However, the performance of the system can reasonably improve by defining additional extraction patterns. Unfortunately, an extended set of experiments which proves this intuition, e.g. by measuring the changes in detection rate while increasing the number of extraction patterns, is lacking in our research. This limitation should be addressed by extending the experiment set. In addition, note that the classes used to categorize personal data items could be improved. In the current solution, we divide Personal Data Collected into two classes: Generic Data and Specific Data. Clearly, a finer-grain classification could have been done. For example, one could include Optional Data as well, this being information that a web site only optionally requests, and which the user can decide to present or not (such as home phone number or mobile phone number or both). In this case, in a sentence like "If you do not want to receive e-mail from us, please adjust your preferences", the term *e-mail* and the term *preferences* would have been classified as optional PDC rather than being ignored as in the current solution. Finally, as already mentioned in Chapter 3.3, another limitation of our solution is given by the fact that only English language policies have been considered.

4.8 Conclusions

In this chapter, we discussed the creation of an IE system, able to extract the list of personal data collected by a website by analyzing the content of its privacy policy. The system shows an accuracy around 80%. Although we believe this accuracy is reasonable and able to benefit the users, improvements can still be made. For example, by adding co-reference and anaphora resolution modules, together with ontologies and thesaurus to enrich the gazetteer lists with synonyms and close lexical concepts. On the other hand, we believe the system would not benefit of the use of syntactic parsing, since its computational costs would not allow to provide real time responses to the users [4].

[4] Preliminary tests show the Stanford Parser needs 123 seconds to process the Google privacy policy.

Chapter 5

Selecting Web Services: Privacy Aware Service Composition

In Chapter 3 and Chapter 4 we evaluated the privacy protection offered by a website, by solely looking at its privacy policy. This approach has the advantage of not requiring the support of the service provider to make privacy assessment and, as such, it can be easily adopted. However, to attract privacy-concerned users, service providers should allow them to express privacy preferences and should guarantee that such preferences are respected during the service provision. In this chapter, we propose a solution for privacy that involves both parties, end-users and service providers. Although, as we have mentioned earlier, solutions that requires server-side adoption are more difficult to spread, privacy-conscious service providers require the availability of privacy solutions they can adopt to address consumers privacy concerns. Existing solutions as P3P [3] have been developed for individual websites and not for composite web services. In this chapter, we focus our attention on the service oriented architecture (SOA) where business use-cases are composed of several web services. We propose a solution to create service compositions which account for user's privacy preferences. The main contributions of our proposal are the following: i) a fine-grained model to express web service providers' privacy policies and users' privacy preferences; ii) an algorithm which creates web service composition satisfying users' privacy preferences; and iii) a ranking mechanism which evaluates compositions w.r.t. the level of privacy they offer.

5.1 Introduction

Many of the services available on the Internet are built as a composition of web services from different providers and organizations. Web service composition enhance the provision of highly personalized services which are tailored to the users' needs. Web services are well defined software components that can be advertised, located, and composed over the Internet by using standards like WSDL, UDDI and BPEL, respectively. Typically, there are multiple web services available over the web that implement the same task. Service composition consists of joining, at runtime, several services that satisfy users' functional and non-functional requirements. Due to the increasing number of available services which offer similar functionalities, it is hard for users to select an optimal service composition among a list of candidate services that satisfy their needs. Therefore, service selection is a key challenge in the Future Internet.

The problem of web service composition and selection has been widely addressed in the literature. Most of the existing approaches focus on the identification of optimal web services among a set of candidates based on constraints on the Quality of Service (QoS) [142, 143, 144, 145, 146, 147], or on their trust and reputation level [148, 149, 150, 151]. Despite the fact that privacy plays a major role in the matter, only few works have investigated privacy issues in service selection [152, 153] and composition [154, 155, 156]. The orchestrator, namely the service in charge of the composition, usually collects a large amount of personal data about its users and it will eventually share such data with the service providers involved in the composition. This exchange of data may lead to privacy risks, e.g. in case a service provider uses personal data for unauthorized purposes. As a consequence, the privacy practices adopted by service providers should represent an important factor to drive service selection: users will more likely use services that take into account their privacy preferences.

In this chapter, we propose an approach to assist both users and web service providers in composing and selecting optimal services with respect to their privacy preferences. We use AND/OR trees to represent the orchestration schema, component services and their privacy policies. Based on this representation, we present an algorithm that determines the web service compositions compliant with user privacy preferences. To help users to select the best web service composition, our approach ranks admissible composite web services (i.e., composite services whose privacy policy satisfy user preferences) with respect to their *privacy level*. The privacy level quantifies the risk of misuse of personal data based on three dimensions: the sensitivity, visibility and retention period of information.

The contribution of this chapter is three-fold. First, we propose a fine-grained model to express web service providers' privacy policies and users' privacy prefer-

ences based on several privacy dimensions – sensitivity, purpose, retention period, visibility – while other approaches to privacy-aware service composition only consider one dimension, e.g. sensitivity or visibility. Second, we propose a web service composition algorithm which merges into a single step the selection of services that satisfy users' functional requirements and the selection of services compliant with users' privacy requirements, while most existing approaches execute these two steps separately. Last but not least, we rank composite services with respect to the level of privacy they offer, while other approaches only focus on the generation of a privacy-preserving composition. We illustrate our privacy-aware composition and selection process using a travel agency web service as a running example.

The remainder of the chapter is structured as follows. Section 5.2 discusses related work. Section 5.3 presents a modeling framework for representing service orchestrations, users' privacy preferences and web service providers' privacy policies. Section 5.4 presents the privacy-aware service composition and selection process. Finally, Section 5.7 concludes the chapter providing directions for future work.

5.2 Related Work

Our work is related to the fields of *service composition modeling*, *service composition*, and *service selection*.

Service composition modeling To model service composition and verify whether it satisfies properties like safety and liveness, several languages, such as WS-BPEL [157], or approaches, such as process algebra [158], Petri nets [159], model checking [160], and finite state machines [161], have been proposed. Contributions to service composition modeling also come from the requirement engineering community, where goal-oriented approaches [162, 163] are used to represent strategic business goals. Similarly, we adopt a goal-oriented approach to model service composition. The advantage of such an approach is that it provides the abstraction necessary to represent privacy policies without getting bogged down into the functioning of web services.

Service composition Service composition is the problem of aggregating services in such a way that given (functional and not functional) requirements are satisfied. The role of privacy in service composition has been investigated in [155], where only services requiring the disclosure of less sensitive information and offered by trusted providers are selected in the composition. Users' privacy concerns are often addressed by providing automated techniques for matching provider's privacy policies with customer's preferences [154, 156, 3, 164, 165]. As we have seen in the previous chapters, on of the most prominent solution for policy matching is P3P [3],

that aims at assisting service providers in specifying their privacy practices on the Web, and users in matching such practices against their preferences. To automate the matching process, P3P has been complemented with privacy preferences languages such as APPEL [166] and XPref [167]. In [164] service composition is the result of a negotiation phase between user privacy preferences (describing the type of access to each piece of personal information) and the web service policy statement (specifying which information is mandatory and which is optional to use a service). Here, the outcome of the negotiation indicates what personal information the user should disclose to the service provider. However, these techniques only focus on the relation between a server and a client. In contrast, our work uses a privacy policy matching approach to build the model of admissible service compositions. In addition, our work goes beyond pure service composition: we also identify the most privacy preserving composition.

Service selection Service composition might return a set of admissible services; thus, service ranking is needed to choose the *best* composition. QoS-based [142, 143, 144, 145, 146, 147] and trust-based [148, 149, 150, 151] service selection has been widely investigated in the literature. Privacy-aware service selection is addressed in [153] which presents a comprehensive framework to protect users' and service providers' privacy needs at selection time. Users' criteria are matched against web services' attributes in a private fashion such that both criteria and service attributes are kept private. This approach mainly focuses on protection of service provision rules from unwanted disclosure, while our goal is to select the most privacy preserving composition. Massacci et al. [152] present an approach to service selection based on the sensitivity of data to be disclosed for the service provision. In contrast, we consider a number of criteria characterizing privacy policy and user preference for selecting the optimal service composition. Similar criteria are also considered in [168]. However, these criteria are not used to assess the privacy level of services. Rather, they are used to capture discrepancies between what was stated in privacy policies and what is done in practice. To allow service ranking, we aggregate the identified criteria using an approach based on the norm. Although more complex solutions like swap [169] or collaborative filtering [170] have been proposed to assist users in multi-criteria decision making, such solutions either require a high level of user interactions and thus cannot be automated, or are not applicable due to the nature of privacy criteria.

5.3 Modeling Service Composition and Privacy

In this section we introduce the models to represent web service orchestration, privacy policies and user privacy preferences on which our approach is based.

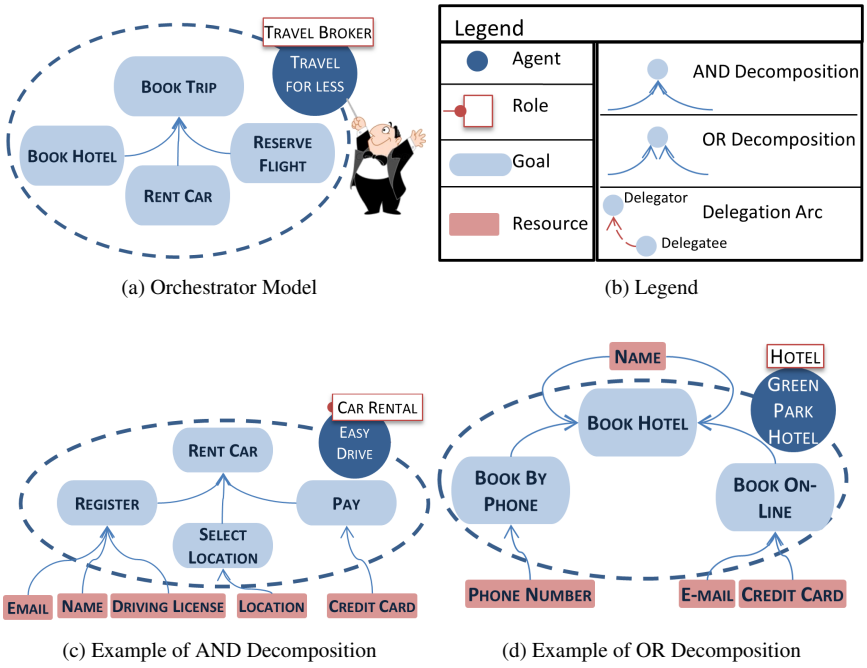
5.3.1 Modeling Service Orchestration

In web services composition typically there is an *orchestrator* which combines the functionalities provided by other services usually denoted as *component services* to satisfy users' requests. Several services may be able to provide the same functionality requested by the user. The service resulting from the orchestration is called *composite service*. We model the composition schema as an *orchestrator model*, each component service as a *component service model*, and all possible alternative instantiations of the schema as a *service orchestration model*.

We represent these models as AND/OR trees where the semantics of nodes and arcs according to the concepts defined by SI* [171], a goal-oriented framework for requirements elicitation and analysis. SI* employs the notions of *actor*, *goal*, *resource*, *decomposition* and *delegation*. *Actors* are active entities that have strategic goals and perform actions to achieve them. Actors can be *agents* or *roles*: agents are used to represent the orchestrator and component services, and roles to represent the types of services. The sets of agents and roles are denoted A and T respectively, with $A \cap T = \emptyset$. We use notation $s \rhd t$ to indicate that a service $s \in A$ is of type $t \in T$. *Goals* represent the functionalities offered by services, while *resources* represent data produced/consumed by a goal. The sets of goals and resources are denoted G and R, respectively. *Decomposition* is used to refine a goal: AND decomposition refines a goal into subgoals and resources needed to achieve the goal, while OR decomposition defines alternatives to achieve a goal. *Delegation* marks a formal passage of responsibility or authority from an actor (*delegator*) to another actor (*delegatee*) to achieve a goal. We use these concepts to define the notion of *service model*.

Definition 8 (Service Model) *A* service model S*, according to [171], is a pair* $\langle V, E \rangle$ *where:* $V = G \cup R$ *is the set of nodes;* E *is the set of decomposition arcs* $\langle Z, g \rangle$ *connecting a node* $g \in G$ *to a non-empty set* $Z \subseteq V$*.*

Example 2 TravelForLess *is a composite web service which provides its customers deals including hotel reservations, flight bookings, car rentals, or any combination of these travel options.To this end,* TravelForLess *relies on partner travel agencies, hotels chains, airline companies and car rental agencies that are dynamically selected. Figure 5.1a illustrates the orchestrator model of* TravelForLess*, while Figure 5.1b shows the list of symbols used through our examples and their meaning.*



(a) Orchestrator Model (b) Legend

(c) Example of AND Decomposition (d) Example of OR Decomposition

Figure 5.1: Examples of our Modeling

TravelForLess *provides goal* book trip*, represented by the top oval. This goal is decomposed into sub-goals* book hotel*,* reserve flight *and* rent car*. Figs. 5.1c and 5.1d illustrate examples of component services of types* Car Rental Agency *and* Hotel*. Note that in case a resource is shared by two sub-goals in an OR composition, e.g.* Name *in Figure 5.1d, the arrows will depart from the super-goal (*book hotel*) rather the the sub-goals.* □

The service orchestration model is obtained by merging the service models associated with the orchestrator and all component services. In particular, we merge the service model of the outsourcer with the service model of the subcontractor by linking the goal of the former with the corresponding goal (with the same name) occurring in the service model associated with the latter. Intuitively, goals with the same name represent the same functionality and, therefore, can be considered equivalent (although they may require different data items or can be decomposed differently). Let S_1 and S_2 be two service models. We write $n_1 \equiv n_2$ to denote that $n_1 \in S_1$ and $n_2 \in S_2$

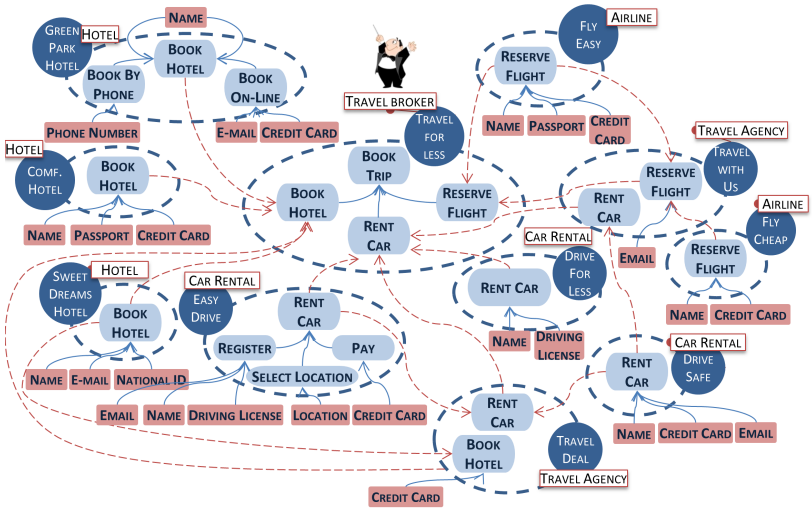


Figure 5.2: Example of Orchestration Model

are equivalent. Arcs linking nodes across service models are called *delegation arcs*. If more than one component service can fulfill the goal, each such component service is linked to the goal of the outsourcer. Notice that a component service may not have the capabilities to fully achieve a goal. In this case, the component service may redelegate the achievement of (part of) the goal to another component service.

Example 3 *Figure 5.2 shows the orchestration model obtained by merging the service model of* TravelForLess *with the ones of the candidate component services. In the figure, delegation arcs are represented as dashed arrows. The model represents all possible alternatives to fulfill the goals of* TravelForLess. *Goal* rent car *can be provided by two car rentals,* DriveForLess *and* EasyDrive, *and by two travel agencies,* TravelDeal *and* TravelWithUs. *Goal* book hotel *can be fulfilled by providers* SweetDreamsHotel, GreenParkHotel, Comfort Hotel *and* TravelDeal *agency, while goal* reserve flight *is delegated to airline company* FlyEasy *and to travel agency* TravelWithUs. *The partner services may require different information to fulfill the goal they provide. For example, to fulfill goal* book hotel, SweetDreamsHotel *requires its customers to provide* name, email, credit card *and* national ID, *while* name, passport, *and* credit card *are the data items required by* Comfort Hotel. □

A composite service is a particular sub-tree of the service orchestration model which represents a possible alternative to fulfill its root goal. Before formally defining

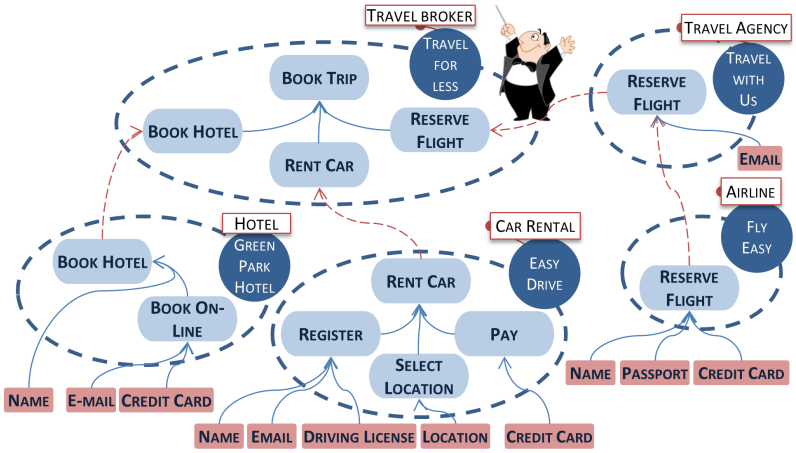


Figure 5.3: Example of Composite Service

a composite service, we introduce the notion of *decomposition path*.

Definition 9 (Decomposition Path) *Let $S = \langle V, E \rangle$ be a service orchestration model, $Z \subseteq V$ be a non-empty set of goals and resources, and g be a goal in V. A* decomposition path *$D_{Z,g}$ is a set of arcs $E' \subset E$ such that either $g \in Z$, or there exists a decomposition arc $\langle T, g \rangle \in E'$ and there are decomposition paths $D_{Z,x} \in E'$ for each $x \in T$.*

Definition 10 (Composite service) *Let $S = \langle V, E \rangle$ be a service orchestration model. A* composite service *is a decomposition path D_{Z,g_0} such that $Z \subseteq V \cap R$ is a set of data items and g_0 is the root goal of S.*

Example 4 *Figure 5.3 shows a possible composite service within the orchestration model of Figure 5.2 where* Travel ForLess*'s goals,* book hotel, rent car, *and* reserve flight *are fulfilled by the services* GreenParkHotel, EasyDrive, *and* TravelWithUs, *respectively. In turn,* TravelWithUs *delegates the fulfillment of* reserve flight *to* FlyEasy. □

5.3.2 Modeling Privacy

To complete the interaction with a web service (composite or simple), the user has to disclose her personal information to the service. However, users may be concerned

about disclosing their personal data. Data protection legislation aims to address these concerns by recognizing the users' right to control their data [172]. To this end, users should be allowed to define *privacy preferences* which specify constraints on the collection and processing of their data. Also, web service providers (both the orchestrator and component services) are obliged by law to publish *privacy policies* in which their privacy practices are declared. Privacy policies should specify the purposes for which data is collected and processed. In addition, privacy policies should specify to whom the data is disclosed for achieving a given purpose, and how long the data can be retained for that purpose. Looking at the policy, users should be able to understand how their personal data will be used and, in case they agree, disclose them.

To model privacy policies and privacy preferences, we consider the following privacy dimensions: *purpose* defines the reason(s) for data collection and usage; *visibility* defines to whom data can be disclosed; *retention period* defines how long data can be maintained; *sensitivity* represents the data subject's perception of the harm the misuse of her data can cause to her. Since we assume that sensitivity is different for different users, this dimension is only used to specify privacy preferences, hence it does not appear in privacy policies. Based on these privacy dimensions, we formally define privacy policies as follows.

Definition 11 (Privacy Policy) *A privacy policy is a set of tuples* $\langle d, p, \nu, \delta, \tau \rangle$ *where:* $d \in R$ *denotes a data item;* $p \in G$ *is the purpose for which* d *can be collected;* $\nu \subset A \cup T$ *is the visibility of* d *for achieving* p*;* $\delta \in \mathbb{N} \cup \{*\}$ *represents the (re)delegation depth which is used to limit the sharing of* d *for achieving* p *(Depth* 1 *means that no further sharing is allowed,* n *means that* $n-1$ *further steps are allowed, and depth "*" means unlimited sharing);* $\tau \in \Re$ *represents the retention period (here in months) of* d *for achieving* p*.*

Although the notation introduced in Definition 11 makes it possible to capture the privacy dimensions necessary to specify privacy policies, it makes it difficult to understand and reason on the specified privacy policies. To this end, we represent privacy policies as AND/OR trees where nodes model the purposes in the policy's tuples and data items protected by the policy. This representation resembles the service model. Figure 5.4 shows the privacy policy for GreenParkHotel and its graphic representation using AND/OR trees. The main difference between the two models is that nodes are annotated with visibility, (re)delegation depth and retention period. Formally, a *privacy policy model* is a tuple $\langle V, E, \Gamma \rangle$ where $\langle V, E \rangle$ is the corresponding service model and Γ is the privacy policy in tabular form. Given a privacy policy Γ and a purpose p, $\Gamma^p = \{\langle d, \nu, \delta, \tau \rangle | \langle d, p, \nu, \delta, \tau \rangle \in \Gamma\}$. We say that a privacy policy Γ is *well-defined* if (i) $\forall \langle d, p, \nu, \delta, \tau \rangle \in \Gamma\ \nu \neq \emptyset$ iff $\delta > 1$ and (ii) for every data item d and purpose p such that $\langle d, \nu, \delta, \tau \rangle \in \Gamma^p$, $\nexists \langle d', \nu', \delta', \tau' \rangle \in \Gamma^{p'}$ with p' sub-purpose

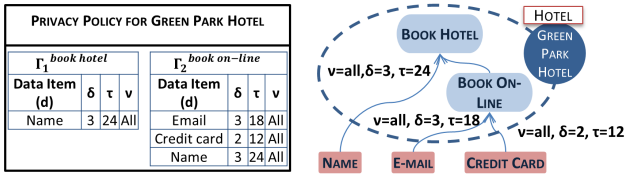


Figure 5.4: Privacy policy for GreenParkHotel and its graphic representation.

of p and $d = d'$. Intuitively, the first condition states that the visibility can be defined if and only if the delegation depth is greater than 1, and the second imposes that the privacy policy for a data item is not redefined during policy refinement. In this chapter, we only consider well-defined privacy policies.

To compare the privacy policy of different services, we introduce the notion of *policy compliance*. We say that the privacy policy of a service complies with the privacy policy of another service if the former is more restrictive than the latter. Policy compliance is formally defined as follows.

Definition 12 (Policy compliance) *Let Γ_x and Γ_y be two well-defined privacy policies. Γ_y complies with Γ_x, denoted as $\Gamma_y \rightsquigarrow \Gamma_x$, if $\forall p\ \forall \langle d_1, \nu_1, \delta_1, \tau_1 \rangle \in \Gamma_y^p\ \exists \langle d_2, \nu_2, \delta_2, \tau_2 \rangle \in \Gamma_x^p$ such that (i) $d_1 = d_2$; (ii) $\delta_1 < \delta_2$; (iii) $\tau_1 \leq \tau_2$.*

Example 5 *The privacy policy of* TravelForLess *is presented in Table 5.1. The policy specifies how* TravelForLess *will use customers' data. For example,* TravelForLess *will collect a customer's* name *to fulfill purpose* rent car *and it will maintain a copy of the data item for 36 months. Moreover, the policy states that* TravelForLess *can disclose customers'* name *to services of any type (which is denoted by the keyword "*All*"). Since the depth is set to *, any service receiving directly or indirectly a copy of* name *can further share it with no limitation. Customers'* national ID *can be collected only for purpose* reserve flight *and has different rules for different agents: if the component service is an instance of* Travel Agency *the* national ID *can be shared with other services and can be stored up to 18 months; in the case the component service is an instance of* Airline*, the* national ID *cannot be delegated further*[1]*, and can be kept only for 12 months.* □

When interacting with the orchestrator, a user should analyze its policy and decide whether it is acceptable. The user can refine the policy by limiting the requested functionalities and restricting the use of data items. In particular, she can restrict the

[1]Note that this specific example is used for demonstrative purposes. In reality, ID information is often transmitted to governments before passenger can fly.

Table 5.1: TravelForLess's Privacy Policy

Data item (d)	Purpose (p)	Visibility (ν)	Depth (δ)	Retention (τ)
Name	Rent Car	All	*	36
	Book Hotel	All	*	24
	Reserve Flight	All	*	24
Email	Rent Car	All	*	24
	Book Hotel	All	*	24
	Reserve Flight	All	*	36
Credit Card	Rent Car	Car Rental, Travel Agency	3	12
	Book Hotel	Hotel, Travel Agency	2	12
	Reserve Flight	Airline	2	12
Passport	Reserve Flight	Travel Agency, Airline	2	12
Driving License	Rent Car	Travel Agency, Car Rental	3	12
Phone Number	Rent Car	All	*	36
	Book Hotel	All	*	36
	Reserve Flight	All	*	36
National ID	Reserve Flight	Travel Agency	3	18
		Airline	2	12
	Book Hotel	Travel Agency, Hotel	3	12
Location	Rent Car	Car Rental	2	12

visibility of a certain data item by denying sharing it with a certain type of service or selecting specific component services. In addition, the user may decide to not disclose a certain data item. Finally, the user should define the sensitivity of each data item, which may vary from purpose to purpose. Users, however, are not allowed to change the delegation depth and retention period. This is because these attributes are often constrained by the business model of the orchestrator as well as by the requirements imposed by the legal framework in force (e.g., telecommunications data have to be stored for six to 24 months according to the EU Directive on data retention). The result of this refinement process represents the privacy preferences of the user. We formally specify users' privacy preferences as follows.

Definition 13 (Privacy Preferences) *The privacy preferences of a user are a set of tuples* $\langle d, p, \sigma, \nu, \delta, \tau \rangle$ *where:* $d \in R$ *denotes a data item;* $p \in G$ *is the purpose for which* d *can be collected;* $\sigma \in [1, 10]$ *is the sensitivity of* d*;* $\nu \subset A \cup T$ *is the visibility of* d *for achieving* p*;* $\delta \in \mathcal{N} \cup \{*\}$ *is the (re)delegation depth which limits the sharing of* d *for achieving* p*;* $\tau \in \Re$ *is the retention period of* d*.*

Example 6 *Let Bob be a new customer of* TravelForLess*. He wants to book a trip to Barcelona but, since he is afraid to fly, he only wants to book a hotel and rent a car.*

Table 5.2: Bob's Privacy Preferences

Data item (d)	Sensitivity (σ)	Purpose (p)	Visibility (ν)	Depth (δ)	Retention (τ)
Name	5	Rent Car	All	*	36
	5	Book Hotel	All	*	24
Email	5	Rent Car	All	*	24
	5	Book Hotel	All	*	24
Credit Card	10	Rent Car	Travel With Us	3	12
	8	Book Hotel	Travel Data, Green Park Hotel	2	12
Driving License	9	Rent Car	Drive For Less	3	12
National ID	6	Book Hotel	Travel Agency, Hotel	3	12

Based on the privacy policy of TravelForLess *(Table 5.1), he specifies constraints on the collection and processing of his data. Bob's privacy preferences are presented in Table 5.2. Since* name *and* email *are usually required by service providers, Bob leaves their visibility to* all. *In contrast, he prefers that his* credit card *is only disclosed to agents he trusts (as introduced in Section 5.3 there is always a mapping between agents and roles), i.e.* TravelWithUs, TravelDeal *and* GreenParkHotel. *Bob also restricts the access to his* driving license *only for the purpose of renting a car, and the* national ID *only for booking a hotel. Finally, Bob prefers to be contacted by* email *and thus he is not willing to disclose his* phone number. □

5.4 Privacy-aware Service Selection

In what follows we describe in details the operations performed by the service compositions and the composite service ranking components.

5.4.1 Service Composition

Service orchestrators usually do not provide the functionalities required by a client directly but they outsource the provision to specialized services. Nonetheless, according to the EU privacy regulation, they are liable for the actions performed by the subcontractors. Therefore, an orchestrator is willing to select a component service only if the privacy policy of the component service complies with its policy and user privacy preferences. The aim of the service orchestration composition step is to identify *admissible composite services*, i.e. those composite services that comply with the user preferences and legal requirements.

After a user has defined her privacy preferences through the refinement of the orchestrator's privacy policy (see Example 6), the orchestrator uses those preferences

Algorithm 1: Service Composition

Input: S_u set of functionalities requested by user u, $P_o = \langle V_o, E_o, \Gamma_o \rangle$ privacy policy model of the orchestrator augmented with the privacy preferences of u, $\mathcal{P}$ set of the privacy policy models of component services

Output: P privacy policy model of the service orchestration

let $P = \langle V, E, \Gamma \rangle$;
let $V = \{root\}, E = \emptyset, \Gamma = \emptyset$;
make Q empty ; `//Q is a queue containing the nodes to be visited`
make S empty ; `//S is a queue containing pairs of nodes where the first element represents the reference node and the second represents the node to be visited`
for $s \in S_u$ **do**
    $V = V \cup \{s\}$;
    $\Gamma^s = \Gamma^s_o$;
    insert s in Q;
$E = E \cup \{\langle S_u, root \rangle\}$;
while Q *is not empty* **do**
    extract s_i from Q;
    if s_i *not leaf node* **then**
        for $\langle Z, s_i \rangle \in E_o$ **do**
            $V = V \cup Z$;
            $E = E \cup \{\langle Z, s_i \rangle\}$;
            for $s_j \in Z$ **do**
                $\Gamma^{s_j} = \Gamma^{s_i} \cup \Gamma^{s_j}_o$;
                insert s_j in Q;
    else
        insert (s_i, s_i) in S
while S *is not empty* **do**
    extract (s_k, s_i) from S;
    let $P_x = \langle V_x, E_x, \Gamma_x \rangle$ be the policy model s.t. $s_i \in V_x$;
    if s_i *not leaf node* **then**
        for $\langle Z, s_i \rangle \in E_x$ **do**
            if $\Gamma^Z_x \rightsquigarrow \Gamma^{s_k}$ **then**
                $V = V \cup Z$;
                $E = E \cup \{\langle Z, s_i \rangle\}$;
                for $s_j \in Z$ **do**
                    $\Gamma^{s_j} = \Gamma^{s_i} \cup \Gamma^{s_j}_x$;
                    insert (s_k, s_j) in S;
    else if s_i *is a purpose node* **then**
        let $W = \{w | \langle d, \nu, \delta, \tau \rangle \in \Gamma^{s_i}_x \wedge ((w \in \nu \cap A) \vee (w \rhd t \wedge t \in \nu \cap T))\}$;
        for $w \in W$ **do**
            let $P_w = \langle V_w, E_w, \Gamma_w \rangle$ be the policy model of w;
            if $\exists s_j \in V_w$ *s.t.* $s_j \equiv s_i$ **then**
                if $\Gamma^{s_j}_w \rightsquigarrow \Gamma^{s_k}$ **then**
                    $V = V \cup \{s_j\}$;
                    $E = E \cup \{\langle \{s_j\}, s_i \rangle\}$;
                    $\Gamma^{s_j} = \Gamma^{s_j}_w$;
                    insert (s_i, s_j) in S;

to identify admissible composite services. Admissible composite services are determined using Algorithm 1. The algorithm builds the privacy model of the service orchestration that includes only those component services whose privacy policy complies with the privacy preferences of the user (for the sake of simplicity, here we omit sensitivity in the user preferences, and represent them using the notation for privacy policies; sensitivity is used in the next step). The algorithm first identifies the portion of the policy model of the orchestrator related to the functionalities required by the user (lines 5-20). The policy associated with a purpose is propagated to sub-purposes (lines 16-17). Intuitively, a purpose inherits the constraints from the higher level purpose. This makes it possible to check the consistency of policies along the service orchestration model.

When the policy of the orchestrator is fully analyzed, the algorithm identifies the component services which offer the functionalities required by the user and whose privacy policy is compliant with the privacy policy of the service delegating the service to them (lines 21-41). If the node to be analyzed is not a leaf node of the policy (line 24), the algorithm checks whether the policy associated with the subnodes of that node complies with the policy associated with the leaf node in the policy of the service delegating the provisioning of the functionality (called *reference node*) (line 26). If it is compliant, the nodes are added to the policy model of the orchestration (lines 27-31).

If the node to be analyzed is a leaf node of the policy, the algorithm checks whether it is a purpose node (line 32). This case corresponds to situations in which the service does not have the capability to provide the functionality and outsources its provision to another service. Visibility is used to determine which component services should be considered in the orchestration (line 33). A component service in the visibility is considered by the algorithm if it actually offers the required functionality (line 36). If the policy associated with the new node complies with the policy of the outsourcer (line 37), the node together with a delegation arc is added to the policy model of the orchestration (lines 38-40).

The privacy policy model returned by Algorithm 1 corresponds to the privacy policy regulating the service orchestration. The composite services in the policy model of the service orchestration are the admissible composite services.

Proposition 1 *Let* Π *be the privacy preferences of a user and* P *the privacy policy model of the service orchestration obtained through Algorithm 1 w.r.t.* Π*. The privacy policy of every composite service* $p \in P$ *complies with* Π*.* □

The proof is by induction on the depth of the privacy policy model of the service orchestration. Notice that some composite services compliant with user privacy preferences may be discarted as compliance of the policy of a service is verified against

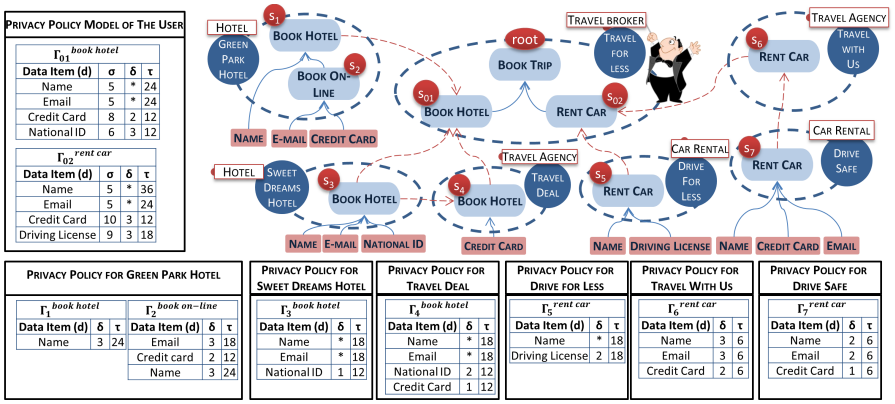


Figure 5.5: Example of Service Composition

the policy of the outsourcer (which may be more restrictive than user privacy preferences). This reflects the fact that, by law, the outsourcer is liable for the subcontractor. Therefore, a service would outsource (part of) its duties only to those services whose privacy practices are acceptable for it.

Example 7 *Figure 5.5 shows the orchestration policy model based on Bob's privacy preferences (Table 5.2) together with the policies of the selected component services. The model describes six admissible composite services that can be employed to provide the functionalities requested by Bob (see Table 5.3 for their description). Note that, for readability reasons, we have omitted the* visibility *field in the figure. However, the* visibility *values for the orchestrator and for Bob can be found, respectively, in Figure 5.1 and Figure 5.2, while we can assume the visibility is set to* all *for each data item of the other agents.* □

5.4.2 Composite Service Ranking

More than one composite service may satisfy a user's privacy preferences. In order to support the user in the decision making, we prioritize admissible composite services according to their privacy level. Intuitively, a composite service is more privacy-preserving if it requires the disclosure of less sensitive data as well as it retains data for less time and its constraints on their delegation are more restrictive.

To assess and compare the privacy level of admissible composite services, we represent their privacy policy in a three dimensional graph whose axes represent retention period, (re)delegation depth and sensitivity. In Definition 11 the privacy pol-

icy is defined as a set of tuples. The overall privacy level with respect to retention period and (re)delegation depth is obtained by aggregating the values of these dimensions in the tuples forming the policy of the composite service. Retention period and (re)delegation depth are weighted with respect to the sensitivity of the data item. This is to reflect the higher privacy risk of storing high sensitive data for a long time and potentially sharing them with more services. The sensitivity value associated with a composite service is given by the sum of the sensitivity of all data items that have to be shared for the execution of the component service. By summing up the sensitivity of all data items we assume that collecting 4 items of sensitivity 1 is equal to collect a single item of sensitivity 4. This is justified by the fact that combining several data items of low sensitivity could actually reveal the same (or more) information than a single data item which is highly sensitive [173, 174]. Finally, note that, although sensitivity is considered "twice", it has a different impact on the privacy level. While sensitivity as a dimension is used to measure the amount of information that needs to be disclosed by the user, sensitivity as a weight for retention period and (re)delegation depth is used to characterize the privacy risks associated with these two dimensions. We represent the privacy level of a composite service as a three dimensional vector.

Definition 14 (Privacy level) *Let Γ_0 be the orchestrator's privacy policy, $\Gamma_1, \ldots, \Gamma_n$ the privacy policies of component services, $P = \langle V, E, \Gamma \rangle$ the privacy policy model of a composite service, and Π the privacy preference of a user. Let $\overline{\Gamma} = \{\langle d, p, \nu, \delta, \tau \rangle$ s.t.$\langle d, p, \nu, \delta, \tau \rangle \in \Gamma \cap \Gamma_i\}$ be a set of tuples containing only the tuples that are relevant for the composite service Γ_i and does not contain duplicates. The privacy level of the composite service is a vector $[\delta, \tau, \sigma]$ such that*

- $\delta = avg\left(\sigma_j \delta_i | \langle d, p, \nu, \delta_i, \tau_i \rangle \in \overline{\Gamma} \wedge \langle d, p, \sigma_j, \nu, \delta_j, \tau_j \rangle \in \Pi\right)$
- $\tau = avg\left(\sigma_j \tau_i | \langle d, p, \nu, \delta_i, \tau_i \rangle \in \overline{\Gamma} \wedge \langle d, p, \sigma_j, \nu, \delta_j, \tau_j \rangle \in \Pi\right)$
- $\sigma = \sum_{\langle d, p_i, \nu_i, \delta_i, \tau_i \rangle \in \overline{\Gamma}} \sigma_j$ *s.t.* $\langle d, p_j, \sigma_j, \nu_j, \delta_j, \tau_j \rangle \in \Pi \wedge \nu_i \subset \nu_j \wedge (p_i = p_j \vee (\exists \langle p_j, Z \rangle \in E$ *s.t.* $p_i \in Z))$

Note that in Γ some tuples are duplicated because Algorithm 1 propagates them to sub-purposes, while the original policies $\Gamma_0, \Gamma_1, \ldots, \Gamma_n$ may contain tuples that are not applicable for the given composite service. The set of tuples $\overline{\Gamma}$ contains only the tuples that are relevant for the composite service and does not contain duplicates. Moreover, notice that every tuple in Γ has a counterpart in Π. If this is not the case, then the composite service is not admissible and thus we will not consider it at this stage.

The dimensions of privacy policies (namely *depth*, *retention* and *sensitivity*) range in different scales. To make them comparable, they need to be normalized. Also, when the (re)delegation depth is unlimited ($\delta = *$), we bound its value to 10 for the

Table 5.3: Admissible Composite Services

Composite Service	Description	Depth (δ)	Retention (τ)	Sensitivity (σ)	Norm
p_1	$s_{01}, s_1, s_2, s_{02}, s_6, s_7$	0.67	0.75	0.80	**1.29**
p_2	$s_{01}, s_1, s_2, s_{02}, s_5$	0.85	1.00	0.64	1.46
p_3	s_{01}, s_3, s_{02}, s_5	1.00	0.96	0.63	1.52
p_4	$s_{01}, s_3, s_{02}, s_6, s_7$	0.79	0.72	0.79	1.33
p_5	$s_{01}, s_4, s_3, s_{02}, s_6, s_7$	0.82	0.74	1.00	1.49
p_6	$s_{01}, s_4, s_3, s_{02}, s_5$	0.99	0.93	0.84	1.59

sake of computation. Note that the binding to any other value (99, 100 or 1000) could be possible since it does not affect the final ranking. However the value of 10 is closer to realistic cases.

Let S be the set of admissible component services and Ω^S the vector space containing the privacy level of the services in S. Let δ_{max}, τ_{max}, σ_{max} be defined as follows: $\delta_{max} = max(\delta_i \mid [\delta_i, \tau_i, \sigma_i] \in \Omega^S)$, $\tau_{max} = max(\tau_i \mid [\delta_i, \tau_i, \sigma_i] \in \Omega^S))$, $\sigma_{max} = max(\sigma_i \mid [\delta_i, \tau_i, \sigma_i] \in \Omega^S))$. Let $\omega_i = [\delta_i, \tau_i, \sigma_i] \in \Omega^S$ be the privacy level of $s_i \in S$, its normalized privacy level $\overline{\omega_i}$ is obtained dividing each component of the vector for the corresponding maximum value, i.e. $\overline{\omega_i} = \left[\frac{\delta_i}{\delta_{max}}, \frac{\tau_i}{\tau_{max}}, \frac{\sigma_i}{\sigma_{max}}\right]$.

If the normalized vector corresponding to a composite service is optimal with respect to all dimensions, such a composite service is the most privacy-preserving composite service. Otherwise, the most privacy-preserving composite service should be determined by analyzing the components forming the privacy level. However, end-users often are not able to understand the consequences of their privacy preferences. In addition, requiring the user to specify additional information makes the level of her involvement too high [170] and, thus, the selection process cannot be automated. Decision making should be simple and intuitive as well easy to review [175]. Therefore, instead of asking the user to set her priorities over the privacy dimensions, we aggregate them using an approach based on the norm. We compute the privacy of a composite service as the average of the criteria forming the privacy level. Given a privacy level $\omega_i \in \Omega^S$, we denote the norm of its normalization as $\|\overline{\omega_i}\|$. The composite service, for which the norm of its normalized privacy level $\|\overline{\omega_i}\|$ is the lowest, is most privacy-preserving composite service, i.e. $min(\|\overline{\omega_i}\| \mid \omega_i \in \Omega^S)$.

Example 8 *Each admissible composite service in Figure 5.5 is represented as a 3D-point in Figure 5.6. The dimensions δ, ν and σ as well as the norm for each composite service are presented in Table 5.3. The height of a point represents its aggregated sensitivity, whereas the most right points are those with a higher depth, and those more*

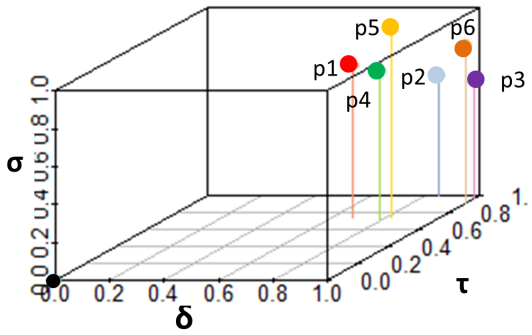


Figure 5.6: Ranking: Graph Representation

in the back have a longer retention period. Intuitively, we prefer those composite services represented by the lowest, left-most, front-most points on the graph. The norm gives a precise measure of the privacy level of composite services and, thus, makes it possible to distinguish the most privacy-preserving composite service, represented by p_1 *in our example.* □

Notice, however, that the framework is flexible enough to allow users to account more a particular dimension by specifying weights for the dimensions. These weights can be used to calculate the (weighted) average of the privacy level. For instance, a user can select the composite service that requires the release of the less sensitive data by setting the weight for the first two components to 0.

5.5 System Architecture

In this section, we describe the architecture of a java-based prototype implementing our approach. As shown in Figure 5.7, our prototype consists of four main components: a *Client Application*, a *Privacy-Aware Orchestrator and Ranker*, a *UDDI Registry* [176], and a *UDDI Database*. The *Client Application* provides the user with a user-friendly interface that allows her to request functionalities and to express privacy preferences. The *Privacy-Aware Orchestrator and Ranker* provides two main functionalities: i) the possibility to query the *UDDI Registry* to select web services that match users' functional requirements and privacy preferences; ii) the ranking of the admissible composite services according to their privacy level. The *UDDI Registry* allows service providers to publish their web services, and applications to locate such services. The information related to the providers, the web services they offer

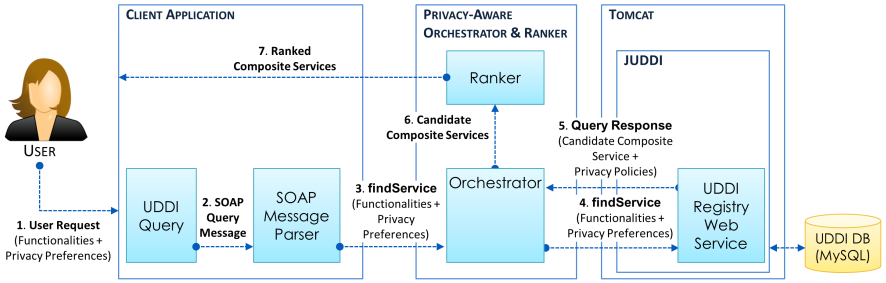


Figure 5.7: Prototype Architecture

and their interfaces is stored in the *UDDI Database*. To implement our prototype, we used jUDDI[2], the java-based implementation of the UDDI standard. With this implementation, the *UDDI Registry* is a web service running under the Apache Tomcat[3] application server. The *Privacy-Aware Orchestrator and Ranker*, also implemented as a web service, acts as a proxy between the *Client Application* and the *UDDI Registry*. The jUDDI client proxy allows the *Privacy-Aware Orchestrator and Ranker* to query the *UDDI Registry*. The *Privacy-Aware Orchestrator and Ranker* component delegates the handling of standard UDDI queries to the *UDDI Registry*. In turn, the *UDDI Registry* locally processes the results of the query to determine all the possible web service compositions for which service providers' privacy policies satisfy the user's privacy preferences. Furthermore, the *Privacy-Aware Orchestrator and Ranker* provides the interface findService, which extends the traditional *findService* operation of the UDDI standard by adding the attribute containing the user's privacy preferences. When the users places a request, a SOAP [177] message containing the query to the UDDI Registry is generated by the *Client Application*. The SOAP message is analyzed by a SOAP message parser which is part of the *Client Application*: in case the SOAP message contains a standard UDDI query, the standard UDDI *findService* operation is invoked; our implementation of findService is called otherwise. In the latter case, the *UDDI Registry* returns the list of candidate composite services to the *Privacy-Aware Orchestrator and Ranker*, which ranks them according to their privacy level and returns them to the *Client Application*.

To allow the automatic handling of privacy policies and privacy preference, we express them by using an extended version of the WS-Policy standard [178] that has been specially devised for this purpose and that is explained in the next subsection.

[2] http://juddi.apache.org/
[3] http://tomcat.apache.org/

Table 5.4: WS-Policy Extension to Express Privacy Preferences

Privacy Assertion	Description
$\langle PrivacyAssertion \rangle$	Assertion representing a privacy preference.
$\langle Purpose \rangle$	Purpose for which users' personal data will be collected.
$\langle ProtectedParts \rangle$	User's personal data item (e.g. name).
$\langle Sensitivity \rangle$	Sensitivity of personal data item.
$\langle AuthorizedEntities \rangle$	Visibility of personal data item.
$\langle RetentionPeriod \rangle$	Retention period of personal data item.
$\langle Delegation \rangle$	Re-delegation depth of personal data item.

5.5.1 Extending WS-POLICY

WS-Policy [178] is a W3C standard which allows users and service providers to agree on a mutual acceptable policy before interacting. Policies expressed with WS-Policy are specified as a set of assertions showing the capabilities of a service or the requirements of a user. Currently, WS-Policy supports assertions related to message integrity and confidentiality (e.g., which parts of a SOAP message need to be encrypted or signed), authentication techniques and the use of specific algorithms. However, it does not allow the specification of privacy related assertions. Thus, to specify users' privacy preferences and service providers' privacy policies as defined in our model, we have extended the WS-Policy standard by introducing new types of assertions.

Table 5.4 lists the new element we added to the WS-Policy standard. The element $\langle PrivacyAssertion \rangle$ is the main container, while the element $\langle Purpose \rangle$ denotes the purpose for which user's personal data can be collected. The element $\langle ProtectedParts \rangle$ specifies, by means of XPath[4] expressions, which part of the SOAP message is carrying the user's personal data items that need to be protected from unauthorized disclosure, while the element $\langle Sensitivity \rangle$ defines the sensitivity level of such data items. The element $\langle AuthorizedEntities \rangle$ identify the URI of web services that are entitled to access the user's personal data items; an empty list of $\langle AuthorizedEntities \rangle$ indicates that no entity is entitled to access the $\langle ProtectedParts \rangle$ while the keywords All can be used to indicate that every entity can access it. The elements $\langle RetentionPeriod \rangle$ and $\langle Delegation \rangle$ represent, respectively, the maximum time for which the service provider can keep a copy of the data and the the delegation depth which limits the sharing of such data with other web service entities.

In the following, we illustrate how the WS-Policy extension we propose can be

[4] `http://www.w3.org/TR/xpath/`

Table 5.5: Example of WS-Policy to express Bob's Preferences w.r.t. its credit card data.

```
<wsp:Policy xmlns:wsp= "http://www.w3.org/ns/ws-policy">
   <wsp:All>
      <PrivacyAssertion>
         <ProtectedParts>
            <sp:XPath>//Body/CreditCard/</sp:XPath>
         </ProtectedParts>
         <Purpose>Rent Car</Purpose>
         <Sensitivity>10</Sensitivity>
         <AuthorizedEntities>
            <AuthorizedEntitiy>
               http://example.com/TravelWithUs
            </AuthorizedEntity>
         </AuthorizedEntities>
         <RetentionPeriod>12</RetentionPeriod>
         <Delegation>3</Delegation>
      </PrivacyAssertion>
      <PrivacyAssertion>
      ....
      </PrivacyAssertion>
   </wsp:All>
</wsp:Policy>
```

used in practice: first, we show how the user Bob can express privacy preferences, then we proceed to express the TravelForLess's privacy policy. As shown in Example 6, Bob is willing to share his Credit Card with the trusted service TravelWithUs to fulfill the purpose Rent Car. This privacy preference is represented by the $\langle PrivacyAssertion \rangle$ shown in Table 5.5. The assertion has several sub-elements representing the different components of privacy preferences as specified in Definition 13. $\langle ProtectedParts \rangle$ includes an XPath expression that indicates the XML element $\langle CreditCard \rangle$ contained in the body of the SOAP message sent to invoke the operations offered by TravelForLess web service. The value of element $\langle Purpose \rangle$ denotes that $\langle CreditCard \rangle$ can be disclosed to fulfill the RentCar purpose. The value of element $\langle Sensitivity \rangle$ specifies how sensitive $\langle CreditCard \rangle$ is for Bob on a scale from 1 to 10. $\langle AuthorizedEntities \rangle$ has a subelement $\langle AuthorizedEntity \rangle$ for each web service entity that is entitled to access $\langle CreditCard \rangle$. In this par-

Table 5.6: Example of WS-Policy to Express TravelForLess Policy

```
<wsp:Policy xmlns:wsp= "http://www.w3.org/ns/ws-policy">
   <wsp:All>
      <PrivacyAssertion>
         <ProtectedParts>
            <sp:XPath>//Body/PhoneNumber/</sp:XPath>
         </ProtectedParts>
         <Purpose>\Rent Car</Purpose>
         <AuthorizedEntities/>
         <RetentionPeriod>36</RetentionPeriod>
         <Delegation>Unlimited</Delegation>
      </PrivacyAssertion>
      <PrivacyAssertion>
      ....
      </PrivacyAssertion>
   </wsp:All>
</wsp:Policy>
```

ticular case, it has only one sub-element whose value denotes the TravelWithUs service. Finally, the value of element $\langle RetentionPeriod \rangle$ specifies that a copy of $\langle CreditCard \rangle$ can only be kept for 12 months, while the value of element $\langle Delegation \rangle$ denotes that the re-delegation path has 3 as maximum length.

Table 5.6 shows how our extension of WS-Policy can be used to express the privacy policy of TravelWithUs. In the example shown in the table, we focus on the privacy statements related to the PhoneNumber data item which are represented by a $\langle PrivacyAssertion \rangle$. The element $\langle ProtectedParts \rangle$ includes an XPath expression referring to the XML element PhoneNumber contained in the body of the SOAP message sent to access the service offered by TravelWithUs. The value of element $\langle Purpose \rangle$ denotes that PhoneNumber can be disclosed only to fulfill RentCar purpose. Note that the element $\langle AuthorizedEntities \rangle$ has no $\langle AuthorizedEntity \rangle$ sub-elements to denote that TravelWithUs can give a copy of PhoneNumber to any other web service entity. Finally, the element $\langle RetentionPeriod \rangle$ specifies that a copy of PhoneNumber will be kept for 36 months and the element $\langle Delegation \rangle$ expresses that TravelWithUs can further share PhoneNumber without limitations.

5.6 Limitations

The solution we presented in the previous sections allows privacy-conscious providers to agree with their users about the personal data that will be shared and used during the service provision. The solution provides a model to represent privacy policies and user preferences that is based on four dimensions: *purpose*, *visibility*, *retention period* and *sensitivity*. Although the presence of such dimensions allows for the specification of fine-grain policies and preferences, it can also become a burden for the user to provide all the required information. This could result in a lack of usability of the solution which could limit its adoption. To test the usability of the system a user-study is necessary. The study should be aimed at testing the perceived ease of use and the perceived usefulness of the system. Also, it would be interesting to measure the trade-off between the privacy benefits provided by the system and the effort needed for its usage. Finally, note that a formal characterization of the properties of our algorithm such as complexity and termination is missing and should be added.

5.7 Conclusions

In this chapter, we presented a novel approach to assist users and providers in the composition and selection of composite services that satisfy users' privacy preferences. With respect to other proposals for privacy-preserving web service composition, our approach supports the specification of fine-grained privacy policies and preferences based on different privacy dimensions. In addition, our approach ranks the generated composite web services with respect to their privacy level, which quantifies the risk of unauthorized disclosure of user information based on sensitivity, visibility and retention period. We also proposed a prototype implementing our solutions using java-base technologies. Finally, we extended the WS-Policy standard to enable the definition of privacy policies and privacy preferences as defined in our model.

Chapter 6

Monitoring Data Usage: Database Leakage Detection

In the previous chapters we presented solutions to support users in choosing online services which offer the most appropriate level of privacy protection. Privacy protection has been evaluated in terms of completeness of privacy policies (Chapter 3), in terms of the amount of personal data collected (Chapter 4) and as a combination of data collected, retention time and data sensitivity (Chapter 5). Once users release their data to accomplish a certain task, the data cycle is not over. Indeed, data is usually stored in data repositories where it keeps being accessed and used (e.g., by provider's employees to conduct their work). At this stage, privacy is still at risk since data can be revealed to unauthorized recipients because of a data leakage. Data leakages can be caused by hackers who gain access to the data repository or by malicious or careless insiders who reveal user's sensitive information. In this chapter, we propose a solution to detect data leakage by monitoring data activities over repositories such as databases. The solution we propose works by learning normal profiles of database usage, and by flagging any deviation from such profiles as an anomaly.

6.1 Introduction

Data represents a great value for most organizations. Databases, storing customer and confidential business data, are a core asset that need to be protected from illegitimate usage. This makes data leakage, i.e. the unauthorized/unwanted transmission of data and information [6], a major threat. According to a Ponemon Institute study[1], in 2009 data breach incidents cost U.S. companies an average of $6.75 million per-incident. These costs include theft of corporate intellectual property, damages to reputation and decrease of costumers' trust. To reduce these enormous costs and comply with legislation, e.g. the new EU Data Protection regulation requiring data breach notification within 24 hours [179], timely detection of data leakage is essential.

Insider threats, i.e. careless, unhappy or malevolent employees are amongst the main sources of data leakage. Since insiders have the right of accessing internal resources such as databases, they can harm a company more easily than external threats such as hackers. Leakages from database account for most of the records globally disclosed in 2012 [180]. For these reasons, in this chapter we focus on the problem of detecting database leakages caused by insiders.

A first line of defense against data leakage is formed by Access Control (AC) mechanisms [4] which aim at regulating users rights to access certain data. AC suffers of some disadvantages, e.g. it is not always expressive enough to allow the definition of fine-grain access rules and, especially in dynamic domains, it has high costs of maintainability (update of rights and roles). In addition, AC might reduce data availability which is critical in emergency situations (e.g. in healthcare domains) or productivity (e.g. time loss to ask for permission to access certain documents). As a result, organizations often apply relaxed AC policies which give users access to more information than they actually need [181]. To solve the problem of reduced data availability in emergency situations, techniques such as Break the Glass (BtG) have been proposed [182]. These techniques enable a user to gain complete access to the system (e.g. in the emergency room when a patient needs immediate care). However, all the actions taken by a user after BtG procedure have been initialized are recorded to guarantee accountability in case the mechanism is misused (e.g. the BtG is used to access sensitive information for illegitimate purposes).

Beside AC, there are tools and methodologies for data leakage detection [5] which can spot leakages by operating at different locations on the data path (e.g. network, workstation or database). In this chapter we focus on solutions operating at a database level so that leakages can be detected at a very early stage, i.e. when sensitive data is leaving its primary source.

Commercial tools and academic solutions addressing database leakage detection

[1] http://www.ponemon.org/news-2/23

typically work by monitoring database traffic in terms of SQL queries. Existing solutions can be divided into *signature-based* and *behavioural-based* systems [183]. Generally, in signature-based systems a blacklist defines the set of dangerous or denied access patterns. There are also whitelist-based approaches where rules of permitted patters are defined, and everything that does not match such patters is blocked. The whitelisting approach can be very costly to setup and to maintain (many rules are involved) and it requires an extensive and comprehensive knowledge of internal roles and responsibilities in order to create an effective rule set. On the other hand, behavioural-based systems automatically learn permitted access patterns by observing normal activities and mark every anomaly as a potential threat. The main problem of signature-based approaches is that they can only detect well-known attacks, whereas behavioural-based approaches have the great potential of detecting unknown database attacks. In addition, by automatically generating fine-grained profiles, behavioural-based solutions require less human-effort thus offering the best possible detection at the lowest cost.

These advantages make behavioural-based approaches widely adopted in literature [184, 185, 186, 187, 188, 189, 190, 191, 192]. However, the existing approaches have several drawbacks. The first problem is the high *False Positive Rate* (FPR) they usually generate. Since each false alert has to be analyzed by a security officer, false positives have a high operational cost. In network anomaly detection [193, 194] (a different yet related field), a system starts to be "usable in practice" when it shows a FPR in the order of 0,01%, a rate by far not attained by present database anomaly detection systems. A way to keep FPR low is to frequently update normal usage profiles, so that they reliably represent actual normal behaviour. However, profiles update is usually a costly operation since it often requires re-training the model. The second drawback is that current solutions provide little or no support for alert handling. Usually, when an alert is raised, it is accompanied by a *deviation degree*, or an *anomaly score*. Unfortunately, this information is virtually useless for the security officer as it does not support him in understanding "what is going on". To this end, signature-based systems have an advantage: when they raise an alert, they can say exactly which policy is violated and why this violation may constitute a problem. For behavior-based systems, it is more difficult to "explain" the reasons of an anomaly mainly because of their *black-box* nature, i.e. the underlying engine (be it a neural network or a machine learning classifier) is difficult to understand by a human. To enable practical detection of unknown database leakage threats we introduce what, to the best of our knowledge, is the first *white-box* behavioural-based database leakage detection system. Our solution advances the state-of-the-art in the following ways:

1. We propose a comprehensive database leakage detection framework. We build histogram-based profiles over a wide feature space which leads to the construc-

tion of fine-grain profiles and allows the detection of several types of attacks. Thanks to the white-box approach, users can easily understand what profiles mean in terms of database activities, and manually inspect and refine profiles if necessary;

2. When an alarm is raised, we clearly determine the origins of each anomaly, hence facilitating the security officer's handling of the alarms;
3. The use of histogram-based profiles enables online-learning (profiles are incrementally built) and facilitates updates (no re-training is required);
4. We introduce a feedback mechanism which enables the security officer to mark false positives. This mechanism has the advantage of: i) speeding-up the post-processing of alarms; and ii) allowing the update of existing profiles to progressively reduce FPR;
5. We introduce a new mechanism to aggregate features based on a coupling score which identifies pairs of features that help to detect more threats if considered together rather than independently;
6. We propose a transaction flow analysis which enables the detection of attacks spanned over multiple transactions;
7. We validate our framework with an extensive set of experiments carried out over two different dataset, one created from simulated scenarios, and the other consisting of more than 12 millions real transactions coming from an enterprise operational database. In addition, in our experiments we benchmark our system against other approaches from the literature.

The remainder of this chapter is structured as follows: in Section 6.2 we describe the state-of-the-art of this field, while in sections 6.3 to 6.8 we discuss the main components of our framework. In Section 6.9 we describe our evaluation methodology, while in Section 6.10, Section 6.11 and Section 6.12 we present the results of our experiments. Finally, in Section 6.14 we discuss the results and draw the conclusions.

6.2 Background and Related Work

6.2.1 Database Leakage Detection Solutions

Several solutions for database leakages detection are available as academic research or commercial applications. Available solutions can be divided into *signature-based* and *behavioural-based*. The signature-based approach detects potential leakages by observing rule violations, while the behavioural-based approach detects anomalies from database usage. Academic solutions, which mostly adopt behavioural-based

approaches, mainly differ from each other for the type of features used to build normal profiles. A feature is a characteristic of a database transaction and it can be of three types: *syntax-centric* (e.g. the query command, the list of columns and tables accessed), *context-centric* (e.g. from *where*, *when* and from *whom* the transaction comes from), and *result-centric* (e.g. the actual data values retrieved).

The works presented in [192], [185] and [184] are purely syntax-centric. In the work discussed in [192], normal profiles are built based on users' frequent item-sets, i.e. the list of tables and columns a user commonly works with. The authors develop a distance measure between such item-sets and each new incoming transaction: if the distance is bigger than a predefined threshold, an alarm is raised. In [185] normal profiles are built using a Naïve Bayes classifier. The system learns to predict the user based on the SQL command and on the list of tables and columns appearing in the query text. When a new transaction takes place, the classifier will predict the user-id: in case a mismatch between the prediction and the actual value exists, an alarm is raised. The approach presented in [187] is very similar to the one just discussed, with a slightly more extended feature space. Finally, the work discussed in [184] builds normal behaviour profiles based on the assumption that SQL commands and their order within database transactions flow are relevant. Here, a database transaction includes several queries and it is represented as a directed graph describing the different execution paths (sequences of selects, inserts, updates, deletes) from the beginning of the transaction to the commit or rollback commands. During detection, if an attacker executes valid queries but in an incorrect order, i.e. not in the paths of the graph, an alarm is raised.

Pure syntax-centric approaches have to deal with the high flexibility of the SQL language, where similar queries can generate different results, and syntactically different queries can access the same data. This limits the attacks that such an approach can detect: it fails to detect situations where the syntax of the query submitted is normal but the data it retrieves is anomalous. These considerations are at the basis of the solution proposed in [189] that suggests to profile normal behaviour based on the data values users retrieve. Here, a statistical summary of the result set (S-Vector) is computed for each query and all the vectors related to the same user form a cluster. When a new query is executed, its S-vector is computed: if it belongs to the user's cluster the query is considered normal, while it is considered anomalous otherwise. A mixed approach which combines result-centric and context-centric is used by the authors in [190] which creates association rules between the context and the result set. During the detection phase, the appropriate set of rules is retrieved according to the context of the request. A record in the result set that matches at least one of the rules is considered to be normal, while it is considered anomalous otherwise. In this way, the same request may be legitimate if performed within one context but abnormal within another. Finally, in [191] syntax-centric and result-centric approaches

Table 6.1: Comparison of Commercial/Academic Database Leakage Solutions.

Approach	**Commercial Solutions**				**Academic Research**					
	Guardium	DbProtect	Power Broker	Secure Sphere	Wu et al.	Kamra et al.	Fonseca et al.	Mathew et al.	Gafny et al.	Santos et al.
Signature Based	✓	✓	✓	✓						
Behavioral Based				✓	✓	✓	✓	✓	✓	✓
Features										
Command	✓	✓	✓	✓	✓	✓	✓		✓	✓
Tables	✓	✓	✓	✓	✓	✓	✓			✓
Columns		✓	✓		✓	✓	✓			✓
Where					✓	✓	✓			
User id	✓	✓			✓	✓	✓	✓	✓	✓
User role	✓	✓			✓	✓	✓	✓	✓	✓
Time		✓	✓	✓	✓					✓
Location		✓	✓	✓	✓					✓
# Rows/Bytes			✓	✓					✓	✓
Result Set			✓	✓				✓	✓	✓

are combined. Normal profiles are represented in terms of statistical distribution. For each feature (all strictly numeric), the probability distribution (H_0) is computed. When a new query is executed, if a certain feature's distribution does not match the original probability distribution H_0, the query is considered anomalous.

Given the importance of data leakage detection from an enterprise point of view, several vendors started to provide database leakage detection solutions. Table 6.1 compares commercial and academic solutions on the basis of the approach (signature or behavioural based), and the feature set used to build profiles of normal behaviour. Whereas academic researches focus on behavioural-based approaches, vendors mostly adopt signature-based approaches (e.g. IBM Guardium[2], Dbprotect[3], PowerBroker[4]) allowing to set policies with varying granularity levels depending on the available features. However, the benefits of the behavioural-based approach have also attracted vendors (e.g. Imperva SecureSphere[5]) which apply learning procedure for the initial policy definition.

[2] `http://www-01.ibm.com/software/data/guardium/`
[3] `http://www.appsecinc.com/index.php/products/dbprotect`
[4] `http://www.beyondtrust.com/Products/PowerBrokerDatabases/`
[5] `http://www.imperva.com/Products/DatabaseSecurity`

6.2.2 Anomaly Detection Techniques

Behavioural-based solutions detect database leakages as deviations from normal behaviour, a problem generally known as anomaly detection. There are several techniques that can be used for anomaly detection, in this section we discuss those that can be applied in the context of database leakage detection. Anomaly detection techniques can be divided into supervised, semi-supervised, and unsupervised ones [195]. In supervised approaches, it is assumed that the training set contains queries labeled either as normal or as anomalous. Standard machine learning algorithms are then applied to generate classifiers able to automatically separate normal queries from anomalous ones. When, as in anomaly detection, anomalies are way less frequent than normal transactions, supervised techniques suffers from the problem of unbalanced class which produces poor classifiers. In the semisupervised approach the assumption is that the training set contains only labels for normal transactions. Finally, in the unsupervised approach no labels are needed: the dataset is sought for similarities so that outliers can easily pop out.

In the context of database anomaly detection a labeled dataset is very hard to obtain, thus unsupervised techniques are the most utilized. Examples of unsupervised techniques include clustering, association rules, and statistical methods. Clustering techniques group similar instances of the training set according to a distance function [196, 197]. Any sample that does not belong to any cluster is considered an anomaly. A major drawback of clustering techniques is that the number of clusters has to be defined a-priori. In addition, a complete re-training is necessary if a new sample has to be added to the model, making update operations very complex and time consuming. Association rules have the main advantage of being self-explanatory, thus creating a white-box system. However, they have a high computational cost and high memory consumption [198]. Finally, statistical methods are based on the probabilistic model generated from the training set: if the probability of the new data instance according to the probabilistic model is very low, then the instance is considered an outlier. Statistical methods include Hidden Markov Models (HMM) and histogram analysis. HMMs have the capabilities of modeling temporal relationships. However, the high computational and memory cost, together with the high number of parameters that need to be set can discourage their usage [183, 198]. Histogram-based approaches, which we use in our solution, are the simplest nonparametric statistical techniques. This technique is computationally inexpensive, and generates a self explanatory model. Its simplicity and intrinsic white-box structure, together with the support to the *online learning* – no need of re-training when a new sample is added to the model– make this solution the best candidate when a model easy to understand and to update is required.

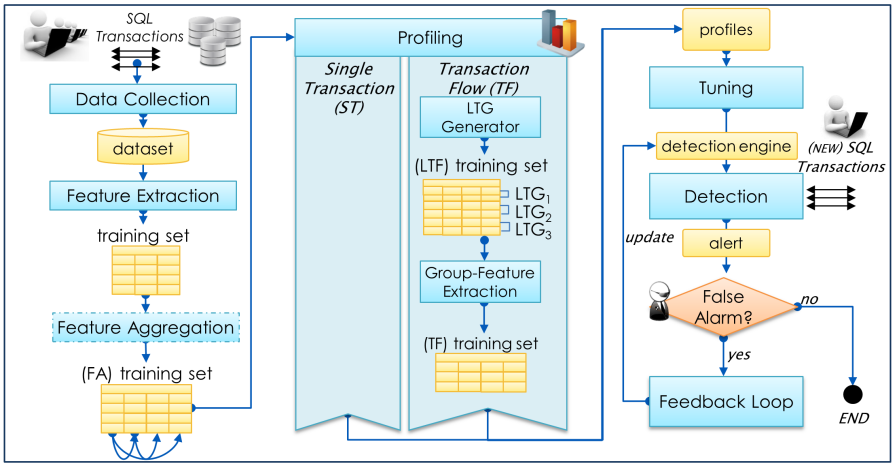


Figure 6.1: Framework Overview. Dashed Lines Refer to Optional Phases.

6.3 Framework Overview

The general goal of our solution is the detection of potential database leakage and misuse, by identifying anomalies w.r.t. *normal* database usage profiles. Our solution pursues two core objectives: i) maximizing the detection rate (DR) of known and unknown threats; and ii) minimizing the false positive rate (FPR), i.e. the number of normal transactions erroneously marked as anomalous. Figure 6.1 shows an overview of the framework we propose and its main phases. The first phase of our framework is the *Data Collection* where users' activities are monitored and the transactions with the database are captured to form a *dataset* of usual behaviour[6]. During the subsequent *Feature Extraction* phase, the dataset is transformed in a tabular version known as *training set*, which contains a column for each feature we intend to use, and a row for each transaction in the dataset. *Feature Aggregation* (FA) is an optional phase which allows the enrichment of the training set by joining features together. Once ready, the training set is given as input to the *Profiling* phase which builds histogram-based profiles. Profiles can be built by using a *single transaction* (ST) or a *transaction flow* (TF) approach and they can be inspected and tuned during the *Tuning* phase. The output of the tuning is a *detection engine* which during the *Detection* phase flags each new transaction as *normal* or *anomalous*. If the transaction is considered normal, it is

[6]In the remainder, we use the term *transaction* to refer to a query submitted to the database and to the contextual information such as who is submitting the query or what is the result set, while we use the term *query text* to refer to the actual SQL query statement.

used to update the profiles (*continuous learning*[7]). In case of an anomalous transaction, an alert which clearly states the origins of the anomaly is raised. Finally, in case of alarm, the *Feedback Loop* allows the security officer to flag possible false positive causing an immediate update of our profiles.

Note that our approach can be consider a detection system rather than a prevention one, i.e. a transaction is not blocked but an alarm is raised if it is suspected to be a threat to data leakage or misuse. To do prevention, a transaction has to be suspended before its execution in case a threat is suspected. While prevention is desirable in certain domains, it might be counterproductive in others (e.g. emergency healthcare). In this chapter we focus on the detection aspect of our solution. In the following sections, each phase of our framework is described in details.

6.4 The Data Collection and Feature Extraction Phase

The first step of our solution is the Data Collection. During this phase user's transactions with the database are intercepted and stored.

Definition 15 (Transaction -Tr) *A* transaction Tr *is a triple formed by the SQL query text* Q*, the corresponding result set* RS*, and additional contextual information* CI *such as the* user *submitting the query, the* response code *given by the database or the* time *when the transaction starts or ends. In the sequel we use* $\mathcal{T}$ *to indicate the set of all possible transactions.*

There are several ways to capture users' transactions [199]. One could enable the auditing facilities provided by database management systems (DBMSs). In case this option is not viable, e.g. because of the high overhead introduced, another option could be to tap the communication channel between the client applications and the database server (SQL traffic sniffing). Independently on how the data it is collected, it is important that the Data Collection phase lasts long enough to capture all (or most of) the transactions normally submitted to the database. Transactions collected during this period need to be preprocessed in order to extract the features relevant to our analysis. A *feature* represents a distinctive attribute that can be extracted from a transaction and that helps to characterise normal behaviour. Examples of *feature* can be the *query command*, the *time* when the transaction starts (e.g. *10:35 a.m. UTC*) or the *userid* of the user submitting the transaction (e.g. *bob* or *sally*). The sequence of all the features used in our analysis is referred as *feature space*.

[7]The Data Collection phase only last for a certain period of time at the beginning. Once the detection engine is ready, all new transaction pass through it.

Table 6.2: Feature Space and Relationships with Detactable Threats.

	Feature	Detected Threats
Syntax Centric	Query (command, tables, columns)	These features help in detecting leakages due to users accessing information out of their working scope (e.g. due to *excessive privileges granted* or *misuse of privileges*).
	Where Clause (length, special chars, columns and tables)	*SQL Injection* and *XSS attacks* usually act on the syntax of a query to inject malicious (SQL) statements which can be used to extract sensitive information from the database. Dangerous injected statements usually contain specific keywords that can be monitored to help the detection.
Context Centric	Response Code	Specific codes are given for specific database events, e.g. failing login attempts. Multiple failures might indicate a *password guess* attack, which might start a data leakage.
	Client/DBMS USERID/ROLE	Identifying which end-user and with which role or which client application is responsible of anomalous activities is helpful for *accountability reasons*.
	Timestamp, IP Address	Access from unusual location or at unusual time might indicate the credentials have been stolen e.g. someone is carrying on a *masquerade attack*.
Result Centric	N. of Records and Bytes	Retrieving a large quantities of data (e.g. copying customer list) might indicate data a data leakage/misuse is taking place.
	Result Set	The data values returned by the query helps in detecting misuses when the query syntax is legitimate (e.g. a doctor can access the table *disease* and *patient*) but the results retrieved are not (e.g. the doctor accesses records of patients he does not treat).

Definition 16 (Feature – f) *A* feature f *is a function* $\mathcal{T} \longrightarrow V_f$ *associating a value* v *from the codomain* V_f *to each transaction* $Tr \in \mathcal{T}$. *We call* feature value $v_f = f(Tr)$ *the result of applying the function* f *to the transaction* Tr.

Definition 17 (Feature space –F) *The* feature space $F = \langle f_1, ..., f_n \rangle$ *is the sequence of features considered in our approach.*

The feature space determines the level of details of the profiles we build and the kind of attacks that can be detected. Table 6.2 describes our feature space; features are grouped in syntax-centric (related to the query text Q), context-centric (related to the contextual information CI) and result-centric (related to the result set RS). The table shows which specific data leakage threat each feature (or group of features) is able to identify. For example, syntax-centric features enable the detection of leakages caused by privileges misuses. By using context-centric features it is possible to detect anomalies related to database usage at unusual location or time. Finally, result-centric features represent the only way to spot leakages caused by the access to illegitimate data values. Once the feature space has been defined, each transaction

Table 6.3: Example of Training Set (TS).

Feature Type	CLIENT_UID nominal	QUERY_LEN numeric	QUERY_COMMAND nominal	QUERY_COLUMN_SET set	TIMESTAMP time
T_1	rob	25	select	{name, sex}	8:47
T_2	rob	200	insert	{name, sex}	9:15
T_3	rob	25	select	{name}	13:25
T_4	rob	200	select	{sex, name}	14:50
T_5	sally	10	select	{age}	15:27
T_6	sally	10	select	{age}	14:30
T_7	sally	38	select	{name, sex, age}	17:10
T_8	sally	5	delete	{age}	19:11

collected during the Data Collection phase has to be analyzed to extract the features: we call *transaction vector* the sequence of feature values resulting from the Feature Extraction operation and *training set*, the collection of all the transaction vectors.

Definition 18 (Transaction vector – T) *Given a feature space $F = \langle f_1, ..., f_n \rangle$ and a transaction Tr, the* transaction vector $T = \langle f_1(Tr) = v_{f_1}, ..., f_n(Tr) = v_{f_n} \rangle$ *is a sequence of feature values v_{f_i}.*

Definition 19 (Training set – TS) *A training set $TS = \{T_1, ..., T_k\}$ contains the $k = |TS|$ transaction vectors we use to build profiles during the Profiling phase.*

In Table 6.3 we show an example of training set containing 8 transaction vectors submitted by users *rob* and *sally*. If our feature space is defined as $F\langle$= CLIENT_UID, QUERY_LEN, QUERY_COMMAND, QUERY_COLUMN_SET, TIMESTAMP$\rangle$ then the transaction vector $T_1 = \langle$ *rob, 25,select, {name, sex}, 8:47*$\rangle$ represents a transaction submitted by user *rob*, with a query text which is 25 characters long, which executes a $select$ statement over the columns $\{name, sex\}$ at the time *8:47*. In the next section we describe how profiles are built. Note that before the *Profiling* phase, the optional *Feature Aggregation* phase can take place. This phase is presented in Section 6.8 after introducing the concepts necessary to its discussion in Section 6.5, Section 6.6 and Section 6.7.

6.5 The Profiling Phase

A main problem in the detection of anomalous database transactions is the definition of normal profiles. Profiles should be able to completely describe normal database usage, i.e. which transactions are submitted, when, what results are retrieved, etc. We propose two profiling approaches: i) the *single transaction profiling* (discussed in the next section); and ii) the *transaction flow profiling* (discussed in Section 6.5.2).

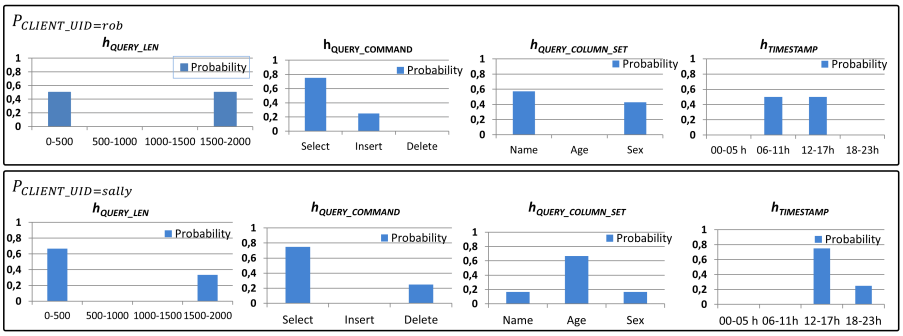


Figure 6.2: Profiles for Users *rob* and *sally* w.r.t. to the Training Set in Table 6.3.

6.5.1 Single Transaction Profiling

In the single transaction profiling each transaction with the database is used independently from the others to build normal profiles. Once we collected the training set, to build the profiles, we need to partition the codomain of each feature of the feature space in a sequence of *bins* (the *bin sequence*). Given a feature, a bin is a subset of its codomain. In presence of a training set TS, we can compute the probability that a feature value v_f of a transaction $T \in TS$ will actually fall in a specific bin b_i. Given a training set, a transaction vector and a bin sequence, we can compute the *bin probability* of each bin in the bin sequence. We formalize these concepts as follows.

Definition 20 (Bin – b_i and Bin Sequence – B_f) *Given a feature f with the corresponding codomain V_f, we define a* bin sequence $B_f = \langle b_1, ..., b_m \rangle$ *as a partitioning of V_f s.t. the bins $b_1, ..., b_m \in B_f$ are disjoint ($b_i \cap b_j = \emptyset \forall i \neq j$) and their union is a superset of V_f ($\bigcup b_i \supseteq V_f$). Each b_i in B_f is called* bin.

Definition 21 (Bin Probability – $prob_{f,b_i}$) *Given a training set TS, a feature f and a corresponding bin sequence $B_f = \langle b_1, ..., b_m \rangle$, for each $b_i \in B_f$ we define the* bin probability *as the probability that a feature value $v_f \in V_f$ falls in the bin b_i, namely* $prob_{f,b_i}(TS) = \frac{|\{v_f | T \in TS\} \cap b_i|}{|TS|}$.

Note that given a training set TS the number of bin can be different for different features (see Section 6.5.1). When computing the bin probability, we divide over the number of transactions rather than the number of bin so that the bin probability can be comparable amongst different features (this is especially useful to allow the definition of a global threshold as discussed in Section 6.6). We can represent such probability in a graphical way by using histograms (as represented in Figure 6.2) which are defined as follows.

Definition 22 (Histogram – h_f and Histogram Size – $|h_f|$) *Given a training set TS, a feature f and a corresponding bin sequence $B_f = \langle b_1, ..., b_m \rangle$, we define a histogram $h_f(TS) = \langle prob_{f,b_1}(TS), ..., prob_{f,b_m}(TS) \rangle$ as a sequence of bin probabilities. In addition, we define the* histogram size $|h_f|$ *as the number of bins in the bin sequence, $|h_f| = |B_f|$.*

Given a training set we want to produce behavior profiles, e.g. profiles for each user or each role in the system. To do so, we first need to select which transactions in TS belong to a specific profile. In this respect, we define a *profile constraint* which maps transactions to *true* or *false*, according to whether they belong or not to a profile. For example, a constraint can be in the form $c : CLIENT\_UID = rob$. At this point we can define a profile $P|_c$ as a sequence of histograms, one for each feature in the feature space; a profile $P|_c$ is buil considering exclusively the subset of the training set for which the constraint c is true. From now on, to simplify the discussion, we look at constraints (and thus profiles) which are mutually exclusive. In this way a transaction vector T can only belong to one profile, the one for which the constraint $c(T)$ is true. Extending the discussion to cases of not mutually exclusive constraints is straightforward and we omit its discussion.

Definition 23 (Profiling Constraint – c) *Given a transaction vector T, a constraint $c(T)$ is a function mapping T into $\{true, false\}$. Given a training set TS we call $TS|_c$ the subset of TS containing exclusively the transactions for which c is true, $TS|c = \{T \in TS : c(T) = true\}$.*

Definition 24 (Profile – P_c) *Given a training set TS, a constraint c, a feature space $F = \langle f_1, ..., f_n \rangle$ and a corresponding set of bin sequences $B = \langle B_{f_1}, ..., B_{f_n} \rangle$, the profile of c w.r.t. TS is a sequence of histograms $P_c(TS|c, F, B) = \langle h_{f_1}(TS|_c), ..., h_{f_n}(TS|_c) \rangle$.*

Figure 6.2 shows the profiles built over the training set of Table 6.3 by setting the constraints $c_1 : CLIENT\_UID = rob$ and $c_2 : CLIENT\_UID = sally$ which respectively lead to the creation of profile $P_{CLIENT\_UID=rob}$ and $P_{CLIENT\_UID=sally}$. As we can see from the figure, each profile is a sequence of histograms (one for each feature of the feature space). In turn, each histogram is a sequence of bin probabilities, one for each bin. A key factor when building a histogram for a given feature is the definition of its bin sequence, namely how the feature's codomain is partitioned in bins. This topic is addressed in the following section.

Bin Sequence Construction

As aforementioned, to build the histogram for a feature f with feature values $v_f \in V_f$, we need to partition V_f into a sequence of disjoints bins. Different features can have

different data types for their values. In our case we have four different types of data, namely *nominal*, *numeric*, *time* and *set*. The way the partitioning for a feature can be made depends on its data type. For instance, the feature QUERY_COMMAND has values in $V_f = \{select, delete, insert, ...\}$ which is of type $nominal$. In this case for the partitioning we can simply take a bin for each different value encountered. However, this approach does not work for the $numeric$ type (e.g. for the feature QUERY_LEN) as having a bin for every different value encountered would lead to an explosion in the number of bins. In this case we have bins as a range of values (e.g. $[0-10], [11, 20], [21, 30]$.). The bin-width (size of the ranges) has to be a right balance between narrow (many bins with low frequency with the risk of increasing the false positive rate) and wide (few bins with high frequency with the risk of missing anomalous instances). We automatically compute the bin-width for numeric data values by using the Freedman-Diaconis rule of thumbs [200]. This rule is commonly used to determine the width of a numeric bin by applying the following formula: $bin_width = 2IQR(x)n^{-1/3}$ where $IQR(x)$ is the inter-quartile range of the numeric vector x and n is the number of element in x. More advanced techniques to determine the best bin-width according to the data values distribution, e.g. dynamic bin-width, could be adopted but that is left as a direction for future work. For features of type $time$ (e.g., the feature TIMESTAMP), the bin-width can be defined according to which representation is more meaningful for a specific domain, e.g., a bin can be as large as a day of the week, or an hour of the day, or a time shift.

Finally, with set data type (e.g., the QUERY_COLUMN_SET) we create a different bin for each element of the set (*option A*). Someone could argue that having a bin for each set instead, is a better approach (*option B*). To understand the difference between the two options in terms of anomaly detection, consider the queries Q.1 and Q.2 and let us assume they are both legitimate. When building the histogram for the features QUERY_COLUMN_SET with *option A*, we will have a histogram with three bins, $\{name\}$, $\{sex\}$, and $\{age\}$.

On the other hand, with *option B* we will have two bins, one with the set $\{age\}$ and one with the set $\{name, sex, age\}$. Let now consider queries Q.3, Q.4, and Q.5 as input to the detection module. By using *option B*, all these queries will be marked as anomalies since there is no exact corresponding bin for the values $\{sex\}$, $\{age, name\}$ and $\{name, disease\}$. If a user may access $name, sex$ and age together (Q.1) it seems reasonable she may also only access $name$ and age. As only Q.5 contains a value – $\{disease\}$ – that has never been used before, only this query should be seen as anomalous. Since *option B* would lead to false alarms for the queries Q.3 and Q.4, while *option A* would flag as anomalous only Q.5 (and only for the value $\{$*disease*$\}$ as it should be), when building histograms for *set* data type we apply *option A* (see example in Figure 6.2).

Q. 1 ***Select*** name, sex, age ***from*** patient*;*

Q. 2 ***Select*** age ***from*** patient*;*

Q. 3 ***Select*** sex ***from*** patient*;*

Q. 4 ***Select*** age, name ***from*** patient*;*

Q. 5 ***Select*** name, disease ***from*** patient*;*

6.5.2 Transaction Flow Profiling

The single transaction profiling discussed so far can detect anomalies by analyzing each transaction with the database independently. However, there are cases where a single transaction is legitimate but a group of transactions is not. For example, let us consider the case of Bob, a database administrator who plans to leave his job because he is disgruntled with his employer. To damage him, Bob plans to get the list of the customers from the organization database and to sell it to a competitor. He knows the detection system in place would raise an alarm if too much data is retrieved with a single transaction. Therefore, Bob copies the complete list of customers by executing several transactions which retrieve a small amount of data, hence passing undetected. The transaction flow profiling we discuss in this section aims to deal with such cases.

The first step necessary to create transaction flow profiles is to decide when a database transaction is connected to others, i.e. which transactions form a *logical transaction group* (LTG). For instance, a logical transaction group can be formed by the transactions *submitted by the same user and accessing the same table*, or by the transactions *submitted by the same user during a certain period of time*. We now introduce the general concepts of event flow and transaction flow. An event flow is a sequence of events which includes temporal events as triggers and transactions. A transaction flow is a subsequence of an event flow consisting of transactions only. Given an event flow, we define the concepts of *scope*, *selector* and *LTG generator* to discriminate logical transaction groups.

Definition 25 (Event Flow and Transaction Flow - EF and EF_T) *An event flow EF is a sequence of events. Examples of events are* triggers *and* transactions *with the database. A transaction flow is a sequence of transactions. Given an event flow EF, we write EF_T for its corresponding transaction flow, namely the subsequence of EF which contains transactions only. For a generic sequence S we write $S[i, j]$ for the subsequence of S consisting of its i-th to its j-th element and we write $S\langle k \rangle$ for the element at position k.*

Definition 26 (Scope - Θ_{EF}) *Let α and β be two boolean functions on event flows. The scope for an EF given by a* start condition α *and an* end condition β *is denoted as $\Theta_{EF}(\alpha, \beta)$ and it is the set of transaction flows $EF[i,j]_T$ s.t.:*

$$\begin{cases} \alpha(EF\langle i\rangle)\ is\ true; \\ \beta(EF\langle j\rangle)\ is\ true; \\ \beta(EF\langle k\rangle)\ is\ false\ \forall\ i \leq k < j. \end{cases} \tag{6.1}$$

For the way a scope is defined, it can identify different transaction flows $EF[i,j]_T$, one for each time the conditions in Equation 6.1 are true. For example, if we define a start-condition α = '*minute = 00*' and an end-condition β = '*minute = 59*', we will have a $EF[i,j]_T$ every hour. We can also define a scope as '*every n minutes*'. Note that the definition above can be generalized to be history-sensitive by defining the conditions α and β over traces (e.g. $\alpha(EF[1,k])\ is\ false$) rather than single elements. In this way it would be possible to express conditions such as '*every n transactions*' or'*every other year*'.

Definition 27 (Selector - σ) *Given a transaction T_r, a selector σ is a boolean function which defines whether T_r belongs to a logic transaction group (LTG). Typically, we express a selector in the form of constraints. For instance, $\sigma : T_r \in LTG \iff c_1 \wedge c_2 \cdots \wedge c_n$ where c_i is a constraint as defined in Definition 23.*

Definition 28 (LTG generator - Φ) *Given a scope $\Theta_{EF}(\alpha, \beta)$ and a selector σ, the LTG generator $\Phi(\Theta_{EF}(\alpha, \beta), \sigma)$ generates a LTG for each transaction flow $EF[i,j]_T$ in the scope. A transaction $T_r \in EF[i,j]_T$ will be part of the related LTG only if $\sigma(T_r)$ is true.*

Example 9 *Let us assume we have an event flow EF related to a database we are monitoring. We want to define a LTG as the group of transactions submitted by user* Bob *and accessing the table* customer *on a daily basis. To do so we first have to define the scope Θ by setting $\alpha : time = 00:00$ and $\beta : time = 23:59$. In this way the scope will identify a transaction flow $EF[i,j]_T$ every day. To select only the transactions submitted by* Bob *and accessing table* customer *we define a selector σ as follows:*

$$\sigma(T_r) = \begin{cases} true & \text{if } f_{userid}(T_r) = Bob \wedge \{customer\} \subseteq f_{TableSet}(T_r) \\ false & \text{otherwise} \end{cases} \tag{6.2}$$

Figure 6.3 shows how the LTGs are generated. Logic transaction groups, once generated, are used to create the profiles. Clearly, if we set a scope on a daily basis, to create meaningful profiles we need to monitor database activities for several days.

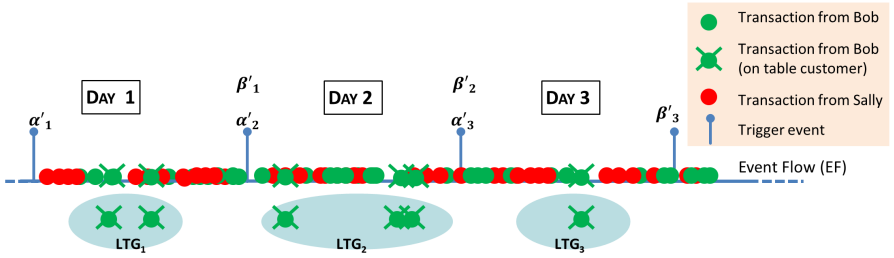


Figure 6.3: Example of Logic Transaction Groups Generation.

Thus, the definition of the scope strongly depends on the application context and the data availability: creating profiles *per day* is only meaningful if data covering activities over several days is available.

Because in the transaction flow profiling the unit of profile is a LTG, the feature space presented in Table 6.2 is no longer valid: e.g., LTG does not have a single QUERY_COMMAND, but it has many different commands. Therefore we need a different set of features to describe the characteristics of a group of transactions rather than a single one. The group-features we use when building transaction flow profiles are listed in Table 6.4. Once LTGs have been extracted from a dataset, the histogram-based profiling described in the previous section will be applied. The only difference is that a profile can be updated not after each transaction, but after each LTG has been generated in accordance with the LTG generator. Note that, depending on the scope, some features in Table 6.4 might be not needed, e.g. the number of distinct tables is useless if the scope is defined as the '*the transactions where BOB accesses the table customer*'.

6.6 The Tuning Phase

Typically, in anomaly detection solutions, the training set is assumed to be *attack-free*, i.e. no misuse is taking place during the Data Collection phase, and *exhaustive*, i.e. it is fully representative of normal behaviour. These assumptions suffer from some limitations. First, if an attacker is already active during the Data Collection, with the *attack-free* assumption there is the risk to learn as normal the attacher's malicious transactions. This is not a problem in our approach as, thanks to the white-box approach, misuses in the training set are easy to spot. For example, in Sally's profile in Figure 6.2, the probability for her to use a *delete* statement is very low: this can repre-

Table 6.4: Group-Features Used for Transaction Flow Profiling.

Group Feature	Detected Attacks
# of transactions submitted # of bytes # of records # of select all # of select	Detect anomalies in the amount of activities (e.g. an user more active than usual in terms of transactions submitted and number of records retrieved.)
# of sensitive columns # of sensitive values	Detect anomalies w.r.t. to the amount of sensitive information retrieved. Sensitive columns/values are defined by a domain expert.
# of distinct columns # of distinct tables	Detect anomalies in the amount of distinct columns/tables accessed which might indicate malicious activities.
# of insert # of update # of delete	Detects anomalies related to tampering activities (e.g. delete records from auditing tables to hide misbehaviour).
# of sequential failed login	Detect password guess attack .

sent either a situation of misuse (the *delete* privilege has been erroneously granted to sally who is abusing it) or a situation of normal behaviour (the *delete* is just normally very rare). Secondly, the *exhaustive* assumption can contribute to the explosion of false positives in case normal behaviour is not fully represented, e.g., certain actions, although normal, did not occur at all during the Data Collection. To overcome these drawbacks, we provide a mechanism to allow a security expert to inspect and adapt normal profiles. During the Tuning phase, the expert can set a *threshold* Δ representing the minimum bin probability a bin must have to be considered normal. The threshold is used to label all bins of all histograms of a profile: if the bin probability is lower than or equal to the threshold, the bin is marked as *anomalous*, it is marked as *normal* otherwise.

Definition 29 (Threshold – Δ) *Let $F = \langle f_1, ..., f_n \rangle$ be a feature space and $B = \big\langle \langle b_{f_1 1}, ..., b_{f_1 m_1} \rangle, ..., \langle b_{f_n 1}, ..., b_{f_n m_n} \rangle \big\rangle$ a corresponding sequence of bin sequences, we define a threshold Δ as a sequence of sequences of real values $\in [0, 1] \cup \{-1\}$, one for each bin of each feature in the feature space: $\Delta = \langle \langle \delta_{f_1 1}, ..., \delta_{f_1 m_1} \rangle, ..., \langle \delta_{f_n 1}, ..., \delta_{f_n m_n} \rangle \rangle$. Usually, a threshold is associated to a profile P_c, in which case we denote the threshold by Δ_{P_c}.*

The "special" value -1 for a threshold is used to mark a bin as *normal* "no matter what" and it is used to indicate that the specific bin should never yield an alert. For instance, in case there are no transactions in TS where *rob* accesses the table *age* but, according to the security officer, rob is allowed to do so. Since we need to set a threshold for each existing profile, a practical way to do it is the following. First,

the security officer can set all the element δ_{ij} of Δ to the same fixed value $\bar{\delta}$. We refer to $\bar{\delta}$ as the *global threshold* and it can be seen as a general tolerance towards rare values. For example, by setting $\bar{\delta} = 0$ one assumes that every transaction in the training set is legitimate (attack-free assumption). Second, during the histogram inspection, the security officer can manually tune the threshold for specific bins to the value 1 if the bin has to be considered anomalous, or to a threshold lower than the bin probability if it has to be considered normal (-1 in case the bin has to be considered normal no matter what its bin probability). In this way the expert can have a general rule of thumb (the global threshold) to initially label all the bins, and the fine-grain mechanism to add exceptions to the general rule.

6.7 The Detection & Feedback Loop Phases

The set of profiles – and the corresponding thresholds – resulting from the tuning phase represents normal database usage. During detection, a new incoming transaction T is matched against the corresponding profile, namely the profile P_c for which $c(T) = true$. In case at least one feature value of T falls in a bin with bin probability $prob_{f_i,b_j}$ less or equal to the corresponding threshold $\delta_{f_i,j}$, the transaction will be considered anomalous. To decide whether a transaction T is anomalous or not we need a *detection engine* defined as follows.

Definition 30 (Detection Engine – m) *Let $F = \langle f_1, ..., f_n \rangle$ be a feature space and $B = \langle \langle b_{f_1 1}, ..., b_{f_1 m_1} \rangle, ..., \langle b_{f_n 1}, ..., b_{f_n m_n} \rangle \rangle$ a corresponding sequence of bin sequences. Given a bin $b_{ij} \in B$ where i defines the feature and j a generic bin in the corresponding bin sequence, we define:*

- *a* detection engine *as a mapping $m : \{b_{ij} : i \in \{1, ..., n\},\ j \in \{1, ..., m\}\} \longrightarrow \{anomalous, normal\}$*
- *given a transaction vector T, we say that T is* anomalous *in m iff $\exists b_{ij} : f_i(T) \in b_{ij} \wedge m(b_{ij}) = anomalous$*

We now define the detection engine for a given profile P_c (based on the constraint c), a corresponding threshold $\Delta_{P_c} = \langle \langle \delta_{f_1 1}, ..., \delta_{f_1 m_1} \rangle, ..., \langle \delta_{f_n 1}, ..., \delta_{f_n m_n} \rangle \rangle$, and a training set $TS|_c$. We call this engine $m_{TS|_c}$, and we define it as:

- *$m_{TS|_c}(b_{i,j}) = anomalous$ iff $prob_{f_i,b_{ij}}(TS|_c) \leqslant \delta_{i,j}$;*

A detection engine defines which bins are anomalous and which bins are normal. A transaction is considered anomalous if one (or more) of its features falls into an anomalous bins, it is considered normal otherwise. When a transaction T is flagged

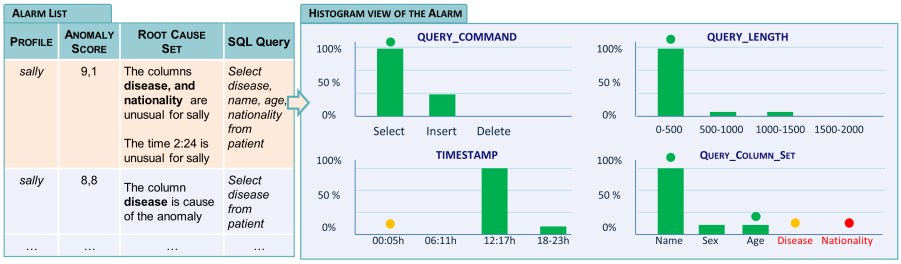


Figure 6.4: Detection Results: How Alarms are Presented to the Security Officer.

as anomalous it is important for the security officer to know which are the causes of the anomaly. To this end we introduce the concepts of *root cause set*, which consists of the transaction's features falling in the anomalous bins hence causing the alarm.

Definition 31 (Root cause set – R) *Given an anomalous transaction vector T, we define the* root cause set *R of the anomaly of T as the set of all the features f_i which feature value falls into a bin b_{ij} for which $m(b_{ij}) = anomalous$, $R = \{f_i : f_i(T) \in b_{ij} \wedge m(b_{ij}) = anomalous\}$.*

For a transaction to be anomalous, it is sufficient that a single feature falls in an anomalous bin; nevertheless, for a security officer is may be important to rank anomalous transactions according to the degree by which they are anomalous. To this end, we introduce the concept of *anomalyscore* which serves to quantify the anomaly's severity and it is computed as follows.

Definition 32 (Anomaly score – $anomalyscore$) *Given an anomalous transaction vector T and its corresponding root cause set R, we define the anomaly score of T as: $anomalyscore(T) = \prod_{f_i \in R} \frac{1}{prob_{f_i, b_{ij}}}$.*

Intuitively, the anomaly score is higher if the corresponding anomalous bins have a low bin probability and if the anomalous transaction has many root causes. When a value falls in a bin with zero bin probability, i.e. unseen values, we use a fixed low probability to avoid division by zero. Given a training set TS, the minimum probability an encountered value can have is $\frac{1}{N}$ where N is the cardinality of TS. Taking a value for MIN_PROB below $\frac{1}{N^n}$ (where n is the number of features in the feature space) would ensure a ranking consistent with taking $1/0 = \infty$ though in practice any sufficiently small value to make the alert stand out will likely suffice. Finally, notice that very low probabilities can lead to very high values for the anomaly score. To solve this, scaling (e.g. logarithmic), normalization or capping of the maximum value

can be applied. In case T is anomalous an alarm is raised, as shown in Figure 6.4. Alarms are listed together with the profile they refer to ($P_{CLIENT\_UID=sally}$ in the example), the *anomaly score*, the *root cause set* and the query text Q. On the other hand, if a transaction is flagged as normal, it is added to the existing profiles. In this way, we have a continuous learning process together with the possibility of adding an aging mechanism, e.g. to forget, or decrease the importance, of older transactions. In this way if a transaction does not take place for a long time, it can eventually be considered anomalous.

6.7.1 Taming False Positives

A high number of false positives is the most common problem of behaviour-based approaches. To deal with such a problem, we introduce the feedback loop mechanism that: i) makes the post-processing of alarms faster; and ii) leverages the security officer's knowledge to improve existing profiles and reduce future FPR. Figure 6.4 shows how alerts can be presented to the officer. Let us consider the first alarm on the left side of the figure, caused by *sally* using the columns *disease* and *nationality*, and submitting the query at *2:24 am*. When the officer selects the alarm, the anomalous values v_{f_i} (red circles) are represented w.r.t. the related profile (histograms) as shown on the right side of the figure. A circle over a bin (bar) means that the feature value falls in that bin. The officer can provide feedback about every alarm, i.e. she can mark each root cause as *true positive* or *false positive*. A true positive means that a specific value represent an actual anomaly and that the officer wants to be warned every time it occurs. A false positive instead means that one of the causes of an alarm is benign and it should not cause an alarm any longer. Circles can be green (for normal values), yellow (for values which still have to be evaluated), and red (for values which have been previously marked as *true positive*, e.g. *nationality*). For example, assuming that *sally* is entitled to use the column *disease*, the officer will mark such value as a false positive, which will cause a refinement of our profiles: the profile *sally* will be updated by incrementing the frequency, thus the probability, of the bin *disease* and by setting the corresponding threshold to -1. This implies that if *sally* uses the column *disease* again, no alarm will be raised (reduction of future FPR). In addition, the other alarms related to *sally* will be updated accordingly, meaning that if another alarm has been caused by the use of *disease*, e.g. the second alarm in Figure 6.4, it will be deleted from the list. In this way a single officer's feedback can result in the deletion of multiple alarms, hence speeding up the alarms' postprocessing.

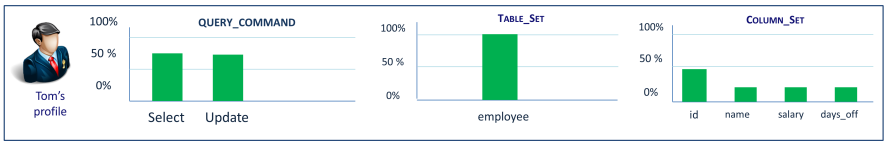


Figure 6.5: Example of Profile for the User *tom*

6.8 The Feature Aggregation Phase

During the Profiling phase discussed in Section 6.5, we assumed features are independent of each other, thus we build profiles containing a histogram for each single feature. However, in certain cases, it might be useful to consider combination of features to improve the detection rate of our solution. For example, let us consider Tom, an employee at the Resource Management department of an organization whose main tasks are checking and printing salary information for all the employees, and to update their days-off balance. Figure 6.5 represent the profile created for *tom* w.r.t. the queries Q.6 and Q.7 (presented below) which are normal for him.

Q. 6 *select salary, name from employee where employee.id* = x*;*

Q. 7 *update employee set days_off* = y *where employee.id* = x*;*

Query Q.8, shown below, is instead an example of malicious query used by Tom to increase his own salary. Such a query should be "rare" and should be considered anomalous (it is at least worth inspecting). However, using the system we introduced so far, the query would be considered non-anomalous. This is because, with the feature independency assumption, we build profiles where it is normal that Tom uses commands as *select* and *update*, and columns as the *salary* and the *days_off*, but we do not keep track of which command is executed on which columns (which would help to detect Q.8).

Q. 8 *update employee set salary = 1357 where employee.id = 003;*

In this section, we introduce the possibility of grouping features together. This allows us to have an even more fine-grained detection engine, which is able to detect attacks like the one we just mentioned. To this end we define a *joint-histogram*, i.e. a histogram referring to a pair of features, as follows:

Definition 33 (Joint-Histogram – h_{f_i,f_j}**)** *Given a training set* TS*, a pair of features* f_i *and* f_j*, and the corresponding bin sequences* $B_{f_i} = \langle b_{f_i 1}, ..., b_{f_i m_i} \rangle$ *and* $B_{f_j} = \langle b_{f_j 1}, ..., b_{f_j m_j} \rangle$ *we define:*

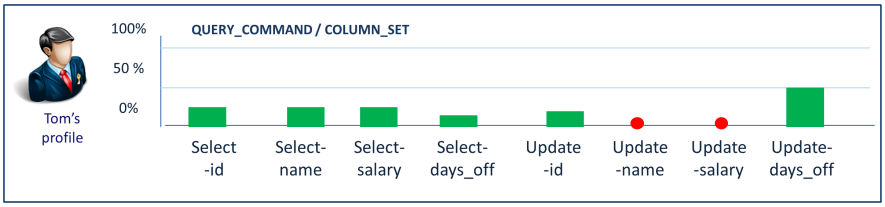


Figure 6.6: Example of *joint-histogram* for User *tom*.

- *a* joint bin sequence, *denoted as* B_{f_i,f_j}, *as the cartesian product of* B_{f_i} *and* B_{f_j}, $B_{f_i,f_j} = \langle b_{f_i1}b_{f_j1}, b_{f_i1}b_{f_j2}, ..., b_{f_im_i}b_{f_jm_j}\rangle$;
- *a* joint bin, *denoted as* $b_{f_ik}b_{f_jz}$, *as a bin in* B_{f_i,f_j};
- *a* joint bin probability, *denoted as* $prob_{f_if_j,b_{f_ik}b_{f_jz}}$, *as the probability of a joint bin* $b_{f_ik}b_{f_jz}$ *w.r.t.* TS;
- *a* joint histogram, *denoted as* h_{f_i,f_j}, *as a sequence of joint bin probabilities,* $h_{f_i,f_j} = \langle prob_{f_if_j,b_{f_i1}b_{f_j1}}, ..., prob_{f_if_j,b_{f_im_1}b_{f_jm_j}}\rangle$;
- *a* joint histogram size, *denoted as* $|h_{f_i,f_j}|$, *as the number of bin in the corresponding joint bin sequence,* $|h_{f_i,f_j}| = |B_{f_i,f_j}|$.

Figure 6.6 shows an example of joint-histogram obtained for the feature pair QUERY_COMMAND and QUERY_COLUMN_SET w.r.t *tom*'s profile of Figure 6.5. Notice that (assuming a threshold of zero) the joint bins *update-name* and *update-salary* are anomalous (their probability is less than or equal to the threshold). In this way, the joint bin *update-salary* will let the system raise an alarm for the transaction Q.8, correctly flagging it as anomalous. Unfortunately, the use of joint-histograms leads to a combinatorial explosion of the histogram size (i.e. number of bins) which might impact performance both in terms of time and false positive rate. To contain the increase of profiles complexity it is necessary to create joint-histograms only for those feature pairs which help to better spot the types of attacks we are interested in. To streamline this process, we propose an automatic way to select feature pairs that are likely to yield an increase in the detection rate. Intuitively, we prefer joint-histograms, hence feature pairs, that maximize the number of anomalous joint bins $b_{f_ik}b_{f_jz}$ similar to the ones in Figure 6.6. These bins have a peculiar characteristic: they are anomalous for the joint-histogram, but the corresponding bins in the originating (single feature) histograms are normal. For example the bin *update-salary* (anomalous

in the joint histogram) is formed by the bin *update* in the histogram for QUERY_COMMAND and *salary* in the histogram for COLUMN_SET which are both normal as shown in Figure 6.5. We are interested in pairing features which lead to joint-histograms with many bins holding such characteristic that we call $anomalous^{++}$. Recall that a bin is anomalous if its bin probability is less than or equal to the corresponding threshold (Definition 30). For sake of simplicity, in the following we assume the threshold value $\bar{\delta}$ is the same for all the bins of all the histograms – the extension to an arbitrary threshold is straightforward – and we define the property $anomalous^{++}$ for a joint bin as follows:

Definition 34 (Anomalous^{++} joint bin – $anomalous^{++}(b_{f_ik}b_{f_jz})$) *Given a training set TS, two features f_i and f_j and the corresponding bin sequences $B_{f_i} = \langle b_{f_i1}, ..., b_{f_im_i}\rangle$ and $B_{f_j} = \langle b_{f_j1}, ..., b_{f_jm_j}\rangle$, let h_{f_i,f_j} be the derived joint-histogram, $B_{f_i,f_j} = \langle b_{f_i1}b_{f_j1}, b_{f_i1}b_{f_j2}, ..., b_{f_im_i}b_{f_jm_j}\rangle$ the corresponding joint bin sequence, and $\bar{\delta}$ the threshold value for each bin in the system. Given a joint bin $b_{f_ik}b_{f_jz} \in B_{f_i,f_j}$ we say that the bin is $anomalous^{++}$ iff:*

$$\begin{cases} b_{f_ik}b_{f_jz} \textit{ is} \text{ anomalous } (prob_{f_if_j,b_{f_ik}b_{f_jz}} \leq \bar{\delta}); \\ b_{f_ik} \textit{ is} \text{ normal } (prob_{f_i,b_{f_ik}} > \bar{\delta}; \\ b_{f_jz} \textit{ is} \text{ normal, } prob_{f_j,b_{f_jz}} > \bar{\delta}. \end{cases} \tag{6.3}$$

A feature pair with many $anomalous^{++}$ joint bins is more likely to detect anomalies which would pass undetected if features f_i and f_j are used independently. Therefore, we evaluate whether two features "pair well" by creating their joint histogram and by counting how many $anomalous^{++}$ joint bins there are in it. To have a measurement of which features "pair well" we compute a *coupling score* for each feature pair (f_i, f_j) w.r.t. our feature space F and our training set TS. The coupling score can be used – before the Profiling phase– to select which features to use as a pair (thus creating a joint-histogram), by selecting only those pairs with a score higher than a certain value. Intuitively, the *coupling score*, defined below, is higher for pairs with a higher number of $anomalous^{++}$ bins in the corresponding joint histogram.

Definition 35 (Coupling Score (χ_{f_i,f_j})) *Given a training set TS, two features f_i and f_j, the corresponding joint-histogram h_{f_i,f_j} and the corresponding bin sequence $B_{f_i,f_j} = \langle b_{f_i1}b_{f_j1}, b_{f_i1}b_{f_j2}, ..., b_{f_im_i}b_{f_jm_j}\rangle$, we define the* coupling score *of f_i and f_j, denoted as χ_{f_i,f_j} as the number of $anomalous^{++}$ joint bins over the total number of bins in the joint histogram $|h_{f_i,f_j}|$:*

$$\chi_{f_i,f_j} = \frac{\sum\{b_{f_ik}b_{f_jz} \in B_{f_i,f_j} : b_{f_ik}b_{f_jz} \textit{ is } anomalous^{++}\}}{|h_{f_i,f_j}|} \tag{6.4}$$

A higher coupling score means a higher anomalous bins rate, hence a higher alarms rate. This alarms rate can increase both the detection rate and the false positive rate. The cost-benefit analysis of applying such an approach is discussed in Section 6.11. Finally, note that the definition of joint-histogram could be easily extended to more than two features, but we left this extension out of the scope of this work.

6.9 Evaluation Methodology

To evaluate our framework, we implemented it as a RapidMiner[8] extension. RapidMiner is a java-based open-source system for data mining, providing an extension mechanism that allows the insertion of customised modules to integrate with existing functionalities. In the following sections we describe the experiments we carried out and the results we obtained. For all the experiments we applied a common methodology: we gathered a dataset consisting of several database transactions and we divided it into a Training Set ($\sim 70\%$ of the dataset), used to learn the profiles, and a Testing Set (the remaining $\sim 30\%$), given as input to the detection engine. The Testing Set allows us to measure the FPR: under the attack-free assumption (global threshold $\bar{\delta}$ set to zero) every alarm raised when the Testing Set is given as input to the detection engine can be considered a false positive. Beside the training and the testing set we also have distinct Attack Sets containing malicious transactions. In our experiments we measure the false positive rate *FPR* and the detection rate *DR* as follows:

- *FPR*=(*#alarms raised on the testing set)/(cardinality of the testing set*)
- *DR* = (*#alarms raised on the attack set)/(cardinality of the attack set*)

For our experiments we use two different datasets:

1. the **Enterprise Dataset (ED)**: is taken from the log of an (Oracle) operational database of a large IT company. The database is accessed through a web application internal to the organization, which has about 100 users, owning 4 different roles. To create the dataset, the DBMS auditing facility was enabled to collect a total of 12,040,910 transactions (8,428,637 in the Training Set and in the 3,612,273 Testing Set).

2. the **Simulated Dataset (SD)**: is constructed using the healthcare management system GnuHealth (`http://health.gnu.org`). We simulated normal behaviour (validated by domain experts) consisting of an *admin* and different

[8] `http://rapidminer.com/`

Table 6.5: Features Extracted for our Experiments.

	Feature	White-Box without RS	(1)White-Box with RS (2)Naïve Bayes (3)Decision Tree	Kamra et al. (c-quiplet)	Kamra et al. (m-quiplet)	Kamra et al. (f-quiplet)	Wu et al.
Syntax Centric	QUERY_COMMAND	✓	✓	✓	✓	✓	✓
	QUERY_LENGTH	✓	✓				
	QUERY_COL_SET	✓	✓			✓	
	QUERY_COL_NUM	✓	✓	✓	✓		✓
	QUERY_TABLE_SET	✓	✓		✓	✓	✓
	QUERY_TABLE_NUM	✓	✓	✓			
	SELECT_ALL	✓	✓				
	WHERE_TABLE_SET	✓	✓		✓	✓	✓
	WHERE_TABLE_NUM	✓	✓	✓			
	WHERE_COL_SET	✓	✓			✓	
	WHERE_COL_NUM	✓	✓	✓	✓		
	WHERE_LENGTH	✓	✓				
	SPEC_CHAR	✓	✓				
Context Centric	OS_USERNAME	✓	✓				
	CLIENT_APP_UID	✓	✓	✓	✓	✓	✓
	CLIENT_APP_ROLE	✓	✓	✓	✓	✓	✓
	RESPONSE_CODE	✓	✓				
	TIMESTAMP	✓	✓				✓
	IP_ADDRESS	✓	✓				✓
Result Centric	BYTES_NUM		✓				
	ROWS_NUM		✓				
	DISEASE		✓				
	PATIENT		✓				
	DOCTOR		✓				
	EMAIL		✓				
	LOGIN		✓				
	PWD		✓				

users of a hospital, where *doctors* take care of *patients* suffering from different *diseases*. Ordinary behaviour includes doctors and nurses making prescriptions and accessing patients data, and the administrator inserting new users and managing users' accounts. We implemented – and added to the GnuHealth source code – a customized logging module able to intercept each transaction with the database. The Simulated Dataset contains a total of 65,291 transactions (45,704 in the Training Set and 19,587 in the Testing Set).

6.9.1 Comparing Diffferent Approaches

To demonstrate the improvements our solution brings to similar state-of-the-art systems in the field of database anomaly detection, we compare the results we achieve with those obtained by other existing approaches, namely the solutions proposed in Kamra et al. [185] and Wu et al. [187]. For the comparison we reproduced their solutions and we tested them on the same datasets (ED and SD) used for our experiments. For all the tested solutions we measure the FPR and the DR as described above. The approaches in [185, 187] work by creating a Naïve Bayes classifier which, given a set of features, learns to predict the value of the *userid* or the user *role*. For each new transaction, both solutions raise an alarm if the label predicted by the classifier dif-

fers from the actual label of the query. The main difference between [185] and [187] is given by the features used to build the classifier. Table 6.5 presents the features extracted for our experiments –White-Box solution with and without Result Set(RS) – and for the other tested approaches. Note that Kamra et al. only use syntax-centric features while Wu et al. also consider context-centric features (e.g. the *timestamp* and *ip address*). The Kamra et al. solution has three variants – c-quiplet, m-quiplet and f-quiplet – which differs in how fine grained their feature space is; e.g. the c-quiplet only considers the number of columns and tables, while the f-quiplet accounts for the exact columns and tables, making the latter retain a lot more information.

Beside the algorithm used to learn profiles, our approach differs from the others because of the extended feature space we use. To test whether not only the algorithm but also the feature space impacts the performance of a detector, we test an additional approach, the Naïve Bayes algorithm as used in [185, 187] but with the same extended feature space used in our solution.

While RS information is not available for the Enterprise Dataset, in the Simulated Dataset we have access to result-centric features, especially the *number of bytes* and *rows* for each transaction, together with the values retrieved for specific sensitive columns namely *disease*, *patient*, *doctor*, *email*, *login* and *pwd*. To allow for a better comparison between the results obtained with RS and without, in our experiments we made two separate measurements for the two cases (with and without RS).

6.9.2 ROC Curves

To measure the performance of our solution and to compare it with other approaches we use receiver operating characteristic (ROC) curves [201], allowing to graphically illustrate the performance of a detector. Generally, a detector performs well if it shows both a low FPR (costs) and a high DR (benefits). FPR and DR depend on the *true* and *false* alarms raised by a detector. Recall that an alarm is raised if $anomalyscore > 0$ in our solution, and if $prediction\ != actual\ class$ for the solutions in [185, 187]. To plot ROC curves we measure FPR and DR for varying values of a decision threshold t – note that this is not the threshold discussed in Section 6.6. The threshold t is used to vary the output of the detectors as follows: for our solution an alarm will be raised if $anomaly\ score > t$, while for [185, 187] an alarm will be generated if $prediction\ != actual\ class$ and the $prediction\ probability > t$. To compare different detectors we plot their ROC curves in a single graph: the best detector is the one in which the ROC curve passes closest to the upper left point (0,1) (zero costs, maximum benefits). In case ROCs intersect, it might be difficult to visually spot which method performs better, thus we reduce ROC curves into a single scalar value: the AUC (Area Under the Curve) [202] which measures the area under the ROC curve and which is widely used as a quantitative summary of the performance of a detector.

The AUC value that ranges from 0 to 1: an AUC=1 means the detector is excellent, while values less than 0.5 mean the detector performs worse than a random guessing. Note that outside the ROC context we use the default value for the threshold which is zero for all the tested approaches and which leads to the best FPR-DR tradeoff.

Finally, although our system is devised from scratch to include a feedback loop, to guarantee fairness we make no use of it when comparing our results with the other approaches. In the coming sections we describe the results obtained by making the three following experiments:

- **Experiment 1**: we measure the FPR and the DR of our solution on the ED and SD datasets (Section 6.10);
- **Experiment 2**: we test the impact of using aggregated features. The goal is to demonstrate that aggregated features improve the detection rate without increasing the FPR (Section 6.11);
- **Experiment 3**: we test the capability of the transaction flow profiling of detecting advanced threat which are spanned over multiple transactions and multiple time frames (Section 6.12).

6.10 Experiment 1: Baseline System Evaluation

In this section, we experimentally validate our baseline framework by addressing the following questions:

- How effective is our framework. To this end, we made an experimental evaluation of its Detection Rate (DR) and its False Positive Rate (FPR).
- How does it compare with existing solutions, e.g. [185, 187] and with standard machine learning approaches (e.g. Naïve Bayes). To help the comparison we use Receiver Operating Characteristics (ROC) curves [201] and we measure the Area Under the Curve (AUC) [202].
- What is the added value of the feedback mechanism. We test this by measuring the impact a feedback operation has in reducing the FPR.

Attacks Description. To answer these questions we use the ED and the SD datasets to learn the profiles and to measure the FPR. In addition, for each dataset we have one or more Attack Set which, given as input to the detection engine, serve to test the DR of our solution (how many attacks are detected). The Attack Sets used in this experiments are the following:

- **ED Attack Set 1 (ED-ATK-1)**: this Attack Set contains 107 malicious transactions w.r.t. the Enterprise Dataset. The transactions have been devised and executed by the security administrator of the database, logged in as one of the profiled users. These malicious transactions represent real threats to the enterprise since they try to access information normal employees should not have interest in, like organizational-sensitive data or other employees' password or userid (e.g. *select * from dba_tab_columns where column_name like '%PASSWORD%';*);
- **SD Attack Set 1(SD-ATK-1)**: this Attack Set contains 277 malicious transactions w.r.t. the Simulated Dataset. It contains the database actions of a malicious admin who accesses parts of the database (e.g. the table *patient* or *disease*) to which she should have no interest in;
- **SD Attack Set 2: (SD-ATK-2)**: this Attack Set contains 26 malicious transactions w.r.t. the Simulated Dataset. The transactions refer to the actions of a malicious GP (General Practitioner) who queries for data of patients with *HIV, cancer, and mental disorder*, while she usually only treats patients with generic disease (e.g. *flu*, *cough*).

Note that ED-ATK-1 and SD-ATK-1 can be considered *syntax-based attacks* since they include transactions which access columns and tables to which the users, although have the right, should not be interested in. On the other hand, SD-ATK-2 can be considered a *result-set-based* attack since the anomalies are due to the values retrieved and not to the query text.

6.10.1 Results

Table 6.6 compares the results of the different solutions we tested over the Enterprise Dataset. The FPR obtained with our solution (without RS) is only 0.003% (116 false alarms over about 4 millions transactions) against the 38.53% of the second best solution (Kamra et al., f-quiplet). The detection rate is also very high (99.07%) with our engine able to detect 106 out of 107 attacks. Although the DR is not the highest for our solution (all the other approaches detect all the attacks), our approach is the one with the best FPR-DR tradeoff, as shown by the ROC curves in Figure 6.7 and by the highest AUC value (0.995). This means we can provide the best benefit at any reasonable cost, hence considerably reducing the officer's work load.

Results related to the Simulated Dataset are shown in Table 6.7. On the Simulated Dataset, we could test the performance of our solution in two configurations: with and without RS features. The results confirm the good performance of our solution in terms of FPR (0.50% without RS and 0.53% with RS) which is basically two

Table 6.6: Results for Enterprise Dataset (*userid* Profiles)

	FPR	**DR (ED-ATK-1)**	**AUC**
White-Box (without RS)	**0.003%**	99.07%	0.995
White-Box (with RS)	*na*	*na*	*na*
Kamra et al. (c-quiplet)	81.38%	100%	0.648
Kamra et al. (m-quiplet)	56.85%	100%	0.454
Kamra et al. (f-quiplet)	38.53%	100%	0.623
Wu et al. (standard)	58.13%	100%	0.434
Naïve Bayes (all features)	57.97%	100%	0.748

Table 6.7: Results for Simulated Dataset (*userid* Profiles)

	FPR	**DR(SD-ATK-1)**	**DR(SD-ATK-2)**	**AUC**
White-Box (without RS)	**0.50%**	100%	3.85%	0.954
White-Box (with RS)	**0.53%**	100%	88.46%	0.991
Kamra et al. (c-quiplet)	64.11%	94.58%	88.46%	0.588
Kamra et al. (m-quiplet)	56.74%	99.64%	76.92%	0.756
Kamra et al. (f-quiplet)	50.31%	100%	73.08%	0.630
Wu et al. (standard)	46.57%	100%	38.46%	0.732
Naïve Bayes (all features)	23.69%	100%	57.69%	0.813

orders of magnitude better (lower) than that of the other compared approaches. The best performance of the other tested approaches is obtained by using Naïve Bayes with the same feature space used for our solution (FPR= 23.69%). Recall that we also use Naïve Bayes with a smaller feature space, i.e. in the solutions proposed in [185, 187]. This suggests that extending the feature space can improve general performance, probably because of the finer grain of the profiles (more features, more information is kept).

The White-Box variant with RS features allows us to detect many of the attacks present in the attack set SD-ATK-2 (which were devised explicitly to be detectable in presence of RS information) at the cost of a slightly increases the FPR (+0.03%). With these settings, it is not surprising that the our solution without RS has a low detection rate (3.85%): it misses the potential of detecting result-centric attacks. What may look surprising is that DR2 is very high for the other approaches: in principle, none of them – being syntax-based – should be able to detect this. On the other hand, when we add RS features we obtain the highest DR together with the c-variant of Kamra et al (88.46%). Overall, over the SD, our solution with RS is the one offering the best FPR-DR tradeoff (AUC= 0.991) as also shown by the ROC in Figure 6.8 (the DR for SD-ATK-1 and SD-ATK-2 are combined to plot the ROC).

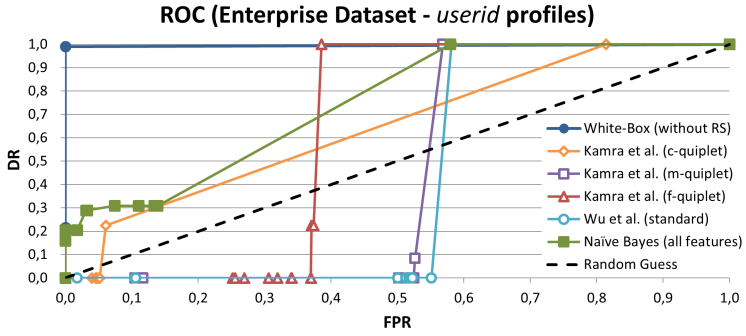


Figure 6.7: ROC Curves Comparison - Enterprise Dataset (*userid* profiles).

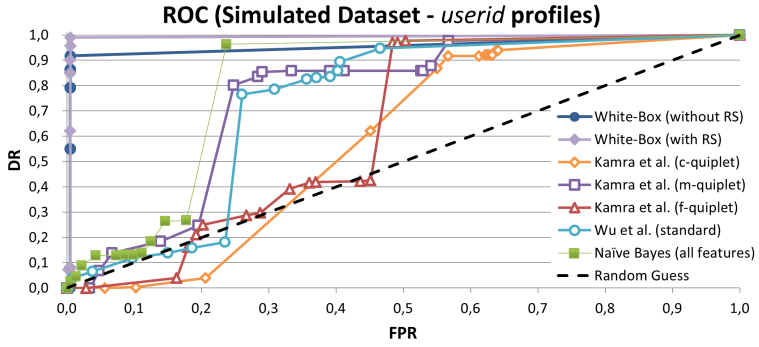


Figure 6.8: ROC Curves Comparison - Simulated Dataset (*userid* Profiles)

Discussion over Naïve Bayes. In this section we want to discuss about the reasons why the solutions we used for comparison show such a high FPR. All the comparing solutions adopt a Naïve Bayes algorithm which we reproduced in our laboratory as described in [185, 187]. Naïve Bayes is a standard machine learning algorithm that does not need the definition of input parameters (its simplicity is the reason why it is widely used). Unfortunately, the algorithm is also well-known for not being resistant to heavily unbalanced dataset [203]. In case there is a class (in our experiment a class is equal to the *username*) which is very unbalanced (e.g. too few query coming from a certain user), the Naïve Bayes classifier will create a model which is not reliable. This is known as the skewed data problem: more training examples for one class

Table 6.8: Feedback Impact on FPR (*userid* Profiles).

	Feedback Operation	Enterprise DS		Simulated DS	
		FPR (%)	FP (#)	FPR (%)	FP (#)
White-Box (without RS)	0	0.0032%	116	0.500%	98
	1	0.0032%	115	0.495%	97
	4	0.0030%	107	0.470%	92
	7	0.0027%	99	0.439%	86
White-Box (with RS)	0	*na*	*na*	0.526%	103
	1	*na*	*na*	0.521%	102
	4	*na*	*na*	0.485%	95
	7	*na*	*na*	0.470%	92

than another can cause the classifier to unwittingly prefer one class over the other. The effects of this problem are also shown by the results presented in [185]. These results are obtained over a dataset automatically generated using a zipf probability distribution. The zipf distribution depends on the parameter s, that for values $s \leq 1$ leads to highly unbalanced distribution. In these settings ($s \leq 1$) the results presented in [185] show an FPR between 20% and 60%. The datasets used in our experiments are heavy unbalanced: this justifies the bad performance obtained by Naïve Bayes-based solutions and suggests that, being real-world data generally skewed – certain roles or users can be more active than others– Naïve Bayes might likely not perform well in this domain.

Impact of the Feedback Loop.

A novel component of our framework is the feedback loop mechanism that improves for false alarms handling and iteratively reduces the FPR by updating existing profiles. We want to stress again that all the results shown in the previous section did not take advantage of this feedback mechanisms to ensure a more fair comparison with the other systems in the literature [185, 187]. Table 6.8 shows how the feedback loop mechanism can improve the FPR. The table refers to the profiles created for the Enterprise Dataset and the Simulated Dataset. In the case of ED, with 7 feedback operations we are able to reduce the FPR from 0.0032% to 0.0027% (from 116 to 99 alarms). On the simulated dataset, 7 feedback operations reduce the FPR from 0.50% to 0.44% and from 0.53% to 0.47% respectively in the solution without and with RS. Note that the feedback loop mechanism has a higher impact in reducing the FPR when the training set is too small to represent all the common activities, thus it is more likely that new transactions (i.e. not yet observed) will pop up during detection.

In our test cases (especially in the case of the Enterprise Dataset) the training set is large enough thus the feedback loop impact is not as effective as in cases with smaller training set (see results in [13]). It is worth mentioning that a single feedback operation consists of only few mouse clicks: those necessary to mark an alarm as false positive.

6.11 Experiment 2: Feature Aggregation Evaluation

In this section we describe the tests to evaluate the effectiveness of the aggregated feature mechanism. In particular we want to measure how much this new mechanism helps to detect new attacks and how it affects the false alarm rate.

ED Attack Set 2 (ED-ATK-2). The attack set used in this experiment is formed by 10 malicious transactions executed over the Enterprise Dataset. To obtain this attack set we asked the security officer of the ED database to craft an attack which could bypass the detection per single feature. He simulated an attack where a malicious system administrator retrieves the password of another system administrator (*Scott*), logs in with the victims credentials and copies the customer list. Finally, the malicious admin tampers with the audit table to delete evidence of such misbehaviour.

While the attacks were crafted by an external party, we independently computed the coupling score over the ED training set to find the features which best pair together. We selected only the pairs with a score higher than 0.98, namely the pairs (QUERY_COMMAND, QUERY_COLUMN_SET) and (QUERY_TABLE_SET, QUERY_-COLUMN_SET) and for them we built the joint-histogram. As usual, we give the attack set and the testing to the detection engine to measure DR and FPR respectively.

6.11.1 Results

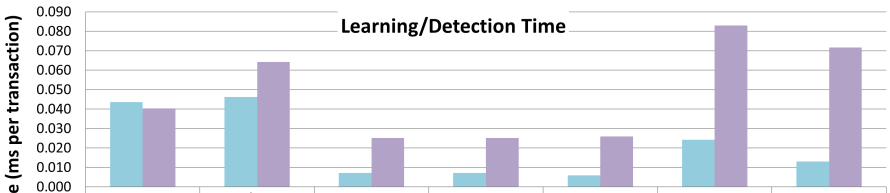
Table 6.9 shows the results we obtain with these settings. To ease the understanding of the impact of feature aggregation, in the table we show again the results obtained when no aggregation is used (standard White-Box approach). With the standard White-Box approach (1st row of the table) we miss the detection of one malicious query over ED-ATK-1(namely, the query "*select * from sys.user$*"). This happens because the user normally accesses table *sys.user* and makes *select all* (on other tables). With the use of aggregated features (2nd row of the table) our approach is now able to detect 100% of the malicious transactions, thus overcoming one of its limitations. The detection over the new attack set (ED-ATK-2) is also very high: we are able to correctly detect 100% of the malicious transactions with almost no impact

Table 6.9: FPR and DR Analysis with Aggregated Features.

Model	DR (ED-ATK-1)		DR (ED-ATK-2)		FPR	
	%	#	%	#	%	#
White-Box (standard)	99.07%	106	0.0%	0	0.003%	116
White-Box(aggregated features)	100.0%	107	100.0%	10	0.003%	121
Karma-Bertino (c-quiplet)	100.0%	107	100.0%	10	81.385%	2939772
Karma-Bertino (m-quiplet)	100.0%	107	90.0%	9	56.864%	2053489
Karma-Bertino (f-quiplet)	100.0%	107	70.0%	7	38.533%	1391586
Wu et al. (standard)	100.0%	107	70.0%	7	58.135%	2099908
Naïve Bayes (all features)	100.0%	107	100.0%	10	57.971%	2093989

on the FPR (only 5 alarms more). The other approaches we tested are able to detect most of the transactions in ED-ATK-2 with a DR that can arrives to 100% as well. However the cost that is necessary to pay in terms of FPR is very high, as already discussed.

Introducing the aggregated-features capability adds complexity to our system due to the explosion of the number of bins in the joint-histograms. This added complexity could increase the time needed to learn the model and the time needed to detect whether a transaction is malicious or not, depending on the number of feature pairs added. Figure 6.9 shows the learning and detection time (in millisecond) for our approach (with and without aggregated features) and for the other tested solutions. The time refers to the average time needed per single transaction. It has been obtained as average of 10 iterations and it is measured over the Enterprise Dataset (training set for learning, testing set for detecting). The results indicate that our solution (in the standard variant and with aggregated features) generally requires more time to carry out the learning phase and to decide whether a transaction is malicious or not when compared to the other solutions. Learning time and Detection time are both around 0,04 ms per transaction. This time is still reasonable if we consider that our system has been developed as a research prototype with no particular attention to time performances and parallelization. In addition, the aggregated-features do not add much overhead for learning, they just delay the detection by about 0,02 ms. Finally, for the other solutions (except Wu et al.) learning time is much faster (about 0,01 ms) and detection is almost twice as fast (0,02 ms).



	White-Box(standard)	White-Box(aggregated features)	Kamra et al. (c-quiplet)	Kamra et al. (m-quiplet)	Kamra et al. (f-quiplet)	Wu et al. (standard)	Naïve Bayes (all features)
Learning	0.044	0.046	0.007	0.007	0.006	0.024	0.013
Detection	0.040	0.064	0.025	0.025	0.026	0.083	0.072

Figure 6.9: Time Analysis for the Different Approaches.

6.12 Experiment 3: Transaction Flow Profiling Evaluation

In this section we evaluate the benefits introduced by the use of the transaction flow (TF) profiling. Transaction flow profiles help to detect attacks where a malicious user submits transactions which are normal if seen as a single unit, but are malicious as a group. In this experiment we use GnuHealth to simulate the malicious scenarios described below. We have chosen to use GnuHealth because it allows to simulate complex attack scenarios which are not easy to reproduce in an operational environment (as the one where the Enterprise Dataset comes from).

Attack 1 (TF-ATK-1): admin steals users email list. Bob is an administrator of a healthcare system. He knows email addresses can be sold on the black market and he decides to copy as many e-mail as he can from the healthcare system database. As daily tasks Bob adds new users to the system, accesses the user table several times and sends a couple of emails to users for general maintenance. The day of the attack Bob works for 8 hours and he accesses the email address of each and every user of the system, copies them to a file and brings the file home. Note that accessing one user is part of normal activities, while accessing all of them in one day represents the malicious activity we aim to detect.

Attack 2 (TF-ATK-2): doctor leaks data of patients with HIV. Robin is a GP at a hospital. As a daily activity, he processes patients's medical data. He usually treats up to 5 patients per day and he occasionally accesses data of patients with HIV. A pharmaceutical firm is interested in testing a new drug for HIV. They contact Robin

and offer to pay him for providing them the personal details of people which might be interested in taking part to such tests. The day of the attack Robin works for 6 hours and he accesses the data of all patients who have been diagnosed HIV positive.

We simulated both scenarios for a time lapse of 16 days. Each day we logged in as Bob and Robin and we performed their normal daily activities as described above. In the last day, we simulated the attacks, which we kept separated from the rest of the dataset. Although we simulated these specific scenarios for 16 days, the overall dataset contains transactions spanning over 239 days. This happens for the following reasons: we had a first phase of simulation in June 2013, and a second one, where we simulated the transaction-flow scenarios, in February 2014. This resulted in having 239 days of activities (the time passing between the two phases). Note that the days between the two phases have no queries. We executed our experiment with three different scopes:

- *per single transaction*: we use the same approach analyzed in Experiment 1 (Section 6.10). Here, we test whether the single user-id profiles are able to detect the attacks presented above.

- *per user per-hour*: a LTG containing all the transactions resulting from the same user is produced after each hour (starting from the *timestamp* of the first transaction) for a total of 5538 hours, 3877 LTG for training and 1661 for testing.

- *per user per-day*: a LTG is generated after each day for a total of 239 days, 167 days used for training and 72 used for testing.

For both scenarios, when learning the transaction-flow profiles, we used the features listed in Table 6.4 and we set columns *email*, *password* and *pathology* as sensitive for the feature '#of sensitive columns', and the values *HIV*, *cancer* and *mental disorder* as sensitive for the feature '#of sensitive values'. In this way, during the profiling we can learn how many sensitive columns and sensitive values a user accesses in an hour or in a day, and detect any deviation (e.g. too many sensitive values are leaving the database).

6.12.1 Results

Table 6.10 shows the detection rate obtained with the different scopes for both attacks scenarios. As we can see, the single transaction profile is not able to detect TF-ATK-1 and TF-ATK-2. This is because the attacks are composed of transactions which are normally executed (what is anomalous is their frequency). The scope per-hour is

Table 6.10: Results for Transaction Flow Detection.

Scope	DR TF-ATK-1		DR TF-ATK-2	
	(#)	(%)	(#)	(%)
single transaction	0	0.00%	0	0.00%
per user per hour	0	0.00%	6	100.0%
per user per day	1	100.0%	1	100.0%

still not able to capture TF-ATK-1 (DR=0%) but it is able to detect TF-ATK-2 (DR = 100%). This means the per-hour scope is too small (in time) to detect the misuse from Bob; on the other hand Robin made more transactions than usual even if we consider the hourly activities, therefore we are able to detect his misbehaviour even with this scope. Finally, the per-day scope is able to detect both attacks (DR 100%).

The experiments show that the definition of the scope is very important to detect transaction flow attacks: if it is too small (e.g. per-hour) it might fail to detect some attacks, while if it is too large a dataset collected during a larger frame of time is necessary to build meaningful profiles. Fortunately our framework allows the definition (and simultaneously execution) of profiles with different scopes, therefore the detection can be done with different granularity.

6.13 Limitations

The solutions we discussed in this chapter boosts the practical applicability of behavior-based techniques to detect data leakages in real scenarios. However, the solution is far from being perfect and it presents several limitations that should be addressed in the near future. First of all, our framework does not yet contain a mechanism to deal with transactions that have temporal constraints. For example, there are cases in which a certain type of transaction is only normal as end-of-the year activity (e.g. for financial budgeting and auditing). The same transaction, if done at any other time of the year, should be anomalous. A way to deal with this situation could be to attach constraints to some transactions (e.g. at tuning time or when giving feedback about an alert). Another limitation, is that currently we only use a lower bound threshold (values below the threshold are considered anomalous) but we do not use an upper bound threshold (value above a certain threshold should be considered anomalous as well). The introduction of a upper bound threshold would be useful especially in the transaction-flow profiling, since it could help to detect attack such as data trawling.

6.14 Conclusions

In this chapter we presented a white-box behaviour-based solution to database leakage detection. Our solution creates fine-grain histogram-based profiles of database usage, and detects anomalies when the features of a transaction assume strange or previously unseen values. The solution we propose provides many improvements w.r.t. existing work: first its white-box nature allows the creation of profiles and detection rules which are easy to understand and modify; second the fine-grained profiles – enabled by the white-box structure – yield a false positive rate that is substantially lower than in competing approaches. It is worth stressing that the FP rate is usually the main "problem" that limits the practical adoption of behaviour-based approaches. The results of our evaluation process show high detection rate for different kinds of attacks – both syntax and data related – while keeping the number of false positives low. In addition, the introduction of the aggregated-features and the transaction flow analysis allows the detection of more advanced threats, namely transactions which are anomalous for a combination of features values and transactions which are anomalous as a group. The main results, as confirmed by our evaluation, can be summarized as follows:

- Our framework is effective in detecting different types of database leakages, while achieving FP rates that, on the same dataset, are much better than the approaches we used for comparison. In particular, we have a FPR of 0.003% over the Enterprise Dataset, and 0.50% over the Simulated Dataset while the other approaches never show a FPR under 20%.

- The possibility of aggregating features allows us to detect anomalous transactions for which each individual feature by itself is normal but it is anomalous if regarded in combination with other features. We also provide an automatic way to select which features are worthy to aggregate to improve the detection rate. Our experiments demonstrate that this new capability makes it possible to detect the threats that are reflected only in combination of features without considerably decreasing the performance of our solution both in terms of time and FPR.

- The transaction flow analysis allows the detection of more advanced threats, namely those spanned over multiple transactions. We give the possibility to set different scopes to monitor activities in terms of groups of transactions. The experiments demonstrate the effectiveness of this technique.

Chapter 7

Concluding Remarks

In this chapter we provide a summary of the main results we achieved in relation to the research questions presented in Chapter 1 and a discussion about open questions and directions for future work.

7.1 Summary of Results

The goal of this thesis is to address the problem of data privacy protection. It tackles it by performing an analysis of the privacy risks that may occur at different stages of what we have called the data cycle. The data cycle is defined as the typical path followed by personal data from the moment it leaves the user's premises until it is stored in data repositories. We now recall the research questions we formulate in Chapter 1, their context, and how they are addressed in this thesis.

The first question has to do with the risks users face when deciding whether to trust a website or an online service which may prove to be fraudulent or not compliant with users' privacy protection expectations. To help users in making trust decisions which reduce regret, we first need to understand how trust decisions are made.

> **RQ.1** *What are the factors that a user takes into account before deciding to trust a website and what can be done to avoid misjudgments which could cause privacy losses?*

To answer this question, in Chapter 2 we propose the General Trust Perception

Model (GTPM) which describes how trust decisions are generally made and provides guidelines on how to drive such decisions. The GTPM is focused on initial trust – no previous interaction between the user and the website is assumed. We argue that a trust decision is made according to a user's perceived trustworthiness of a website. The perceived trustworthiness, in turn, depends on certain characteristics of the user (e.g., her disposition to trust) and on the user's perceived values for the website's factors of trust. The list of factors of trust has been created by consulting a wide body of literature which analyzes trust relationships in different environment (human-to-human, human-to-machine, etc.). Examples of such factors are reputation, security, privacy, and brand name. The model suggests that, to reduce regret caused by bad trust decisions, it is necessary that i) perceived values for the trust factors are as close as possible to objectives values (reducing the Trust Indicator Gap); and ii) users give the right importance to the trust factors that actually have an impact on their cyber risks exposure (reducing the Factor Importance Gap). To determine to which factors the users give more importance when it comes to make trust decisions, we carried out a user study involving 335 respondents. The study revealed that users care a lot about factors such as the brand name of a service, its reputation and its reliability ("it does what it is supposed to do"). On the other hand, factors such as privacy and security are generally not considered as important. The results of the user study also suggested that, in some cases, the indicator used to communicate the value of a factor of trust is not effective. This is the case of privacy policies: although they contain valuable information, users refuse to read them due to their complexity. This is an example of Trust Indicator Gap. To reduce the Trust Indicator Gap, it is necessary to create indicators which are well representative of a certain factor of trust. To this end, and by focusing on the privacy factor, we formulated our second research question as follows:

RQ.2 *Can we evaluate the privacy level of a website by automatically analyse its natural language privacy policy?*

We answer this question with the solutions proposed in Chapter 3 and Chapter 4. In Chapter 3 we present a new metric to measure the privacy completeness of a policy. In this metric, the privacy completeness is evaluated in terms of how many privacy categories are covered. Privacy categories are in turn taken from privacy regulations and guidelines. Example of privacy categories are *Data Collection*, *Data Sharing*, and *Retention Time*: the first refers to the description of what personal data is collected for what purpose, the second refers to the rules followed by the website to share personal data with partners and third parties, while the last one refers to the period of time for which personal data will be kept. To evaluate the completeness of privacy

policies written in natural language, we use machine learning techniques. Given a corpus of privacy policies, we trained a classifier to associate to each paragraph of the policy a specific privacy category. Validation results show the feasibility of our approach; an automatic classifier is able to associate the right category to paragraphs of a policy with an accuracy up to 92% – approximating that obtainable by a human judge. Once the privacy policy has been structured into categories, its completeness can be graded, and a user can easily access the text related to a category of her interest.

The completeness grade gives a measure to inform the user on how complete a privacy policy is. However, it does not provide information about the semantics of the policy. For example, a website may have a high rate of completeness because it thoroughly describes *what* data is gathered, but users may find that the *amount* or *sensitivity* of data collected is excessive for the service provided (e.g. the social security number is required to register to a blog).

To address this problem, we define a metric based on the semantic value of a privacy policy. Specifically, in Chapter 4 we propose a solution to extract the list of all personal data items collected by a website. The solution is based on the use of Information Extraction techniques which take advantage of the strong formality and fixed patters of privacy policies. We define a general semantic model of privacy polices based on how the sentences regarding the *data collection* category are formulated. The model has been built by analyzing dozens of policies of the most used websites. We validated our approach over a set of manually labeled policies. The results show that it is possible to reach a high accuracy – around 80%– in terms of correctly identified data items. The same approach can be easily extended to other categories of privacy in order to obtain a comprehensive semantic grade for a policy.

Going further along the data cycle, we run into the problem of adapting the service offering to the privacy wishes of the user. Having a privacy-preserving service selection mechanism can be a great aid in creating a market for privacy-conscious service providers. This is reflected in the following research question:

> **RQ.3** *How can we identify the service composition which best preserves privacy and best matches a user's preferences?*

Our answer to this question is provided in Chapter 5, where we propose a formal fine-grained model for users to express privacy preferences and for service providers to express their privacy policy. We also define an algorithm allowing a service orchestrator to select only those service compositions which comply with the user's preferences. For the case in which several compositions satisfy users preferences, we also defined a ranking mechanisms which allows to choose the composition which is more privacy-preserving.

Having analyzed the problem of supporting users in choosing online services which offer the most appropriate level of privacy protection, we finally moved our focus towards the analysis of privacy risks that can arise when data is at rest in data repositories. This has lead to the definition of our last research question, namely:

> **RQ.4** *How can we monitor access to data repositories containing sensitive information in order to detect privacy infringements such as data leakages and misuses?*

In Chapter 6 we answer this question by providing a novel technique to detect unusual database activities. Our solution observes traffic from and to the database and builds profiles of normal database usage. The profiles we build have the following innovative characteristics: i) they are fine-grained, i.e. they are built over a wide set of database transactions features such as the query command, the user id, the tables and columns involved as well as the result set; ii) they are white-box, namely they are easy to inspect, understand and modify as opposed to most of the existing profiling techniques which use complex networks and equations difficult to interpret by a human being; iii) they can be updated in-line, meaning that database transactions can be incrementally added to the profiles with no need of retraining. In addition, our solution embeds a feedback loop mechanism which eases the handling of false alarms and leverages the security officer's knowledge to improve existing profiles and reduce future FPR. We validated the system using a simulated as well as a real dataset, the latter containing millions of database transactions. The experiments show that our solution has a very high detection rate, and it is able to detect several different types of attacks. In addition, and most importantly for anomaly-based solutions, the experiments indicate a very low false positive rate – around 0.003%, which is extremely low when compared to similar solutions. Finally, our system is the first of its type to allow the detection of attacks spanned over multiple database transactions, e.g., if a large amount of sensitive data is leaked along a prolonged period of time to avoid detection.

7.2 Implications for Researchers and Practitioners

The solutions proposed and discussed in this thesis can widely benefit both researcher and practitioners in the privacy and data protection field. Incidents such as Snowden and the NSA case are increasing user awareness of privacy concerns. In addition, the EU is developing a body of privacy directives that work towards the empowerment of the user, such as the "cookie regulation" that requires websites to get consent from

visitors to store or retrieve any information on their devices. The 'Privacy throughout the Data Cycle' approach, which looks at the different aspects of privacy protection, provides a suite of solutions that increases users privacy awareness, enables user empowerment and allows companies to achieve compliance with privacy regulations.

The GTPM focuses on understanding how to increase users privacy awareness by looking at what factors most affect users' trust towards websites. The study highlights the factors that are more important for the users and proves that by increasing users' knowledge, it is possible to increase privacy and security awareness. The GTPM and the user trust perception study are currently part of the foundation "Authentication and Authorisation for Entrusted Unions (AU2EU)[1]", a collaborative project between the EU and Australia that aims to foster the adoption of security and privacy-by-design technologies in European and global markets. To help bridge the gap between research solutions and practical deployment, the GTPM and methodology for evaluating user trust perception are being used in AU2EU to determine the trust impact of new (privacy enhancing) technologies such as anonymous authentication systems in key areas like eHealth and Ambient Assisted Living. In our study, we also propose a model that suggests the presence of two importance gap that can lead a user to experience feelings of regret: the Factor Importance Gap and the Trust Indicator Gap. While the Factor Importance Gap has been treated in this thesis, we left the analysis of the Trust Indicator Gap out of the scope of our analysis. Researchers could follow this lead and carry out a more in depth study of the Trust Indicator Gap, e.g. by quantifying the level of regret users experience when "things go bad" and whether the regret can be diminished by improving the trust indicators. Finally, the results of our user study show that trust perception is affected by users knowledge. Hence, training the users to increase their knowledge can contribute to create a more privacy-aware society, able to deal with the growing number of privacy and security threats to which it is exposed.

The work on the privacy policy evaluation provides a valid framework to help policy's authors to automatically test the completeness of their policies and accordingly editing them to increase their quality. In addition, developer could embed the solutions proposed in Chapter 3 and Chapter 4 into their websites to attract privacy-aware consumers. Integrating these solutions e.g. as indicators in web browsers, can also increase user awareness that, as argued above, is of primary importance. Training users to be more aware of potential privacy issues is necessary to implement the user empowerment, e.g. enabling users to select services according to their privacy preferences. In this respect, solutions as proposed in Chapter 5 would create added value.

Finally, the work presented in Chapter 6 boosts the practical usage of behavior-

[1] www.au2eu.eu

based anomaly detection techniques. The study, which has been empirically validated, shows the effectiveness of our approach in terms of detection rate with a very low cost in terms of false positive rate. Although in this thesis we adopt our white-box leakage detection in a database setting, it has a much wider range of applicability. It can be used in any setting where meaningful features that can help to distinguish undesired from desired activities are available. For example, ongoing work in network intrusion detection for critical infrastructure (being developed in the NWO SpySpot project[2]) show very promising initial results. Adapting, applying and validating effectiveness of the approach in different settings is thus a significant opportunity for further research. The potential practical impact of this work is even more appealing. The orders of magnitude improvement in the false positive rates (at similar detection rates) that the white-box method achieves is essential for the practical applicability of the system in a wide range of scenarios/environments and now, for the first time, this is possible or at least within reach.

7.3 Limitations and Directions for Future Work

In this final section we discuss possible future directions with respect to the privacy topics discussed in this thesis.

Extension to the GTPM In Chapter 2 we discuss GTPM and the results of our user study. Our results suggest that to reduce regret caused by bad trust decisions, it is necessary to reduce the Trust Indicator Gap and the Factor Importance Gap. To reduce the Trust Indicator Gap w.r.t. the privacy factor, in Chapter 3 and Chapter 4 we propose two new privacy indicators: the privacy completeness and the data collection quantification. Here, we see two possible future research directions. First, the execution of a user study to verify whether these metrics actually represent a better privacy indicator if compared to the sole presence of privacy policies. Second, the validation of the GTPM, by experimentally demonstrating that reducing the aforementioned gaps will also reduce future regret and make a real difference in the way users make trust decisions. In particular, it would be interesting to investigate more the situations in which users experience regret, and how external events influence a-posteriori trust and a-posteriori factor importance.

Website Evaluation Framework. In Chapter 3 and Chapter 4 we suggest two metrics to evaluate the privacy level of a website based on the contents of its privacy policy. The completeness grade can be seen as a tool giving the user a measure to

[2] `http://security1.win.tue.nl/spyspot/`

evaluate privacy policies. However, although the machine learning approach demonstrated good results for the completeness analysis, it has the disadvantage of disregarding the semantic information. In this sense, two policies that are completely different from a semantic perspective, but covering the same amount of categories, achieve – according to our approach – the same level of completeness. That means a user can only check whether a specific category is covered, but she needs to read the related paragraph(s) in the privacy policy to know how such category is addressed (e.g. which data is collected, with whom it is shared, etc.). To address this limitation we proposed the data collection metric presented in Chapter 4, which measures with precision which data items are collected.

We can see a number of challenges that still needs to be addresses w.r.t. website evaluation. First, in the privacy completeness metric presented in Chapter 3 we assume a single-label classification model, i.e. a model where a single category is assigned to each privacy paragraph. Since it is possible to have paragraphs that cannot be sorted into any of the privacy categories, and paragraphs that may be relevant to more than one category, a multi-label classification may be useful to improve the performances of the classifier. Moreover, some categories (e.g. collection, sharing, choice and access, and cookies) may be divided in sub-categories, and apply a two-layer classification model. In this way we may be possible to provide the user with a more granular information. Second, extending the semantic analysis to the other privacy categories, e.g., data sharing, data retention, etc., is a step towards providing a comprehensive analysis of a privacy policy. Third, it is also important to verify that users find our privacy indicators beneficial, for example by implementing a proof of concepts and by carrying out a guided user study to prove its ease of use and usefulness. The user study could be done by evaluating users interactions with a website when our tool (e.g., implemented as a plugin) is available and when it is not. Users behavior can be monitored to verify whether the tool is used at all (perceived usefulness) and how much time the user spend for its setting (required effort). After a first interaction with the tool, the user could be asked to fill-in a questionnaire to carry on a qualitative assessment aimed at understanding whether the presence of the tool increases privacy-awareness and what is the perceived ease of use. Finally, since the manual creation of extraction rules is a complex and error prone task, the use of systems for the automatic creation of extraction rules, e.g. by applying machine learning techniques, can also be considered. A first work in this direction can be found in [204, 205].

Automatic Authoring of Access Control Policies. In Chapter 5 we propose a model to let users express their privacy preferences and to match such preference with service providers' privacy policies. We think that an interesting future direction

would be the assessment of perceived ease of use, and the perceived usefulness, of the system we propose. This could be done by using the Technology Acceptance Model (TAM). Furthermore, consider that service providers often allow users to specify access control policies that reflect their privacy preferences. However, this task is in almost all cases too complex and too time consuming for the average users. To alleviate the users, we think it would be useful to study an automatic way to translate privacy preferences in enforceable access control policies (e.g., written in XACML).

Extension to the Database Leakage Detection. In Chapter 6 we presented an innovative solution for database leakage detection. In this area there are a number of challenges that we think are worth addressing. First, a general problem of behavioral-based anomaly detection is the high number of false positive. We devised the feedback loop mechanism to deal with false positives, however improving the way alarms are shown to the security office in order to ease her job is still possible. An interesting challenge is also the automatic classification of alarms: for instance, by associating a possible threat description to each alarm (or a set of alarms). For example, an alarm caused by a query string containing many unusual characters and unusual commands might be tagged as a SQL Injection attempt. Second, in our approach we define the *anomalyscore* to rank anomalies. This score is based on the probability of occurrence of events. However, the score does not take into account the sensitivity of the data involved in the anomaly nor the estimate risks that the leakage would have on an organization. Quantifying an alarm in terms of sensitivity and risk is an interesting direction for future work which would help organizations to focus on the potentially most severe incidents.

Bibliography

[1] "Charter of Fundamental Rights of the European Union," 2000.

[2] J. Gross and M. Rosson, "End user concern about security and privacy threats," in *Proceedings of the Third Symposium on Usable Privacy and Security*, ACM, 2007.

[3] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, "The platform for privacy preferences 1.0 (P3P1. 0) specification," *W3C*, 2002.

[4] P. Samarati and S. de Vimercati, "Access control: Policies, models, and mechanisms," *Foundations of Security Analysis and Design*, 2001.

[5] A. Shabtai, Y. Elovici, and L. Rokach, *A survey of data leakage detection and prevention solutions*. SpringerBriefs in Computer Science, Springer US, 2012.

[6] P. Gordon, "Data Leakage - Threats and Mitigation," tech. rep., SANS Institute, 2007.

[7] E. Costante, J. den Hartog, and M. Petković, "Understanding perceived trust to reduce regret," *Computational Intelligence*, 2014.

[8] E. Costante, J. D. Hartog, and M. Petković, "On-line Trust Perception : What Really Matters," in *Socio-Technical Aspects in Security and Trust (STAST), 2011 1st Workshop on*, pp. 52–59, IEEE, 2011.

[9] E. Costante, Y. Sun, J. den Hartog, and M. Petković, "A Machine Learning Solution to Assess Privacy Policy Completeness," in *10th annual ACM workshop on Privacy in the electronic society*, ACM, 2012.

[10] E. Costante, J. den Hartog, and M. Petković, "What Websites Know About You - Privacy Policy Analysis Using Information Extraction," in *7th DPM International Workshop on Data Privacy Management*, Springer-Verlag, 2012.

[11] E. Costante, F. Paci, and N. Zannone, "Privacy-aware web service composition and ranking," *Int. J. Web Service Res.*, vol. 10, no. 3, pp. 1–23, 2013.

[12] E. Costante, F. Paci, and N. Zannone, "Privacy-aware web service composition and ranking," in *2013 IEEE 20th International Conference on Web Services, Santa Clara, CA, USA, June 28 - July 3, 2013*, pp. 131–138, 2013.

[13] E. Costante, J. den Hartog, M. Petković, M. Pechenizkiy, and S. Etalle, "Hunting the Unknown White-Box Database Leakage Detection," in *DBSec'14 28th Annual IFIP WG 11.3 Working Conference*, Elsevier, 2014.

[14] S. Ruohomaa and L. Kutvonen, "Trust Management Survey," in *Procedings of the iTrust 3rd International Conference on Trust Management*, (Rocquencourt, France), pp. 77–92, 2005.

[15] J. Rotter, "Generalized expectancies for interpersonal trust.," *American Psychologist*, vol. 26, no. 5, 1971.

[16] D. Gambetta, "Can we trust trust," *Trust: Making and breaking cooperative relations*, pp. 213–237, 2000.

[17] S. Ganesan, "Determinants of Long-Term Orientation in Buyer-Seller Relationships," *Journal of Marketing*, vol. 58, no. 2, 1994.

[18] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *The Semantic Web - ISWC 2003*, vol. 2870 of *Lecture Notes in Computer Science*, pp. 351–368, Springer Berlin Heidelberg, 2003.

[19] S. P. Marsh, *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

[20] B. Friedman, P. H. Khan, and D. C. Howe, "Trust online," *Communications of the ACM*, vol. 43, no. 12, pp. 34–40, 2000.

[21] A. Beldad, M. de Jong, and M. Steehouder, "How shall I trust the faceless and the intangible? A literature review on the antecedents of online trust," *Computers in Human Behavior*, vol. 26, no. 5, pp. 857–869, 2010.

[22] A. Fruhling and S. Lee, "The influence of user interface usability on rural consumers' trust of e-health services," *International journal of electronic healthcare*, vol. 2, no. 4, pp. 305–321, 2006.

[23] S. Faja, "E-Health : An Exploratory Study of Trust Building Elements in Behavioral Health Web Sites," *Journal of Information Science and Technology*, vol. 3, no. 1, 2006.

[24] S. Yousafzai, J. Pallister, and G. Foxall, "A proposed model of e-trust for electronic banking," *Technovation*, vol. 23, no. 11, pp. 847–860, 2003.

[25] A. Kini and J. Choobineh, "An Empirical evaluation of the factors affecting trust in web banking systems," in *Proceedings of the Sixth Americas Conference on Information Systems*, pp. 185–191, 2000.

[26] A. Azam, P. F. Qiang, and M. I. Abdullah, "Consumers' E-commerce acceptance model: Antecedents of trust and satisfaction constructs," in *2012 IEEE Business, Engineering & Industrial Applications Colloquium (BEIAC)*, pp. 371–376, IEEE, 2012.

[27] L. Mui, M. Mohtashemi, and A. Halberstadt, "A computational model of trust and reputation," in *Annual Hawaii International Conference on System Sciences*, no. c, pp. 2431–2439, 2002.

[28] M. Deutsch, "Cooperation and trust: Some theoretical notes.," in *Nebraska Symposium on Motivation* (N. S. O. M. MR Jones, Editor, ed.), pp. pp. 275–320, Lincoln: Univer. Nebraska Press, 1962.

[29] J. B. Rotter, "Interpersonal trust, trustworthiness, and gullibility.," *American Psychologist*, vol. 35, no. 1, pp. 1–7, 1980.

[30] R. Mayer, J. Davis, and F. Schoorman, "An integrative model of organizational trust," *Academy of management review*, vol. 20, no. 3, pp. 709–734, 1995.

[31] C. Johnson-George and W. Swap, "Measurement of specific interpersonal trust: Construction and validation of a scale to assess trust in a specific other.," *Journal of Personality and Social Psychology*, vol. 43, no. 6, p. 1306, 1982.

[32] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th international conference on Intelligent user interfaces*, ACM, 2005.

[33] B. Muir, *Trust between humans and machines, and the design of decision aids*, vol. 27. Elsevier, 1987.

[34] M. Madsen and S. Gregor, "Measuring human-computer trust," in *11th Australasian Conference on Information Systems*, vol. 53, 2000.

[35] H. Atoyan, J. Duquet, and J. Robert, "Trust in new decision aid systems," in *Proceedings of the 18th International Conferenceof the Association Francophone d'Interaction Homme-Machine*, pp. 115–122, ACM, 2006.

[36] J. D. Lee and K. a. See, "Trust in automation: designing for appropriate reliance.," *Human factors*, vol. 46, no. 1, pp. 50–80, 2004.

[37] L. J. Hoffman, K. Lawson-Jenkins, and J. Blum, "Trust beyond security," *Communications of the ACM*, vol. 49, no. 7, pp. 94–101, 2006.

[38] L. J. Camp, "Designing for Trust," *Trust, Reputation, and Security: Theories and Practice*, vol. 35, no. 3, pp. 203–209, 2003.

[39] C. Corritore, B. Kracher, and S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model," *International Journal of Human-Computer Studies*, vol. 58, no. 6, pp. 737–758, 2003.

[40] D. H. Mcknight, C. J. Kacmar, and V. Choudhury, "Shifting Factors and the Ineffectiveness of Third Party Assurance Seals: A Two-Stage Model of Initial Trust in a Web Business," *Electronic Markets*, vol. 14, no. 3, pp. 252–266, 2004.

[41] A. Everard and D. Galletta, "Effect of Presentation Flaws on Users Perception of Quality of On-Line Stores Web Sites: Is it Perception that Really Counts?," *Second Annual Workshop on HCI Research*, p. 60, 2003.

[42] S. Flinn and J. Lumsden, "User perceptions of privacy and security on the web," in *The Third Annual Conference on Privacy, Security and Trust*, 2005.

[43] J.-C. Jiang, C.-A. Chen, and C.-C. Wang, "Knowledge and Trust in E-consumers' Online Shopping Behavior," in *2008 International Symposium on Electronic Commerce and Security*, pp. 652–656, IEEE, 2008.

[44] D. Hoffman, T. Novak, and M. Peralta, "Building consumer trust online," *Communications of the ACM*, vol. 42, no. 4, pp. 80–85, 1999.

[45] D. H. McKnight, V. Choudhury, and C. Kacmar, "Developing and Validating Trust Measures for e-Commerce: An Integrative Typology," *Information Systems Research*, vol. 13, no. 3, pp. 334–359, 2002.

[46] D. McKnight and N. Chervany, "What trust means in e-commerce customer relationships: an interdisciplinary conceptual typology," *International Journal of Electronic Commerce*, vol. 6, no. 2, pp. 35–59, 2001.

[47] D. McKnight, V. Choudhury, and C. Kacmar, "Trust in e-commerce vendors: a two-stage model," in *Proceedings of the twenty first international conference on Information systems*, no. Heider 1958, pp. 532–536, 2000.

[48] D. McKnight, L. Cummings, and N. Chervany, "Initial trust formation in new organizational relationships," *The Academy of Management Review*, vol. 23, no. 3, pp. 473–490, 1998.

[49] A. Kini and J. Choobineh, "Trust in electronic commerce: definition and theoretical considerations," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, no. C, pp. 51–61, IEEE, 1998.

[50] C. Hsu, "Dominant Factors for Online Trust," in *International Conference on Cyberworlds 2008*, pp. 165–172, IEEE, 2008.

[51] K. Lee, I. Kang, and D. McKnight, "Transfer from offline trust to key online perceptions: an empirical study," *Engineering Management, IEEE Transactions on*, vol. 54, no. 4, pp. 729–741, 2007.

[52] E. Sillence, P. Briggs, P. Harris, and L. Fishwick, "Developing trust practices for e-health," in *Trust in E-services: Technologies, Practices and Challenges*, pp. 235–258, IGI Global, 2007.

[53] B. Suh and I. Han, "The Impact of Customer Trust and Perception of Security Control on the Acceptance of Electronic Commerce," *International Journal of Electronic Commerce*, vol. 7, no. 3, pp. 135–161, 2003.

[54] C. Cheung and M. Lee, "An integrative model of consumer trust in internet shopping," in *European Conference on Information Systems (ECIS)*, (Naples, Italy), 2003.

[55] K. Jones, "Trust in consumer-to-consumer electronic commerce," *Information & Management*, vol. 45, no. 2, pp. 88–95, 2008.

[56] S. San-Martín and C. Camarero, "A Cross-National Study on Online Consumer Perceptions, Trust, and Loyalty," *Journal of Organizational Computing and Electronic Commerce*, vol. 22, no. 1, pp. 64–86, 2012.

[57] F. Li, D. Piekowski, A. van Moorsel, and C. Smith, "A Holistic Framework for Trust in Online Transactions," *International Journal of Management Reviews*, vol. 14, no. 1, pp. 85–103, 2012.

[58] J. Ermisch, D. Gambetta, H. Laurie, T. Siedler, and S. Noah Uhrig, "Measuring people's trust," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 172, no. 4, pp. 749–769, 2009.

[59] A. Josang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[60] J. a. Colquitt, B. a. Scott, and J. a. LePine, "Trust, trustworthiness, and trust propensity: a meta-analytic test of their unique relationships with risk taking and job performance.," *The Journal of applied psychology*, vol. 92, no. 4, 2007.

[61] L. Festinger, *A theory of cognitive dissonance*. Stanford University Press, 1957.

[62] D. F. Halpern, *Thought and knowledge: An introduction to critical thinking*. Lawrence Erlbaum Associates, Inc, 1989.

[63] S. Marsh and P. Briggs, "Examining Trust, Forgiveness and Regret as Computational Concepts," in *Computing with social trust* (J. Golbeck, ed.), Human-Computer Interaction Series, pp. 9–43, Springer London, 2009.

[64] S. Etalle, J. den Hartog, and S. P. Marsh, "Trust and Punishment," *Proceedings of the First International Conference on Autonomic Computing and Communication Systems*, 2007.

[65] R. Christie and F. Geis, *Studies in machiavellianism*. Academic Pr, 1970.

[66] M. Lynn, "Determination and quantification of content validity," *Nursing Research*, vol. 35, no. 6, p. 382, 1986.

[67] Statistics Netherlands, "The digital economy 2009," 2009.

[68] J. J. Meulman, "Optimal scaling methods for multivariate categorical data analysis ," spss - white paper, 1998. http://www.unt.edu/rss/class/Jon/SPSS_-SC/Module9/M9_CatReg/SWPOPT.pdf.

[69] J. de Leeuw and P. Mair, "Gifi methods for optimal scaling in r: The package homals," *Journal of Statistical Software*, vol. 31, no. 4, 2009.

[70] R. a. Mcdonald, P. W. Thurston, and M. R. Nelson, "A Monte Carlo Study of Missing Item Methods," *Organizational Research Methods*, vol. 3, no. 1, pp. 71–92, 2000.

[71] P. M. Podsakoff, S. B. MacKenzie, J.-Y. Lee, and N. P. Podsakoff, "Common method biases in behavioral research: a critical review of the literature and recommended remedies.," *Journal of Applied Psychology*, vol. 88, no. 5, pp. 879–903, 2003.

[72] A. Jø sang, L. Fritsch, and T. Mahler, "Privacy policy referencing," in *Trust, Privacy and Security in Digital Business*, pp. 129–140, 2010.

[73] E. Andrade, V. Kaltcheva, and B. Weitz, "Self-disclosure on the web: The impact of privacy policy, reward, and company reputation," *Advances in Consumer Research*, vol. 29, no. 1, pp. 350–353, 2002.

[74] L.-E. Holtz, K. Nocun, and M. Hansen, "Towards Displaying Privacy Information with Icons," *Privacy and Identity Management for Life*, pp. 338–348, 2011.

[75] A. Anton, J. Earp, Q. He, W. Stufflebeam, D. Bolchini, and C. Jensen, "Financial privacy policies and the need for standardization," *Security & Privacy, IEEE*, vol. 2, no. 2, pp. 36 – 45, 2004.

[76] C. a. Brodie, C.-M. Karat, and J. Karat, "An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench," *Proc. of SOUPS '06*, 2006.

[77] C. Brodie, C. Karat, J. Karat, and J. Feng, "Usable security and privacy: a case study of developing privacy management tools," in *Proc. of SOUPS '05*, pp. 35–43, 2005.

[78] W. Yu, S. Doddapaneni, and S. Murthy, "A Privacy Assessment Approach for Serviced Oriented Architecture Application," in *Proc. of SOSE'06*, pp. 67–75, 2006.

[79] W. D. Yu and S. Murthy, "PPMLP: A Special Modeling Language Processor for Privacy Policies," in *Proc. of ISCC 2007*, pp. 851–858, 2007.

[80] L. Shi and D. W. Chadwick, "A controlled natural language interface for authoring access control policies," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pp. 1524–1530, ACM, 2011.

[81] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise privacy authorization language (EPAL)," tech. rep., IBM Research, 2003.

[82] OASIS, "extensible access control markup language (xacml) version 2.0," tech. rep., OASIS, 2008.

[83] R. Schwitter, "English as a formal specification language," in *Proc. of DEXA '02*, pp. 228–232, 2002.

[84] J. Reagle and L. Cranor, "The platform for privacy preferences," *Communications of the ACM*, vol. 42, no. 2, pp. 48–55, 1999.

[85] L. F. Cranor, M. Arjula, and P. Guduru, "Use of a p3p user agent by early adopters," in *Proc. of WPES '02*, pp. 1–10, 2002.

[86] J. Tsai, S. Egelman, and L. Cranor, "The effect of online privacy information on purchasing behavior: An experimental study," *Information Systems*, vol. 21, 2011.

[87] P. Beatty, I. Reay, S. Dick, and J. Miller, "P3P Adoption on E-Commerce Web sites: A Survey and Analysis," *IEEE Internet Computing*, vol. 11, no. 2, pp. 65–71, 2007.

[88] W3C, "Privacy Enhancing Browser Extensions," tech. rep., W3C, 2011.

[89] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[90] S. Weiss, *Text mining: predictive methods for analyzing unstructured information*. Springer, 2005.

[91] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of Database Systems*, pp. 532–538, Springer, 2009.

[92] H. Farrell, "Constructing the International Foundations of E-Commerce - The EU-U.S. Safe Harbor Arrangement," *International Organization*, vol. 57, no. 02, pp. 277–306, 2003.

[93] J. Hiller, "The Regulatory Framework for Privacy and Security," *International Handbook of Internet Research*, pp. 251–265, 2010.

[94] S. Kotsiantis and D. Kanellopoulos, "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.

[95] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[96] T. Ayodele, "Types of Machine Learning Algorithms," in *New Advances in Machine Learning*, pp. 19–48, InTech, 2010.

[97] L. Kotthoff, I. Gent, and I. Miguel, "A Preliminary Evaluation of Machine Learning in Algorithm Selection for Search Problems," in *Proc. of SoCS-2011*, pp. 84–91, 2011.

[98] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pp. 3–24, 2007.

[99] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive bayes for text categorization revisited," in *Proc. of AI'04*, pp. 488–499, 2004.

[100] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," 1998.

[101] V. Vladimir, *The nature of statistical learning theory*. Springer, 1995.

[102] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine Learning: ECML-98*, pp. 137–142, 1998.

[103] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[104] J. Zhang and Y. Yang, "Robustness of regularized linear classification methods in text categorization," in *Proc. of SIGIR 2003*, pp. 190–197, 2003.

[105] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[106] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information retrieval*, vol. 1, no. 1, pp. 69–90, 1999.

[107] S. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 3, pp. 660–674, 1991.

[108] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inf. Syst.*, vol. 12, no. 3, 1994.

[109] L. Breiman, *Classification and regression trees*. Wadsworth International Group, 1984.

[110] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[111] G. Smits and E. Jordaan, "Improved SVM regression using mixtures of kernels," in *Proc. of IJCNN'02*, vol. 3, pp. 2785–2790, 2002.

[112] Z. Wang, Y. He, and M. Jiang, "A comparison among three neural networks for text classification," in *Proc. of ICSP 2006*, vol. 3, pp. 1–4, 2006.

[113] R. Polikar, "Ensemble based systems in decision making," *Circuits and Systems Magazine, IEEE*, 2006.

[114] G. Brown, "Ensemble learning," *Encyclopedia of Machine Learning*, pp. 1–24, 2010.

[115] D. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[116] G. Sigletos, G. Paliouras, and C. D. Spyropoulos, "Combining Information Extraction Systems Using Voting and Stacked Generalization," *Journal of Machine Learning Research*, vol. 6, pp. 175–1782, 2005.

[117] D. S. Wilks, "Statistical forecasting," in *International Geophysics*, ch. 7, pp. 215–300, Academic Press, 2011.

[118] C.-w. C. Hsu, C.-c. C. Chang, and C. C.-j. Lin, "A practical guide to support vector classification," *Bioinformatics*, vol. 1, no. 1, pp. 1–16, 2003.

[119] H. Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in *Proc. of ICTAI 1995*, pp. 388–391, 1995.

[120] P. Guarda and N. Zannone, "Towards the development of privacy-aware systems," *Information and Software Technology*, vol. 51, no. 2, pp. 337–350, 2009.

[121] S. Spiekermann, "Engineering privacy," *Software Engineering, IEEE*, vol. 35, no. 1, 2009.

[122] O. Tene, "Privacy in the Age of Big Data: A Time for Big Decisions," *Stanford Law Review Online*, 2012.

[123] E. Aïmeur, S. Gambs, and A. Ho, "UPP: User Privacy Policy for Social Networking Sites," in *Proc. of ICIW 2009*, IEEE, 2009.

[124] C. Jensen and C. Potts, "Privacy Policies As Decision-making Tools: An Evaluation of Online Privacy Notices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pp. 471–478, ACM, 2004.

[125] D. Bolchini, Q. He, A. Antn, and W. Stufflebeam, "i need it now: Improving website usability by contextualizing privacy policies," in *Web Engineering*, Lecture Notes in Computer Science, pp. 31–44, Springer Berlin Heidelberg, 2004.

[126] R. Agrawal, W. Grosky, and F. Fotouhi, "Ranking privacy policy," in *Proc. of ICDE 2007*, pp. 192–197, 2007.

[127] B. Miller, K. Buck, and J. Tygar, "Systematic analysis and evaluation of web privacy policies and implementations," in *Internet Technology And Secured Transactions, 2012 International Conference for*, pp. 534–540, 2012.

[128] J. W. Stamey and R. A. Rossi, "Automatically Identifying Relations in Privacy Policies," in *Proceedings of the 27th ACM International Conference on Design of Communication*, SIGDOC '09, pp. 233–238, ACM, 2009.

[129] C. Nédellec and A. Nazarenko, "Ontologies and Information Extraction," *CoRR*, vol. abs/cs/060, no. July, 2006.

[130] H. Cunningham, "Information extraction, automatic," in *Encyclopedia of Language and Linguistics* (K. Brown, ed.), vol. 5, Elsevier, 2005.

[131] J. Turmo and A. Ageno, "Adaptive information extraction," *ACM Computing Surveys (CSUR)*, vol. 38, no. 2, 2006.

[132] J. Hobbs, "The generic information extraction system," in *Proc. of MUC 1993*, 1993.

[133] K. Deemter and R. Kibble, "On coreferring: Coreference in MUC and related annotation schemes," *Computational linguistics*, 2000.

[134] L. Hirschman, P. Robinson, J. D. Burger, and M. B. Vilain, "Automating coreference: The role of annotated training data," *CoRR*, vol. cmp-lg/9803001, 1998.

[135] H. Cunningham, "GATE, a General Architecture for Text Engineering," *Computers and the Humanities*, vol. 36, no. 2, 2002.

[136] H. Cunningham, D. Maynard, and K. Bontcheva, *Text Processing with GATE (Version 6)*. GATE, 2011.

[137] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications," in *Proc. of ACL 2002*, 2002.

[138] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization," *UCLA Law Review*, vol. 57, 2010.

[139] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu, "SystemT: a system for declarative information extraction," *SIGMOD Rec.*, vol. 37, no. 4, 2009.

[140] N. Ashish, S. Mehrotra, and P. Pirzadeh, "Xar: An integrated framework for information extraction," in *WRI World Congress on Computer Science and Information Engineering*, 2009.

[141] H. Cunningham, D. Maynard, and V. Tablan, "Jape: a java annotation patterns engine," 1999.

[142] M. Alrifai, T. Risse, and W. Nejdl, "A hybrid approach for efficient web service composition with end-to-end qos constraints," *TWEB*, vol. 6, no. 2, pp. 7:1–7:31, 2012.

[143] K.-M. Chao, M. Younas, C.-C. Lo, and T.-H. Tan, "Fuzzy matchmaking for web services," in *Proc. of AINA*, pp. 721–726, IEEE, 2005.

[144] B. Jeong, H. Cho, and C. Lee, "On the functional quality of service (fqos) to discover and compose interoperable web services," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5411–5418, 2009.

[145] V. X. Tran and H. Tsuji, "QoS Based Ranking for Web Services: Fuzzy Approaches," in *Proc. of NWeSP*, pp. 77–82, 2008.

[146] P. Wang, K.-M. Chao, C.-C. Lo, C.-L. Huang, and Y. Li, "A Fuzzy Model for Selection of QoS-Aware Web Services," in *Proc. of ICEBE*, pp. 585–593, IEEE, 2006.

[147] C. B., M. Di Penta, and G. Canfora, "An empirical comparison of methods to support qos-aware service selection," in *Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, PESOS 2010, Cape Town, South Africa, May 1-2, 2010*, pp. 64–70, 2010.

[148] E. M. Maximilien and M. P. Singh, "Toward autonomic web services trust and selection," in *Proc. of SOC*, pp. 212–221, ACM, 2004.

[149] S. Paradesi, P. Doshi, and S. Swaika, "Integrating behavioral trust in web service compositions," in *Proc. of ICWS*, pp. 453–460, IEEE, 2009.

[150] P. Wang, K.-M. Chao, C.-C. Lo, R. Farmer, and P.-T. Kuo, "A reputation-based service selection scheme," in *Proc. of ICEBE*, pp. 501–506, IEEE, 2009.

[151] Z. Xu, P. Martin, W. Powley, and F. Zulkernine, "Reputation-Enhanced QoS-based Web Services Discovery," in *Proc. of ICWS*, pp. 249–256, IEEE, 2007.

[152] F. Massacci, J. Mylopoulos, and N. Zannone, "Hierarchical hippocratic databases with minimal disclosure for virtual organizations," *VLDB J.*, vol. 15, no. 4, pp. 370–387, 2006.

[153] A. Squicciarini, B. Carminati, and S. Karumanchi, "A privacy-preserving approach for web service selection and provisioning," in *Proc. of ICWS*, pp. 33–40, IEEE, 2011.

[154] S.-E. Tbahriti, M. Mrissa, B. Medjahed, C. Ghedira, M. Barhamgi, and J. Fayn, "Privacy-Aware DaaS Services Composition," in *Database and Expert Systems Applications*, LNCS 6860, pp. 202–216, Springer, 2011.

[155] R. Hewett and P. Kijsanayothin, "Privacy and recovery in composite web service transactions," *International Journal for Infonomics*, vol. 3, no. 2, pp. 240–248, 2010.

[156] W. Xu, V. N. Venkatakrishnan, R. Sekar, and I. V. Ramakrishnan, "A framework for building privacy-conscious composite web services," in *Proc. of ICWS*, pp. 655–662, IEEE, 2006.

[157] OASIS, "Web Services Business Process Execution Language Version 2.0," OASIS Standard, 2007.

[158] H. Foster, S. Uchitel, J. Magee, and J. Kramer, "Ws-engineer: A model-based approach to engineering web service compositions and choreography," in *Test and Analysis of Web Services*, pp. 87–119, Springer, 2007.

[159] R. Hamadi and B. Benatallah, "A Petri net-based model for web service composition," in *Proc. of ADC*, pp. 191–200, Australian Computer Society, Inc., 2003.

[160] X. Fu, T. Bultan, and J. Su, "Formal verification of e-services and workflows," in *Web Services, E-Business, and the Semantic Web*, LNCS 2512, pp. 188–202, Springer, 2002.

[161] D. Berardi, G. D. Giacomo, M. Lenzerini, M. Mecella, and D. Calvanese, "Synthesis of underspecified composite *e*-services based on automated reasoning," in *Proc. of SOC*, pp. 105–114, ACM, 2004.

[162] A. Mahfouz, L. Barroca, R. C. Laney, and B. Nuseibeh, "Requirements-driven collaborative choreography customization," in *Proc. of ICSOC*, LNCS 5900, pp. 144–158, Springer, 2009.

[163] M. P. Singh, A. K. Chopra, and N. Desai, "Commitment-based service-oriented architecture," *IEEE Computer*, vol. 42, no. 11, pp. 72–79, 2009.

[164] A. Tumer, A. Dogac, and I. H. Toroslu, "A semantic-based user privacy protection framework for web services," in *Proc. of ITWP*, LNCS 3169, pp. 289–305, Springer, 2005.

[165] A. Nyre, K. Bernsmed, S. Bo, and S. Pedersen, "A server-side approach to privacy policy matching," in *Proc. of ARES*, pp. 609 –614, 2011.

[166] L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle, "A P3P Preference Exchange Language 1.0 (APPEL1.0)." W3C Recommendation, 2002.

[167] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "XPref: a preference language for P3P," *Computer Networks*, vol. 48, no. 5, pp. 809–827, 2005.

[168] M. Banerjee, R. K. Adl, L. Wu, and K. Barker, "Quantifying privacy violations," in *Secure Data Management*, LNCS 6933, pp. 1–17, Springer, 2011.

[169] J. S. Hammond, R. L. Keeney, and H. Raiffa, *Smart choices : a practical guide to making better life decisions*. Broadway Books, 2002.

[170] L. Liu, N. Mehandjiev, and D.-L. Xu, "Multi-criteria service recommendation based on user criteria preferences," in *Proc. of RecSys*, pp. 77–84, ACM, 2011.

[171] F. Massacci, J. Mylopoulos, and N. Zannone, "Security Requirements Engineering: The SI* Modeling Language and the Secure Tropos Methodology," in *Advances in Intelligent Information Systems*, vol. 265 of *Studies in Computational Intelligence*, pp. 147–174, Springer, 2010.

[172] P. Guarda and N. Zannone, "Towards the Development of Privacy-Aware Systems," *Information and Software Technology*, vol. 51, no. 2, pp. 337–350, 2009.

[173] A. Harel, A. Shabtai, L. Rokach, and Y. Elovici, "M-score: A misuseability weight measure," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 3, pp. 414–428, 2012.

[174] S. Vavilis, M. Petković, and N. Zannone, "Data leakage quantification," in *Data and Applications Security and Privacy XXVIII*, vol. 8566 of *Lecture Notes in Computer Science*, pp. 98–113, Springer Berlin Heidelberg, 2014.

[175] T. Saaty, "How to make a decision: The Analytic Hierarchy Process," *EJOR*, vol. 48, pp. 9–26, 1990.

[176] L. Clement, A. Hately, C. von Riegen, and T. Rogers, "UDDI Spec Technical Committee Draft 3.0.2," oasis committee draft, 2004.

[177] "Simple Object Access Protocol (SOAP) 1.2," tech. rep., World Wide Web Consortium (W3C), 2007.

[178] "Web Services Policy 1.5 - Framework," tech. rep., World Wide Web Consortium (W3C, 2007.

[179] Information Age, "New EU data laws to include 24hr breach notification," 2013.

[180] Verizon, "The 2013 Data Breach Investigations Report," tech. rep., Verizon, 2013.

[181] D. Caputo, M. Maloof, and G. Stephens, "Detecting insider theft of trade secrets," *Security & Privacy, IEEE*, no. December, 2009.

[182] A. D. Brucker and H. Petritsch, "Extending access control models with break-glass," in *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies*, SACMAT '09, pp. 197–206, ACM, 2009.

[183] A. Patcha and J. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, 2007.

[184] J. Fonseca, M. Vieira, and H. Madeira, "Integrated intrusion detection in databases," in *Dependable Computing*, Springer Berlin Heidelberg, 2007.

[185] A. Kamra, E. Terzi, and E. Bertino, "Detecting anomalous access patterns in relational databases," *The VLDB Journal*, vol. 17, no. 5, 2007.

[186] A. Roichman and E. Gudes, "Diweda-detecting intrusions in web databases," in *Data and Applications Security XXII*, Springer Berlin Heidelberg, 2008.

[187] G. Wu, S. Osborn, and X. Jin, "Database intrusion detection using role profiling with role hierarchy," in *Secure Data Management*, Springer Berlin Heidelberg, 2009.

[188] C. Bockermann, M. Apel, and M. Meier, "Learning sql for database intrusion detection using context-sensitive modelling," in *6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer-Verlag, 2009.

[189] S. Mathew and M. Petropoulos, "A data-centric approach to insider attack detection in database systems," in *Recent Advances in Intrusion Detection*, Springer-Verlag, 2010.

[190] M. Gafny, A. Shabtai, L. Rokach, and Y. Elovici, "Poster: applying unsupervised context-based analysis for detecting unauthorized data disclosure," in *18th ACM Conference on Computer and Communications Security*, (Chicago, Illinois, USA), ACM, 2011.

[191] R. Santos, J. Bernardino, M. Vieira, and D. Rasteiro, "Securing Data Warehouses from Web-Based Intrusions," in *Web Information Systems Engineering*, Springer Berlin Heidelberg, 2012.

[192] C. Chung, M. Gertz, and K. Levitt, "Demids: A misuse detection system for database systems," in *Integrity and internal control information systems*, 2000.

[193] D. Bolzoni, S. Etalle, and P. Hartel, "Panacea: Automating attack classification for anomaly-based network intrusion detection systems," in *Recent Advances in Intrusion Detection*, SpringerLink, 2009.

[194] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, "N-Gram against the machine: on the feasibility of the n-gram network analysis for binary protocols," in *Research in Attacks, Intrusions, and Defenses*, SpringerLink, 2012.

[195] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, 2009.

[196] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, no. c, Springer Berlin Heidelberg, 2006.

[197] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *8th nternational Conference on Data Mining*, SIAM, 2008.

[198] O. Mazhelis, "One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection.," *South African Computer Journal*, 2006.

[199] X. Jin and S. Osborn, "Architecture for data collection in database intrusion detection systems," in *SDM*, Springer, 2007.

[200] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L2 theory," *Probability theory and related fields*, 1981.

[201] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, 2006.

[202] A. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition*, 1997.

[203] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *In Proceedings of the Twentieth International Conference on Machine Learning*, pp. 616–623, 2003.

[204] F. Xu, *Bootstrapping Relation Extraction from Semantic Seeds*. PhD thesis, Saarland University, 2008.

[205] S. Zimmeck and S. M. Bellovin, "Privee: An Architecture for Automatically Analyzing Web Privacy Policies," in *23rd USENIX Security Symposium (USENIX Security 14)*, pp. 1–16, USENIX Association, 2014.

Appendix A

Trust Perception Questionnaire

1. **What is your gender?**
   - ☐ Male
   - ☐ Female
2. **What is your age?**
   - ☐ 12-24
   - ☐ 25-44
   - ☐ 45-64
   - ☐ 65+
3. **What is the highest Educational Level Completed?**
   - ☐ Primary School
   - ☐ Secondary School
   - ☐ Bachelor Degree
   - ☐ Master Degree
   - ☐ Doctor of Philosophy (PhD)
   - ☐ Other
4. **Is your job/study field related to the ICT (Information and Communication Technology)?**
   - ☐ yes
   - ☐ no

5. **What is your job position?** ***Multiple choices are allowed***

   ☐ Student

   ☐ Unemployed

   ☐ Looking for a job

   ☐ Employee

   ☐ Employer

   ☐ Retired

   ☐ Other

6. **When it comes to use a computer I consider myself to have:**

   ☐ No knowledge *(not able to use it)*

   ☐ Limited Knowledge *(able to use it but not really confident with it)*

   ☐ Good Knowledge *(able to use it with high confidence in what I do)*

   ☐ Expert Knowledge *(able to use it with high confidence and understanding of technical aspects)*

7. **When it comes to using the Internet I consider myself to have:**

   ☐ No knowledge *(not able to use it)*

   ☐ Limited Knowledge *(able to use it but not really confident with it)*

   ☐ Good Knowledge *(able to use it with high confidence in what I do)*

   ☐ Expert Knowledge *(able to use it with high confidence and understanding of technical aspects)*

8. **Please, rate your knowledge about each of the following items:**

	No Knowledge	Limited Knowledge	Good Knowledge	Expert Knowledge
Encryption	()	()	()	()
HTTPS	()	()	()	()
PKI	()	()	()	()
Trust Policy	()	()	()	()
Privacy policy	()	()	()	()
Reputation	()	()	()	()
Digital Certificate	()	()	()	()
Service Provider	()	()	()	()

9. Please, read the following e-services descriptions and then answer the questions imagining you are in each of them (try to give an answer even if you have never used the services described).

e-banking	Nowadays banks usually offer to their customers the possibility to manage their bank account on-line. This kind of service is known as e-banking or internet banking. Suppose it is the first time you want access to such services allowing you to make payment, transfer money or check your balance.
e-commerce	The term e-commerce usually refers to the on-line shopping activities. You choose the product you need and pay for it through a web site. Suppose you want to buy a product (a shirt, a belt, a notebook) using an on-line shop you never experienced before.
e-health	There are many web sites providing patients with general medical information (e.g. about diabetes, flu, cancer...) and allowing them to chat with a doctor or with other patients suffering from the same disease. Suppose you are feeling sick and you want to find out more information about the symptoms; how serious they are and whether you need to see a doctor. Suppose you are looking for web sites offering such services and optionally giving you the possibility to consult a doctor on-line.
e-portfolio	The term e-portfolio refers to web sites for the employability, helping people looking for a job and developing their career. These web sites usually offer services like the uploading of CV, the settings of preferences for the job one would like to get and offers for suitable vacancies or specialization courses. Suppose you are looking for a job and you are intending to use a web site of this kind.

	e-banking	e-commerce	e-health	e-portfolio
Would you usually verify the presence of trustworthy third party logos on the web-site pages (i.e. seals like VerySign, eTrust, Ministry of Health)?	never ( ) almost never ( ) very often ( ) always ( )	( )	( )	( )
Would you verify that the connections of your transactions are secured (e.g. https in the url or the lock sign on the browser)?	( )	( )	( )	( )
Would you verify that the web service provider is a well-known brand you link with high competence in the sector (e.g. Amazon, Rabobank, or Ministry of Health)?	( )	( )	( )	( )
Would you analyse the risks associated to the usage of that website (e.g. money loss, personal information disclosure)?	( )	( )	( )	( )
Would you verify the website (or the service provider) has a good reputation (e.g. looking for its ranking among other users)?	( )	( )	( )	( )
Would you ask your friends if and what kind of experience they had with the website?	( )	( )	( )	( )
Would you read the privacy policy stated by the website?	( )	( )	( )	( )

	e-banking	e-commerce	e-health	e-portfolio
Would the way the website manages your personal data influence your trust in it?[1]	( )	( )	( )	( )
Would any crashes or error messages shown during your use of the website influence your trust in it?	( )	( )	( )	( )
Would the fact that the website is often not available (i.e. the web site is not reachable or not working when you need to access it) influence your trust in it?	( )	( )	( )	( )
Would the design aspects of the website (e.g. attractive colors, professional icons) influence your trust in it?	( )	( )	( )	( )
Would the presence of spelling or grammatical mistakes influence your trust in the website?	( )	( )	( )	( )
Would the ease of use of the website influence your trust in it?	( )	( )	( )	( )

10. Please, indicate now how frequently do you use each of the websites typology introduced before.

	never	almost never	very often	always
e-banking	( )	( )	( )	( )
e-commerce	( )	( )	( )	( )
e-health	( )	( )	( )	( )
e-portfolio	( )	( )	( )	( )

Summary

Modern society relies heavily on the availability of large quantities of personal information in digital form. Private data is stored not only at the user's premises, but at a whole range of public and private institutions as well, where it is often accessible remotely. We call *data cycle* the route typically followed by data from the moment it leaves the users premises, until it is stored in data repositories from where it can be accessed by the user and other actors as well. Along this route risks may arise at any time: at the very beginning (e.g., a user trust –and release data to– a site that proves to be fraudulent), along the way (e.g., privacy-invasive services are used to process an order) or at the end (e.g., sensitive information is leaked from repositories where data is stored). These risks expose individuals and society to new types of threats such as privacy breaches, identity theft and frauds. This thesis addresses the problem of data privacy protection by performing a comprehensive analysis of the privacy risks that may occur at different stages throughout the data cycle. Especially, we propose a suite of privacy solutions addressing the following challenges.

Understanding the user's perception of privacy risks and how users establish trust online. To this end we executed a user study. The study shows that awareness of privacy risks is a crucial element in determining which factors influence trust and that by increasing awareness one can drive trust decisions.

Evaluating websites with respect to the privacy protection they offer. Providing users with an objective value of the privacy quality of a service or a website is a way to guide their decisions. For this reason we propose a solution which automatically analyzes websites by applying machine learning and natural language processing techniques to their privacy policies.

Identifying the web service composition which best preserves privacy. Although websites are often seen as single entities, they usually group many web services together to reach a more complex scope. The way services are composed is usually transparent to users but it does affect their privacy. Therefore, we propose a solution for privacy-aware service composition which takes into account privacy concerns and users' preferences and identifies the web-service composition which best preserves privacy and best matches a user's preferences.

Detecting privacy infringements at data repositories where data is ultimately stored. Privacy breaches may happen, e.g. because of hackers gaining access to the data or malicious employees abusing their rights. To reduce these risks we propose a monitoring solution which analyzes transactions with the repositories and applies anomaly detection techniques to identify misuse and data leakage.

Author's Publications

Journal Publications.

1. Costante, E., den Hartog, J. and Petković, M. (2013). Understanding Perceived Trust to Reduce Regret. Computational Intelligence. Wiley-Blackwell, 1-21.

2. Costante, E., Paci, F. and Zannone, N. (2013). Privacy-aware web service composition and ranking. International Journal of Web Services Research. IGI Global, 1-23.

3. Costante, E., Hartog, J.I. den, Petković, M., Etalle, S. and Pechenizkiy, M. (2014). A Behavioral Based Approach to Database Leakage Detection. (*submitted for review*)

Book Chapters.

4. Costante, E., Hartog, J.I. den and Petković, M. (2012). Trust Management and User's Trust Perception in e-Business. Handbook of Research on e-Business Standards and Protocols : Documents, Data and Advanced Web Technologies. IGI Global, 321-341.

International Conferences.

5. Costante, E., Paci, F. and Zannone, N. (2013). Privacy-Aware Web Service Composition and Ranking. In Proc. 20th International Conference on Web Services. IEEE Computer Society, 131-138.

6. (Short Paper) Costante, E., Vavilis, S., Etalle, S., Petković, M. and Zannone, N. (2013). Database Anomalous Activities: Detection and Quantification. In proc. 10th International Conference on Security and Cryptography. SciTePress, 603-608.

7. Costante, E., Hartog, J.I. den, Petković, M., Etalle, S. and Pechenizkiy, M. (2014). Hunting the Unknown - White-Box Database Leakage Detection. In Proc. 28th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy, Volume 8566 of LNCS. Springer Berlin Heidelberg, 243-259.

International Workshops.

8. Costante, E., Hartog, J.I. den and Petković, M. (2011). On-line Trust Perception: What Really Matters. In Proc. 1st Workshop on Socio-Technical Aspects in Security and Trust. IEEE, 52-59.

9. Costante, E., Sun, Y., Petković, M. and Hartog, J.I. den (2012). A machine Learning Solution to Assess Privacy Policy Completeness. In Proc. ACM Workshop on Privacy in the Electronic Society. ACM, 91-96.

10. Costante, E., Hartog, J.I. den and Petković, M. (2013). What Websites Know About You: Privacy Policy Analysis Using Information Extraction. In Proc. Data Privacy Management and Autonomous Spontaneous Security, Volume 7731 of LNCS. Springer Berlin Heidelberg, 146-159.

Curriculum Vitae

Elisa Costante was born on 10-02-1984 in Avellino, Italy. In March 2010 she received a MSc Degree in Software Engineering from the University of Sannio in Benevento (Italy) with a thesis on trust and reputation for web services. In May 2010 she started a PhD project in the security group at the Eindhoven University of Technology. The results of her research are presented in this dissertation. Since May 2014 she is employed at SecurityMatters as Product Manager Enterprise Line.

IPA Dissertations

Titles in the IPA Dissertation Series since 2009

M.H.G. Verhoef. *Modeling and Validating Distributed Embedded Real-Time Control Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2009-01

M. de Mol. *Reasoning about Functional Programs: Sparkle, a proof assistant for Clean*. Faculty of Science, Mathematics and Computer Science, RU. 2009-02

M. Lormans. *Managing Requirements Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-03

M.P.W.J. van Osch. *Automated Model-based Testing of Hybrid Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-04

H. Sozer. *Architecting Fault-Tolerant Software Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-05

M.J. van Weerdenburg. *Efficient Rewriting Techniques*. Faculty of Mathematics and Computer Science, TU/e. 2009-06

H.H. Hansen. *Coalgebraic Modelling: Applications in Automata Theory and Modal Logic*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-07

A. Mesbah. *Analysis and Testing of Ajax-based Single-page Web Applications*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-08

A.L. Rodriguez Yakushev. *Towards Getting Generic Programming Ready for Prime Time*. Faculty of Science, UU. 2009-9

K.R. Olmos Joffré. *Strategies for Context Sensitive Program Transformation*. Faculty of Science, UU. 2009-10

J.A.G.M. van den Berg. *Reasoning about Java programs in PVS using JML*. Faculty of Science, Mathematics and Computer Science, RU. 2009-11

M.G. Khatib. *MEMS-Based Storage Devices. Integration in Energy-Constrained Mobile Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-12

S.G.M. Cornelissen. *Evaluating Dynamic Analysis Techniques for Program Comprehension*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-13

D. Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-14

H.L. Jonker. *Security Matters: Privacy in Voting and Fairness in Digital Exchange*. Faculty of Mathematics and Computer Science, TU/e. 2009-15

M.R. Czenko. *TuLiP - Reshaping Trust Management*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-16

T. Chen. *Clocks, Dice and Processes*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-17

C. Kaliszyk. *Correctness and Availability: Building Computer Algebra on top of Proof Assistants and making Proof Assistants available over the Web*. Faculty of Science, Mathematics and Computer Science, RU. 2009-18

R.S.S. O'Connor. *Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory*. Faculty of Science, Mathematics and Computer Science, RU. 2009-19

B. Ploeger. *Improved Verification Methods for Concurrent Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-20

T. Han. *Diagnosis, Synthesis and Analysis of Probabilistic Models*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-21

R. Li. *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to Medical Image Analysis*. Faculty of Mathematics and Natural Sciences, UL. 2009-22

J.H.P. Kwisthout. *The Computational Complexity of Probabilistic Networks*. Faculty of Science, UU. 2009-23

T.K. Cocx. *Algorithmic Tools for Data-Oriented Law Enforcement*. Faculty of Mathematics and Natural Sciences, UL. 2009-24

A.I. Baars. *Embedded Compilers*. Faculty of Science, UU. 2009-25

M.A.C. Dekker. *Flexible Access Control for Dynamic Collaborative Environments*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-26

J.F.J. Laros. *Metrics and Visualisation for Crime Analysis and Genomics*. Faculty of Mathematics and Natural Sciences, UL. 2009-27

C.J. Boogerd. *Focusing Automatic Code Inspections*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2010-01

M.R. Neuhäußer. *Model Checking Nondeterministic and Randomly Timed Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-02

J. Endrullis. *Termination and Productivity*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-03

T. Staijen. *Graph-Based Specification and Verification for Aspect-Oriented Languages*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-04

Y. Wang. *Epistemic Modelling and Protocol Dynamics*. Faculty of Science, UvA. 2010-05

J.K. Berendsen. *Abstraction, Prices and Probability in Model Checking Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2010-06

A. Nugroho. *The Effects of UML Modeling on the Quality of Software*. Faculty of Mathematics and Natural Sciences, UL. 2010-07

A. Silva. *Kleene Coalgebra*. Faculty of Science, Mathematics and Computer Science, RU. 2010-08

J.S. de Bruin. *Service-Oriented Discovery of Knowledge - Foundations, Implementations and Applications*. Faculty of Mathematics and Natural Sciences, UL. 2010-09

D. Costa. *Formal Models for Component Connectors*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-10

M.M. Jaghoori. *Time at Your Service: Schedulability Analysis of Real-Time and Distributed Services*. Faculty of Mathematics and Natural Sciences, UL. 2010-11

R. Bakhshi. *Gossiping Models: Formal Analysis of Epidemic Protocols*. Faculty of Sciences, Department of Computer Science, VUA. 2011-01

B.J. Arnoldus. *An Illumination of the Template Enigma: Software Code Generation with Templates*. Faculty of Mathematics and Computer Science, TU/e. 2011-02

E. Zambon. *Towards Optimal IT Availability Planning: Methods and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-03

L. Astefanoaei. *An Executable Theory of Multi-Agent Systems Refinement*. Faculty of Mathematics and Natural Sciences, UL. 2011-04

J. Proença. *Synchronous coordination of distributed components*. Faculty of Mathematics and Natural Sciences, UL. 2011-05

A. Moralı. *IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-06

M. van der Bijl. *On changing models in Model-Based Testing*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-07

C. Krause. *Reconfigurable Component Connectors*. Faculty of Mathematics and Natural Sciences, UL. 2011-08

M.E. Andrés. *Quantitative Analysis of Information Leakage in Probabilistic and Nondeterministic Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2011-09

M. Atif. *Formal Modeling and Verification of Distributed Failure Detectors*. Faculty of Mathematics and Computer Science, TU/e. 2011-10

P.J.A. van Tilburg. *From Computability to Executability – A process-theoretic view on automata theory*. Faculty of Mathematics and Computer Science, TU/e. 2011-11

Z. Protic. *Configuration management for models: Generic methods for model comparison and model co-evolution*. Faculty of Mathematics and Computer Science, TU/e. 2011-12

S. Georgievska. *Probability and Hiding in Concurrent Processes*. Faculty of Mathematics and Computer Science, TU/e. 2011-13

S. Malakuti. *Event Composition Model: Achieving Naturalness in Runtime Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-14

M. Raffelsieper. *Cell Libraries and Verification*. Faculty of Mathematics and Computer Science, TU/e. 2011-15

C.P. Tsirogiannis. *Analysis of Flow and Visibility on Triangulated Terrains*. Faculty of Mathematics and Computer Science, TU/e. 2011-16

Y.-J. Moon. *Stochastic Models for Quality of Service of Component Connectors*. Faculty of Mathematics and Natural Sciences, UL. 2011-17

R. Middelkoop. *Capturing and Exploiting Abstract Views of States in OO Verification*. Faculty of Mathematics and Computer Science, TU/e. 2011-18

M.F. van Amstel. *Assessing and Improving the Quality of Model Transformations*. Faculty of Mathematics and Computer Science, TU/e. 2011-19

A.N. Tamalet. *Towards Correct Programs in Practice*. Faculty of Science, Mathematics and Computer Science, RU. 2011-20

H.J.S. Basten. *Ambiguity Detection for Programming Language Grammars*. Faculty of Science, UvA. 2011-21

M. Izadi. *Model Checking of Component Connectors*. Faculty of Mathematics and Natural Sciences, UL. 2011-22

L.C.L. Kats. *Building Blocks for Language Workbenches*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2011-23

S. Kemper. *Modelling and Analysis of Real-Time Coordination Patterns*. Faculty of Mathematics and Natural Sciences, UL. 2011-24

J. Wang. *Spiking Neural P Systems*. Faculty of Mathematics and Natural Sciences, UL. 2011-25

A. Khosravi. *Optimal Geometric Data Structures*. Faculty of Mathematics and Computer Science, TU/e. 2012-01

A. Middelkoop. *Inference of Program Properties with Attribute Grammars, Revisited*. Faculty of Science, UU. 2012-02

Z. Hemel. *Methods and Techniques for the Design and Implementation of Domain-Specific Languages*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-03

T. Dimkov. *Alignment of Organizational Security Policies: Theory and Practice*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-04

S. Sedghi. *Towards Provably Secure Efficiently Searchable Encryption*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-05

F. Heidarian Dehkordi. *Studies on Verification of Wireless Sensor Networks and Abstraction Learning for System Inference*. Faculty of Science, Mathematics and Computer Science, RU. 2012-06

K. Verbeek. *Algorithms for Cartographic Visualization*. Faculty of Mathematics and Computer Science, TU/e. 2012-07

D.E. Nadales Agut. *A Compositional Interchange Format for Hybrid Systems: Design and Implementation*. Faculty of Mechanical Engineering, TU/e. 2012-08

H. Rahmani. *Analysis of Protein-Protein Interaction Networks by Means of Annotated Graph Mining Algorithms*. Faculty of Mathematics and Natural Sciences, UL. 2012-09

S.D. Vermolen. *Software Language Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-10

L.J.P. Engelen. *From Napkin Sketches to Reliable Software*. Faculty of Mathematics and Computer Science, TU/e. 2012-11

F.P.M. Stappers. *Bridging Formal Models – An Engineering Perspective*. Faculty of Mathematics and Computer Science, TU/e. 2012-12

W. Heijstek. *Software Architecture Design in Global and Model-Centric Software Development*. Faculty of Mathematics and Natural Sciences, UL. 2012-13

C. Kop. *Higher Order Termination*. Faculty of Sciences, Department of Computer Science, VUA. 2012-14

A. Osaiweran. *Formal Development of Control Software in the Medical Systems Domain*. Faculty of Mathematics and Computer Science, TU/e. 2012-15

W. Kuijper. *Compositional Synthesis of Safety Controllers*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-16

H. Beohar. *Refinement of Communication and States in Models of Embedded Systems*. Faculty of Mathematics and Computer Science, TU/e. 2013-01

G. Igna. *Performance Analysis of Real-Time Task Systems using Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2013-02

E. Zambon. *Abstract Graph Transformation – Theory and Practice*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-03

B. Lijnse. *TOP to the Rescue – Task-Oriented Programming for Incident Response Applications*. Faculty of Science, Mathematics and Computer Science, RU. 2013-04

G.T. de Koning Gans. *Outsmarting Smart Cards*. Faculty of Science, Mathematics and Computer Science, RU. 2013-05

M.S. Greiler. *Test Suite Comprehension for Modular and Dynamic Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-06

L.E. Mamane. *Interactive mathematical documents: creation and presentation*. Faculty of Science, Mathematics and Computer Science, RU. 2013-07

M.M.H.P. van den Heuvel. *Composition and synchronization of real-time components upon one processor*. Faculty of Mathematics and Computer Science, TU/e. 2013-08

J. Businge. *Co-evolution of the Eclipse Framework and its Third-party Plug-ins*. Faculty of Mathematics and Computer Science, TU/e. 2013-09

S. van der Burg. *A Reference Architecture for Distributed Software Deployment*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-10

J.J.A. Keiren. *Advanced Reduction Techniques for Model Checking*. Faculty of Mathematics and Computer Science, TU/e. 2013-11

D.H.P. Gerrits. *Pushing and Pulling: Computing push plans for disk-shaped robots, and dynamic labelings for moving points*. Faculty of Mathematics and Computer Science, TU/e. 2013-12

M. Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-13

M.J.M. Roeloffzen. *Kinetic Data Structures in the Black-Box Model*. Faculty of Mathematics and Computer Science, TU/e. 2013-14

L. Lensink. *Applying Formal Methods in Software Development*. Faculty

of Science, Mathematics and Computer Science, RU. 2013-15

C. Tankink. *Documentation and Formal Mathematics — Web Technology meets Proof Assistants*. Faculty of Science, Mathematics and Computer Science, RU. 2013-16

C. de Gouw. *Combining Monitoring with Run-time Assertion Checking*. Faculty of Mathematics and Natural Sciences, UL. 2013-17

J. van den Bos. *Gathering Evidence: Model-Driven Software Engineering in Automated Digital Forensics*. Faculty of Science, UvA. 2014-01

D. Hadziosmanovic. *The Process Matters: Cyber Security in Industrial Control Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-02

A.J.P. Jeckmans. *Cryptographically-Enhanced Privacy for Recommender Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-03

C.-P. Bezemer. *Performance Optimization of Multi-Tenant Software Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2014-04

T.M. Ngo. *Qualitative and Quantitative Information Flow Analysis for Multi-threaded Programs*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-05

A.W. Laarman. *Scalable Multi-Core Model Checking*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-06

J. Winter. *Coalgebraic Characterizations of Automata-Theoretic Classes*. Faculty of Science, Mathematics and Computer Science, RU. 2014-07

W. Meulemans. *Similarity Measures and Algorithms for Cartographic Schematization*. Faculty of Mathematics and Computer Science, TU/e. 2014-08

A.F.E. Belinfante. *JTorX: Exploring Model-Based Testing*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-09

A.P. van der Meer. *Domain Specific Languages and their Type Systems*. Faculty of Mathematics and Computer Science, TU/e. 2014-10

B.N. Vasilescu. *Social Aspects of Collaboration in Online Software Communities*. Faculty of Mathematics and Computer Science, TU/e. 2014-11

F.D. Aarts. *Tomte: Bridging the Gap between Active Learning and Real-World Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2014-12

N. Noroozi. *Improving Input-Output Conformance Testing Theories*. Faculty of Mathematics and Computer Science, TU/e. 2014-13

M. Helvensteijn. *Abstract Delta Modeling: Software Product Lines and Beyond*. Faculty of Mathematics and Natural Sciences, UL. 2014-14

P. Vullers. *Efficient Implementations of Attribute-based Credentials on Smart Cards*. Faculty of Science, Mathematics and Computer Science, RU. 2014-15

F.W. Takes. *Algorithms for Analyzing and Mining Real-World Graphs*. Faculty of Mathematics and Natural Sciences, UL. 2014-16

M.P. Schraagen. *Aspects of Record Linkage*. Faculty of Mathematics and Natural Sciences, UL. 2014-17

G. Alpár. *Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World*. Faculty of Science, Mathematics and Computer Science, RU. 2015-01

A.J. van der Ploeg. *Efficient Abstractions for Visualization and Interaction*. Faculty of Science, UvA. 2015-02

R.J.M. Theunissen. *Supervisory Control in Health Care Systems*. Faculty of Mechanical Engineering, TU/e. 2015-03

T.V. Bui. *A Software Architecture for Body Area Sensor Networks: Flexibility and Trustworthiness*. Faculty of Mathematics and Computer Science, TU/e. 2015-04

A. Guzzi. *Supporting Developers' Teamwork from within the IDE*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-05

T. Espinha. *Web Service Growing Pains: Understanding Services and Their Clients*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-06

S. Dietzel. *Resilient In-network Aggregation for Vehicular Networks*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2015-07

E. Costante. *Privacy throughout the Data Cycle*. Faculty of Mathematics and Computer Science, TU/e. 2015-08