Universitat Politècnica de València

Departament de Sistemes Informàtics i Computació

ENHANCING PRIVACY IN MULTI-AGENT SYSTEMS

Jose Miguel Such Aparicio

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the subject of Computer Science (Doctor en Informàtica)

Under the supervision of:
Dr. Ana García-Fornes
Dr. Agustín Espinosa

November 2011

PhD Thesis

Title:	Enhancing Privacy in Multi-agent Systems
Author:	Jose Miguel Such Aparicio
Supervisors:	Dr. Ana García Fornes Dr. Agustín Espinosa Minguet
Defense Date:	November 3rd, 2011

Examination Board:

President	Prof. Carles Sierra (IIIA-CSIC)
Secretary	Prof. Vicent J. Botti (Universitat Politècnica de València)
Member	Prof. Andrea Omicini (Università di Bologna)
Member	Prof. Sascha Ossowski (Universidad Rey Juan Carlos)
Member	Dr. Michael Rovatsos (University of Edinburgh)

A la Natalia i a la meva família.
Ells són allò que més feliç em fa.

Learn from yesterday,
live for today,
hope for tomorrow.

Albert Einstein

Acknowledgments

En primer lugar me gustaría agradecer la inestimable ayuda y apoyo de mis dos directores Ana y Agustín. Siempre han sabido estar ahí cuando los he necesitado, y los dos me han demostrado durante todo este tiempo tener una calidad humana excepcional. Para ellos sólo tengo palabras de agradecimiento, y sin ellos esta tesis no habría sido posible.

També m'agradaria agraïr a Vicent Botti el seu suport des que vaig entrar a formar part del GTI-IA. La seua ajuda i predisposició ha sigut sempre indispensable en temes tant d'investigació com personals.

Al Carles Sierra m'agradaria agraïr-li la seua ajuda en una part d'esta tesi. Les seues indicacions i consells han sigut de gran importància. A més, la meua estada al IIIA va resultar inoblidable, incloent, com no, el dinar a Mas Puig.

I would also like to thank Michael Rovatsos for his very useful comments on this thesis. Moreover, he actually was a more than excellent host during my stay at the University of Edinburgh.

Finalment, també m'agradaria agraïr a la gent amb la que he compartit el laboratori 205 el bon ambient que sempre hem tingut allí, les voltes que ens hem divertit quan hem viatjat per assistir a algún congrés, així com també el tracte excel·lent que hem tingut fora del treball. Menció especial als dinars al bar del conservatori amb alguns de vosaltres. A tots ells i per ordre alfabètic: Elena, Jaume, Joan, Joanmi, Maria, Toni, i Víctor, moltes gràcies. També m'agradaria extendre estos agraïments a tots els membres del GTI-IA, realment és un luxe poder treballar amb tots vosaltres.

Abstract

Privacy is of crucial importance in the era of global connectivity in which everything is inter-connected anytime and everywhere, with almost 2 billion world-wide users with connection to the Internet. Indeed, most of these users are concerned about their privacy. These concerns also apply for the new emerging research fields in computer science such as Multi-agent Systems. A Multi-agent System consists of a number of agents (which can be intelligent and/or autonomous) that interact with one-another. An agent usually encapsulates personal information describing its principal (names, preferences, tastes, credit card numbers, etc.). Moreover, agents carry out interactions on behalf of their principals. As a result, agents usually exchange personal information about their principals. This may have a direct impact on their principals' privacy.

In this thesis, we focus on avoiding undesired information collection and information processing in Multi-agent Systems. In order to avoid undesired information collection we propose a decision-making model for agents to decide whether or not to disclose personal information to other agents. We also contribute a secure Agent Platform that allow agents to communicate with each other in a confidential fashion, i.e., external third parties cannot collect the information that two agents exchange. In order to avoid undesired information processing, we propose an identity management model for agents in a Multi-agent System. This model allows agents to avoid undesired information processing by holding as many identities as needed for minimizing data identifiability, i.e., the degree by which personal information can be directly attributed to a particular principal. Finally, we describe how we implemented this model into an existing agent platform.

Resum

La pèrdua de privacitat està esdevenint un dels majors problemes al món de la informàtica. De fet, la majoria d'usuaris d'Internet (que actualment arriba a la quantitat de 2 bilions d'usuaris en tot el món) mostren una creixent preocupació per mantenir la seua privacitat. Estes preocupacions també afecten a les noves branques de la informàtica que estan emergint als últims anys. Específicament, en esta tesi ens centrem en la privacitat en Sistemes Multiagent. En estos sistemes, diversos agents (que, a més, poden ser intel·ligents i/o autònoms) interactuen entre ells per resoldre problemes. Estos agents solen encapsular informació personal dels usuaris als que representen (noms, preferències, targetes de crèdit, etc.). A més a més, estos agents solen intercanviar este tipus d'informació quan interactuen entre ells. Tot açò té el potencial d'acabar resultant en pèrdua de privacitat per part dels usuaris, i per tant, provocar en els usuaris una certa reticència a emprar estes tecnologies.

En esta tesi ens centrem en evitar la recol·lecció i el processat d'informació personal en Sistemes Multiagent. Per tal d'evitar la recol·lecció d'informació, proposem un model perquè un agent siga capaç de decidir quíns atributs (de tots els possibles que conformen la totalitat de la informació personal que té sobre l'usuari al que representa) revelar a altres agents. A més, proporcionem una infraestructura d'agents segura, perquè quan un agent decidisca revelar un atribut a un altre, només este últim agent siga capaç de tindre accés a l'atribut revelat, evitant que terceres parts puguen tindre accés a l'atribut en qüestió. Per tal d'evitar el processat d'informació personal, proposem un model de gestió de les identitats dels agents. Este model, permet als agents fer ús de diferents identitats per reduir el risc de processat de la informació. Finalment, en esta tesi també descrivim la implementació que hem fet d'este model en una plataforma d'agents.

Resumen

La pérdida de privacidad se está convirtiendo en uno de los mayores problemas en el mundo de la informática. De hecho, la mayoría de los usuarios de Internet (que hoy en día alcanzan la cantidad de 2 billones de usuarios en todo el mundo) están preocupados por su privacidad. Estas preocupaciones también se trasladan a las nuevas ramas de la informática que están emergiendo en los últimos años. En concreto, en esta tesis nos centramos en la privacidad en los Sistemas Multiagente. En estos sistemas, varios agentes (que pueden ser inteligentes y/o autónomos) interactúan para resolver problemas. Estos agentes suelen encapsular información personal de los usuarios a los que representan (nombres, preferencias, tarjetas de crédito, roles, etc.). Además, estos agentes suelen intercambiar dicha información cuando interactúan entre ellos. Todo esto puede resultar en pérdida de privacidad para los usuarios, y por tanto, provocar que los usuarios se muestren adversos a utilizar estas tecnologías.

En esta tesis nos centramos en evitar la recolección y el procesado de información personal en Sistemas Multiagente. Para evitar la recolección de información, proponemos un modelo para que un agente sea capaz de decidir qué atributos (de la información personal que tiene sobre el usuario al que representa) revelar a otros agentes. Además, proporcionamos una infraestructura de agentes segura, para que una vez que un agente decide revelar un atributo a otro, sólo este último sea capaz de tener acceso a ese atributo, evitando que terceras partes puedan acceder a dicho atributo. Para evitar el procesado de información personal proponemos un modelo de gestión de las identidades de los agentes. Este modelo permite a los agentes la utilización de diferentes identidades para reducir el riesgo del procesado de información. Además, también describimos en esta tesis la implementación de dicho modelo en una plataforma de agentes.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1. Introduction to Privacy

Privacy should not be seen as a problem associated only to new technologies (Yao et al., 2007). Indeed, privacy has been a concern long before the emergence of information technologies and the explosive growth of the Internet. There are studies that suggest that privacy is probably as old as the human race itself (Schermer, 2007). Even in primitive societies individuals have always had a desire for privacy (Westin, 1984). This desire for privacy is usually related to the tendency toward territoriality that most animals have. Moreover, the claim of a right for privacy is often related to the instinct of defending against intrusion.

The modern conception of privacy started more than a hundred years ago, with the seminal work of Warren and Brandeis (1890) *The right of privacy*. These two lawyers defined privacy as “the right to be let alone”. They were pioneers in considering the implications of technology in privacy. Specifically, they were very concerned about the implications of instantaneous photographs and portraits in injuring the feelings of the people in those photographs and portraits. Privacy was later recognized as a fundamental human right by the United Nations Declaration of Human Rights, the International Covenant on Civil and Political Rights, the Charter of Fundamental Rights of the European Union, and many other international treaties (Acquisti et al., 2008).

In the second part of the twentieth century, Alan Westin defined privacy as “the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated” (Westin, 1967). This is what is currently known as the informational self-determination right (Rannenberg et al., 2009). The concept of informational self-determination changed the right to privacy from the right to be let alone to its current incarnation as a means to limit the abuse of personal data (Schermer, 2007). Informational self-determination represents today’s European understanding and regulation of privacy in the context of information and communication technology (EU Directives 95/46/EC, 45/2001/EC, and 2002/58/EC).

Despite all these regulations, as the Internet has no governing or regulating body, privacy breaches are still possible. Nowadays, in the era of global connectivity (everything is inter-connected anytime and everywhere) with more than 2 billion world-wide users with connection to the Internet as of 2011[1], privacy is of great concern. In the real world, everyone decides (at least implicitly) what to tell other people about themselves. In the digital world, users have more or less lost effective control over their personal data. Users are therefore exposed to constant personal data collection and processing without even being aware of it (Fischer-Hübner and Hedbom, 2008). Garfinkel (2001) suggests that nowadays users have only one option to preserve their privacy: becoming hermits and not using online social networks, e-commerce sites, etc. Considering the increasing power and sophistication of computer applications that offer many advantages to individuals, becoming a hermit may not really be an option. However, all of these advantages come at a significant loss of privacy (Borking et al., 1999). Recent studies show that 90% of users are concerned or very concerned about privacy (Taylor, 2003). Moreover, almost 95% of web users admitted they have declined to provide personal information to web sites at one time or another when asked (Hoffman et al., 1999).

1.2. Motivation

Agent-based systems is one of the most important and exciting research areas that have arisen in the field of Information Technologies (IT) in the last decade (Luck et al., 2005). Nowadays, such systems successfully support many aspects of current applications and computing frameworks. In fact, the concept of *intelligent* agent has become ubiquitous in many IT disciplines, such as software engineering, computer networks, object-oriented programming, artificial intelligence, human-computer interaction, concurrent and distributed systems, mobile systems, telematic systems, computer-based cooperative systems, control systems, and e-commerce.

According to Wooldridge and Jennings (1995), an agent is defined by its flexibility, which implies that an agent is: reactive, an agent must answer to its environment;

[1] Internet Usage is for March 31, 2011. Please, refer to INTERNET USAGE STATISTICS `http://www.internetworldstats.com/stats.htm` to consult updated statistics on world Internet users and population.

proactive, an agent has to be able to try to fulfill his own plans or objectives; and social, an agent has to be able to communicate with other agents by means of some kind of language. A Multi-agent System (MAS) consists of a number of agents that interact with one-another (Wooldridge, 2002).

An intelligent agent usually encapsulates personal information describing its principal[2] (Fasli, 2007b), such as names, preferences, tastes, location (permanent address, geo-location at a given time), credit card number, transactions performed, roles in organizations and institutions, social characteristics (affiliation to groups, friends), and other personal information. Moreover, agents carry out interactions on behalf of their principals. As a result, agents usually exchange personal information about their principals. This may have a direct impact on their principals' privacy. For instance, agents act on behalf of their users in agent-mediated e-commerce (Sierra, 2004), as personal assistants (Mitchell et al., 1994), in virtual worlds like Second Life[3] (Weitnauer et al., 2008), as recommenders (Montaner et al., 2003), in agent-mediated knowledge management (van Elst et al., 2004), in agent-based semantic web services (Gibbins et al., 2004), in distributed problem solving (Wallace and Freuder, 2005), and many other current and future applications. Therefore, they play a crucial role to safeguard and preserve her user's privacy.

To our knowledge, privacy is seldom considered in the MAS research field. This leads to agent-based applications that invade individuals' privacy, causing concerns about their use. Therefore, studies that enhance privacy in Multi-agent Systems are needed. Moreover, Piolle et al. (2007) claim that a great number of researchers in the agent community acknowledge the importance of privacy and believe that more efforts should be made to improve privacy in Multi-agent Systems.

This thesis has been developed under the frame of three research projects on Multi-agent Systems. Privacy is a common and transversal topic in all of these projects. Thus, in this thesis, we focus on solving some of the privacy problems that are most related to these projects. Moreover, we incorporated some of the main

[2] In this thesis, we use the terms principal and user indistinctly to refer to the user that the agent is acting on behalf of. Principals are also called agent owners, or simply users in the related literature.

[3] `http://secondlife.com/`

results of this thesis directly in the models and infrastructures developed in these projects. Specifically, this thesis has been developed under the frame of the following projects funded by the Spanish Government:

- “Magentix: A Multiagent Platform Integrated into Linux” under grant TIN2005-03395 (Main Researcher: Ana Garcia-Fornes, from 2005 to 2008). Magentix is an Agent Platform (AP) that focusses on offering high levels of efficiency. For this reason, it was developed in the C language directly using the Linux OS services. As we will see in Chapter 2, privacy requires security for the control of information. Therefore, APs that aim at preserving privacy should firstly be secure. Moreover, as the main objective of Magentix is to be efficient, its security mechanisms need to be as efficient as possible.

- “Magentix2: A Multiagent Platform for Open Multiagent Systems” under grant TIN2008-04446 (Main Researcher: Ana Garcia-Fornes, from 2008 to 2011). Magentix2 is an AP that supports Open MAS in which previously unknown agents can engage in interactions with each other. To enhance privacy in Open MAS, an AP needs not only to be secure but also allow agents to be in control of what information about themselves is revealed to other agents. This includes that agents should be in control of their degree of identifiability. This usually requires the use of mechanisms such as pseudonymity. Therefore, APs need to incorporate these mechanisms and provide them to the agents running on top of them to facilitate the development of privacy-enhanced agents in Open MAS.

- “Agreement Technologies” CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 (Main Researcher: Carles Sierra, from 2007 to 2012). Agreement Technologies (AT) refer to computer systems in which autonomous software agents negotiate with one another, typically on behalf of humans, in order to come to mutually acceptable agreements. In these systems, agents need to decide whether or not they reveal personal information to other agents. Moreover, trust and reputation models play a crucial role for agents to choose their interaction partners in these systems. The possibility of having multiple identities, as usually required to enhance privacy, may even make worse the identity-related vulnerabilities that most of the current trust and reputation models have.

1.3. Objectives

The main objective of this thesis is to enhance privacy in Multi-agent Systems. We aim at enhancing privacy in Multi-agent Systems at two levels: agent models and agent infrastructures. To fulfill this general objective, we deal with the following sub-objectives:

- To study and identify the possible privacy breaches that can occur in Multi-agent Systems.
- To survey, classify, and review the existing literature on privacy and Multi-agent Systems, and to identify open challenges in this field.
- To propose and validate a security infrastructure that supports confidentiality in agent communication without requiring the identity of agents' principals.
- To propose and validate a privacy-enhancing agent identity management model that supports multiple identities and the selective disclosure of identity attributes.
- To integrate our proposed privacy-enhancing agent identity management model into an agent infrastructure.
- To propose and validate a disclosure decision-making model for agents to decide whether or not to disclose personal information to other agents.

1.4. Contributions

The specific contributions of this thesis are:

- **State of the Art**. We contribute an study that identifies the possible privacy breaches that can occur in Multi-agent Systems. This study also contains a review and a classification of previous works on privacy and Multi-agent Systems that provide satisfactory solutions for some specific privacy breaches. Moreover, we identify many open challenges in privacy and Multi-agent Systems based on this study. Some of these open challenges are solved in this thesis, but others are left as future work. Thus, this study can also serve as a roadmap on research on privacy and Multi-agent Systems.

- **Secure Agent Platform**. We contribute a secure Agent Platform that allow agents to communicate with each other in a confidential fashion, i.e., external third parties cannot access the information that two agents exchange. Moreover, this agent platform allows agents to communicate with each other without disclosing their principals' identities, which remain hidden. Although principals' identities are not known a priori, principals' identities can be obtained for accountability concerns (e.g. law enforcement). Including security features obviously makes an Agent Platform to perform worse because expensive cryptographic computations are required in order to assure integrity and confidentiality of the messages exchanged among agents. However, as shown in section 3.4, the performance degradation introduced by the Agent Platform is absolutely bearable. Another important feature of the secure Agent Platform is that it is almost transparent for agent developers. We show this by describing a simple application developed on top of the secure Agent Platform.

- **Privacy-enhancing Agent Identity Management**. We propose an identity management model for agents in a Multi-agent System. This model enhances privacy by allowing agents to hold as many identities as needed for minimizing data identifiability, i.e., the degree by which personal information can be directly attributed to a particular principal. We experimentally demonstrate that agents can reduce privacy loss by holding many identities. Moreover, privacy is enhanced without compromising accountability and other crucial aspects for agents in a Multi-agent System, such as trust and reputation. To this aim, the model proposes a solution for the well-known identity-related vulnerabilities of trust and reputation models. Otherwise, these vulnerabilities can be exploited through whitewashing and sibyl attacks.

- **Privacy-enhancing Agent Platform**. We propose an agent infrastructure that supports the development and execution of privacy-enhancing Multi-agent Systems. Our proposed agent infrastructure integrates an implementation of our privacy-enhancing agent identity management model into an existing agent platform. The resulting agent platform enhances privacy by providing mechanisms supporting our privacy-enhancing agent identity management model. This agent platform is also suitable to develop and execute Multi-agent Systems in which trust and reputation play a crucial role. For instance, agent-based e-commerce applications need to preserve principals' privacy to be of broad use. However, enhancing privacy may directly impact other crucial issues in agent-based e-commerce such as accountability, trust, and reputation that are also needed for principals to be willing to engage with and delegate tasks to agents. Finally, we perform experiments that demonstrate that the overhead of changing identities in this implementation has a temporal cost that is linear with the number of changes to be made. Therefore, agents developed in this agent infrastructure can minimize information processing about their principals' privacy loss without incurring in a not affordable temporal cost.

- **Self-disclosure Decision Making**. We propose a decision-making model for agents to decide whether or not to disclose personal information to other agents is acceptable or not. Current self-disclosure decision-making mechanisms consider the direct benefit and the privacy loss of disclosing an attribute. However, there are many situations in which the direct benefit of disclosing an attribute is a priori unknown. This is the case in human relationships, where the disclosure of personal data attributes plays a crucial role in their development. We propose a model based on psychological findings regarding how humans disclose personal information in the building of their relationships. This model considers intimacy on the one hand and privacy loss on the other hand. We experimentally show that agents using this decision-making model lose less privacy than agents that do not use this model while achieving the same intimacy to other agents. Moreover, in environments in which agents must interact with a percent of malicious agents less than or equal to 60%, agents using this decision-making model achieve even greater intimacy than agents that do not use it while losing less privacy. In environments in which agents must interact with a percent of malicious agents of almost 100%, agents that use this model lose much less privacy than agents that do not use it.

1.5. Structure of the Document

This document is organized as detailed bellow:

- Chapter 2 presents our first contribution in this thesis: the state of the art of studies in the field that falls into the intersection between privacy and Multi-agent Systems.
- Chapter 3 presents our second contribution in this thesis: a secure agent platform.
- Chapter 4 presents our third contribution in this thesis: a privacy-enhancing agent identity management model.
- Chapter 5 presents our fourth contribution in this thesis: a privacy-enhancing agent platform that implements our agent identity management model.
- Chapter 6 presents our fifth and last contribution in this thesis: a self-disclosure decision making model.
- Chapter 7 presents some concluding remarks, the author's related scientific publications, and future work.

CHAPTER 2

State of the Art

Contents

2.1. Introduction

Privacy can be threatened by three main information-related activities (Solove, 2006): information collection, processing, and dissemination. Information collection refers to the process of gathering and storing data about an individual. Information processing refers to the use or transformation of data that has been already collected. Information dissemination refers to the transfer of collected (and possibly processed) data to other third parties (or making it public knowledge).

Figure 2.1 depicts a visual scheme that details when information-related activities can be performed in the process of information exchanges among agents. Information collection occurs when agent A communicates personal information about its principal to agent B. In this case, agent B is the one that collects the information. Moreover, although not depicted in the figure for the sake of clarity, a malicious agent could overhear the communications between agent A and agent B and collect information about A. Once agent B has collected information about agent A, it can then process this information. Finally, agent B can disseminate the information it has about agent A (processed or not) to agent C.

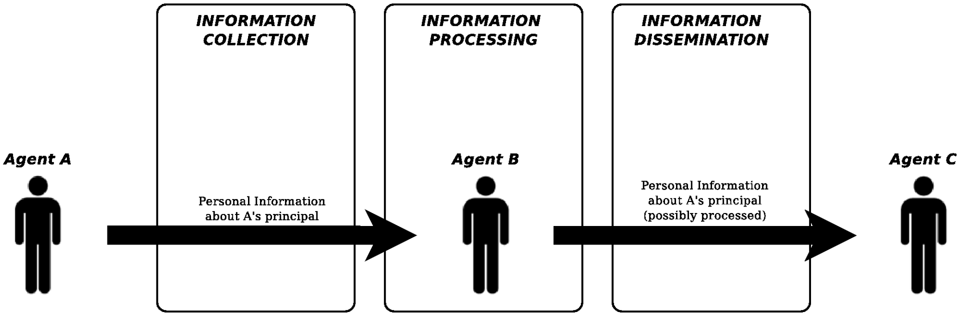
INFORMATION COLLECTION
INFORMATION PROCESSING
INFORMATION DISSEMINATION
Agent A
Agent B
Agent C
Personal Information about A's principal
Personal Information about A's principal (possibly processed)

Figure 2.1: Information-related Activities that can threaten Privacy

The information-related activities described above can represent a chance to breach the privacy of an agent's principal. Examples of possible privacy breaches that can emerge due to these activities are, but not limited to:

- **Secondary Use** refers to the use of collected information for purposes different from the purposes for which the information was initially collected and without the data subject's consent for doing so (Solove, 2006). There are potentially infinite types of secondary uses. In the following, we describe some of these possible *secondary uses*:

    - **Profiling**. Hildebrandt and Gutwirth (2008) define profiling as "*the process of 'discovering' patterns in data that can be used to identify or represent a human or nonhuman subject (individual or group) and/or the application of profiles (sets of correlated data) to individuate and represent an individual subject or to identify a subject as a member of a group (which can be an existing community or a discovered category) and/or the application of profiles to individuate and represent principals or groups*". According to this definition profiling can be achieved through information collection and processing. One of the most common types of profiling is called buyer profiling in e-commerce environments, in which vendors obtain detailed profiles of their customers and tailor their offers regarding customer's tastes.

    - **Price discrimination**. Vendors could charge customers different prices for the same good according to the customers' profiles (Odlyzko, 2003), i.e., if a vendor knows that some good is of great interest to one customer, the vendor could charge this customer more money for this good than other customers for the same good. For instance, in 2000, Amazon started to charge customers different prices for the same DVD titles (Spiekermann, 2006). When the story became public, Amazon claimed that this was part of a simple price test and discontinued this practice.

    - **Poor judgment**. This is when principals are judged and subsequently treated according to decisions made automatically based on incorrect or partial personal data (Smith and Milberg, 1996). For instance, companies usually divide their potential customers into similar groups based on customers' characteristics (known as customer segmentation). This practice can lead to exclusion of people from services based on potentially distorted judgments (Spiekermann and Cranor, 2009).

- **Identity theft** is “*fraud or another unlawful activity where the identity of an existing person is used as a target or tool without that person’s consent*” (Koops and Leenes, 2006). For instance, Bilge et al. (2009) present how to clone an existing account in an online social network and to establish a friendship connection with the victim in order to obtain information about her/him.

- **Spy Agents**. An agent could transfer information about its principal to other third parties without its principal’s consent and without its principal being aware of the transfer. For instance, an agent provider that hires or sells agents to principals may design and develop these agents so that they collect information on the principals and their activities (Bygrave, 2001).

- **Unauthorized Access**. Sensitive information about principals is transferred online even across the Internet and is stored in local and remote machines. Without appropriate protection mechanisms a potential attacker could easily obtain information about principals without their consent. For instance, an attacker can be listening to transferred information over the network (files, messages, e-mails, etc) and simply gather the information flowing in the network (Stallings, 2010). This is usually solved by encrypting the information exchanged over a network.

- **Traffic Analysis**. Although information exchanged over a network is encrypted, a potential attacker could also gather information about who is communicating with whom. This is because there is information such as the IP address and other whereabout information of both sender and receiver that is available even if the content of the transferred network packet is encrypted. Thus, this potential attacker could also know how often two individuals communicate to each other and even infer that two individuals are closely related to each other (Korba et al., 2002).

- **Unauthorized Dissemination or Exposure** refers to the transfer of previously collected and possibly processed information to other third parties, which are different from the one that collected (processed) the information, without the consent of the subject of this information. For instance, an agent A collects (and possibly processes) the information that it receives about another agent B.

> Agent A can transfer information about agent B to another agent C for whatever reason, e.g., to receive a monetary compensation. Thus, agent C can perform any of the aforementioned privacy breaches. Moreover, agent C could even make the information about agent A public knowledge, e.g., if agent C publishes the information about agent A, e.g., in an (online) journal/blog.

All of these privacy breaches may injure the feelings of the principal involved, as pointed out more than a hundred years ago by Warren and Brandeis (1890). Moreover, these privacy breaches can cause other consequences to the principal involved, such as money loss. These privacy breaches could even cause a principal to be summoned by a court, e.g., an attacker can steal the identity of a principal and impersonate her/him to carry out unlawful behaviors.

Agents play a crucial role in safeguarding and preserving their principals' privacy. They usually have a detailed profile of their principals' names, preferences, tastes, location (permanent address, geo-location at a given time), social characteristics (affiliation to groups, friends), roles in organizations and institutions, transactions performed, credit card numbers, and other personal information. To our knowledge, privacy is seldom considered in the Multi-agent Systems research field. This leads to agent-based applications that invade principals' privacy, causing concerns about their use and the privacy breaches explained above.

It is crucial for Multi-agent Systems to consider privacy in order to be of wide use. This can potentially promote principals' trust in agent-based technologies. This trust is needed for principals to be willing to engage with and delegate tasks to agents (Fasli, 2007a). To this aim, studies that enhance privacy in Multi-agent Systems technologies are needed. Moreover, agent designers and developers also need to be mindful of possible privacy implications when developing agent-based applications (Chopra and White, 2007). This means that agent designers and developers should choose to apply Multi-agent Systems technologies that preserve privacy, instead of Multi-agent Systems technologies that are unconcerned about privacy.

Despite having the potential to compromise their principals' privacy, Multi-agent Systems can also be used to preserve it (Solanas and Martínez-ballesté, 2009). Multi-agent Systems can offer themselves opportunities to enhance privacy beyond what other disciplines in information sciences can do due to their intrinsic features such as intelligence, pro-activeness, autonomy, and the like. According to (Westin, 1967), privacy can also be seen as a "personal adjustment process" in which individuals balance "the desire for privacy with the desire for disclosure and communication".

Humans have different general attitudes towards privacy that influence this adjustment process (Olson et al., 2005; Ackerman et al., 1999; Westin, 1967): privacy fundamentalists are extremely concerned about privacy and reluctant to disclose personal information; privacy pragmatists are concerned about privacy but less than fundamentalists and they are willing to disclose personal information when some benefit is expected; and finally, privacy unconcerned do not consider privacy loss when disclosing personal information. This view of privacy requires a dynamic management of privacy instead of a static one (Palen and Dourish, 2003). Multi-agent Systems can help to support this dynamism, as we will see during this survey.

2.2. Protection against Information Collection

As described above, information collection can play a key role in breaching privacy, i.e., collected data about a principal can be used to breach her/his privacy. In this section, we describe works in the agent research field that prevent undesired collection of sensitive information. According to Spiekermann and Cranor (2009), information collection involves data transfer and data storage. For the case of agents, this means that information collection involves one agent sending sensitive information to another agent, and that both agents are able to store this sensitive information. Therefore, it is crucial for agents to first decide which information to transfer to which other agent by means of a decision making mechanism (as described in Section 2.2.1), and then transfer and store it securely using traditional security mechanisms, such as those that provide confidentiality (as described in Section 2.2.2).

Another approach for avoiding undesired information collection is the use of third parties. In this case, agents does not send the information directly to the intended destination agents, instead agents provide sensitive information to third parties. These third parties process the information and return the obtained outcomes to the intended destination agents. We describe studies that follow this approach in Section 2.2.3.

Figure 2.2 depicts a conceptual map for all of the studied approaches that provide support for protecting against information collection.

2.2.1. Disclosure decision making

A first important approach to prevent information collection is to decide exactly which information to disclose to other agents. Agents should be able to decide which information to disclose according to their principals' preferences about privacy. This is crucial to prevent undesired information collection. Thus, agents need to incorporate disclosure decision-making mechanisms allowing them to decide whether disclosing personal information to other agents is acceptable or not.

Based on policies

One approach for disclosure decision making is based on policies. In this approach, agents usually specify their policies for both disclosing information and the information they want to collect from others. Then, if an agent's policy for disclosing information matches another agent's policy for collecting information from others, the former agent sends the information to the latter.

Tentori et al. (2006) presents a privacy-aware agent-based framework that allows agent developers to indicate two privacy-related policies (following an XML schema) per agent: one specifying the privacy policy for information that the agent communicates to others and the other specifying the privacy policy for the information that it requires. There is an agent broker that checks that both policies are compatible. Then, the agent broker monitors and ensures that the information that the two agents exchange complies with the policies. Although it allows the real compliance of privacy policies to be checked, the agent broker becomes a clear performance bottleneck and a single point of failure. Moreover, the agent broker itself can be a source of privacy

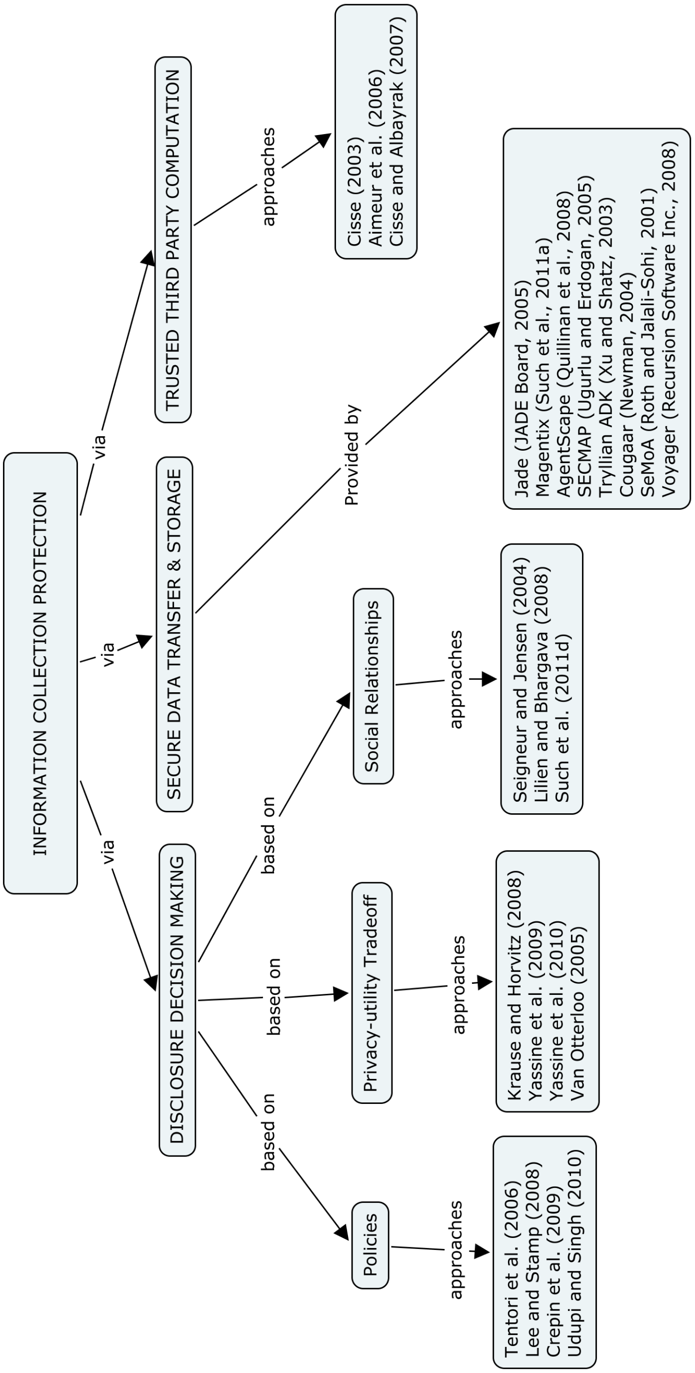
INFORMATION COLLECTION PROTECTION
via
via
via
TRUSTED THIRD PARTY COMPUTATION
SECURE DATA TRANSFER & STORAGE
DISCLOSURE DECISION MAKING
approaches
Cisse (2003)
Aimeur et al. (2006)
Cisse and Albayrak (2007)
Provided by
Jade (JADE Board, 2005)
Magentix (Such et al., 2011a)
AgentScape (Quillinan et al., 2008)
SECMAP (Ugurlu and Erdogan, 2005)
Tryllian ADK (Xu and Shatz, 2003)
Cougaar (Newman, 2004)
SeMoA (Roth and Jalali-Sohi, 2001)
Voyager (Recursion Software Inc., 2008)
based on
based on
based on
Social Relationships
Privacy-utility Tradeoff
Policies
approaches
Seigneur and Jensen (2004)
Lilien and Bhargava (2008)
Such et al. (2011d)
approaches
Krause and Horvitz (2008)
Yassine et al. (2009)
Yassine et al. (2010)
Van Otterloo (2005)
approaches
Tentori et al. (2006)
Lee and Stamp (2008)
Crepin et al. (2009)
Udupi and Singh (2010)

Figure 2.2: Information Collection Conceptual Map

concerns because it knows all the information that two agents communicate to each other.

Lee and Stamp (2008) present an approach based on P3P[1].The privacy-enhancing agent (PEA) is in charge of automatically retrieving P3P policies of service providers and evaluates whether or not these policies are compliant with its principal's policy. When a principal attempts to access a website, PEA automatically retrieves the website P3P policy and compares it to its principal's preferences. If PEA detects potential privacy violations (i.e., the principal's preferences and the website's P3P policy do not match) or is unable to read the policy of the website, it notifies its principal so that the principal can decide to desist in accessing the website. This approach does not consider that a website may not comply with its announced policy, and, thus, principals' privacy breaches are still possible.

Crépin et al. (2009) present an ontology described using OWL[2]. Agents can define their policies using this ontology in terms of the Hippocratic MAS (Crépin et al., 2008) concepts. They differentiate between data provider and data consumer agents. Both of them define their privacy policies according to this ontology. They also propose a protocol by which data consumers request sensitive data from data providers. Data consumers include their policies in the request. If the policy matches the data provider's preferences, the data provider sends the consumer the requested sensitive data. If not, the data provider proposes some modifications to the policy in order to reach an agreement. If the data consumer accepts these adaptations, the data provider sends the requested sensitive data to the data consumer, otherwise the consumer cancels the transaction. Again, they do not consider that data consumers may not comply with the policy they committed to.

Udupi and Singh (2010) present InterPol, a policy language and a framework for interaction in multi-agent referral networks. Policies are logic rules that can be implemented in Prolog. Policies can dynamically adapt to changes in the relationships with other agents. Policies are dynamic in the sense that new predicates can be added to the agent's Knowledge Base. InterPol provides two privacy mechanisms: (i) mark-

[1] The Platform for Privacy Preferences 1.0 http://www.w3.org/TR/P3P/

[2] OWL Web Ontology Language http://www.w3.org/tr/owl-features/

ing a rule of fact with its visibility (public or private); and (ii) using two privacy-related predicates, servicePrivacyNeed and agentPrivacyTrust, that dynamically manage the privacy decision making. This decision making is based on the privacy that an agent needs for a service and the trust it has in other agents when dealing with privacy issues. However, they do not provide any mechanism for deciding how and when both servicePrivacyNeed and agentPrivacyTrust should be updated.

All of the works presented in this section check that the data exchanged complies with the policies. However, none of them checks that once the data is collected it is treated as stated in the policies. Thus, an agent can disclose personal information to another agent expecting that this second agent will comply with its policy. However, this second agent may not comply with the policy, incurring in possible breaches of privacy.

Based on privacy-utility tradeoffs

There are a great number of people that are willing to trade part of their privacy in exchange for some benefit (64% of US citizens according to Taylor (2003)). They are known as privacy pragmatists, as mentioned above. There are many studies that have focused on providing models of the so-called privacy-utility tradeoff (Krause and Horvitz, 2008; Lebanon et al., 2006). The decision in this case is whether or not a particular privacy-utility tradeoff is acceptable for disclosing information, and then allowing the destination party to collect this information.

The privacy-utility tradeoff is usually modeled as follows. Given a set of personal data attributes A, the utility function of disclosing these attributes $U(A)$, and the privacy cost function of disclosing these attributes $C(A)$, the privacy-utility tradeoff is modeled as $A^* = \arg\max_A U(A) - C(A)$. An example of utility function is the one used by Krause and Horvitz (2008) that measures the reduction of time for performing an online search if some personal data attributes such as the geographical location are given. The privacy cost is usually defined taking into account the sensitivity of the information to be disclosed.

Yassine and Shirmohammadi (2009) present an agent-based architecture that negotiates a reward to be paid to agents' principals by the service providers in return for their disclosures. The data categorization agent is in charge of classifying principals' information into different categories (Yassine et al., 2010). The data categorization agent is able to calculate the privacy cost of the information about its principal considering the categories that this information falls into and the sensitivity for these categories. Principals define the sensitivity for each category and each service provider. The privacy cost is used to calculate an expected reward for disclosing the information. The payoff negotiator agent negotiates a reward for the information disclosed with the service provider, discarding any deal that provides its principal less than her/his expected reward. In this approach, principals must define the expected privacy cost for each category and for each service provider. This can be a burden for principals when considering a large number of service providers.

Another different approach is the one presented by van Otterloo (2005). The author does not focus on information directly disclosed to another party but on the information that can be collected by observing the strategies an agent follows in a game. The author defines minimal information games as games in which the agent tries to maximize its utility while minimizing the privacy loss. Privacy loss is calculated as the uncertainty (Shannon (1948) entropy) of the strategy that the agent will use. Thus, if an agent uses strategies with high uncertainty, other agents cannot predict their behavior. The author also defines most normal games as games in which the agent tries to maximize its utility while deviating the minimum from the *normal* strategy that other agents will play. In this sense, the agent tries to hide the preferences that differ from the normal behavior of the rest of the agents. The deviation from the normal strategy is calculated as the relative entropy between the agent strategy and the normal strategy. The author of this work does not consider that different actions may have different privacy sensitivity.

The research based on the privacy-utility tradeoff does not consider that there are also other reasons that make people decide whether or not to disclose information about them rather than an increase in utility or a decrease in privacy loss. There are many cases where the direct benefit of disclosing information is not known in advance. The decisions on whether or not to disclose information are based on other reasons in these situations. For instance, the well-known psychological phenomenon called *disclosure reciprocity* (Green et al., 2006) states that one person's disclosure encourages the disclosure of the other person in the interaction, which in turn, encourages more disclosures from the first person.

2.2.2. Secure Data Transfer and Storage

Once an agent has decided what information to disclose to which agent, this information must be protected from access by any other third party that is different from the agent that the information is intended for. This includes parties from their local computer and network but also different locations, even across the Internet. As stated by Petkovic and Jonker (2007), privacy protection is only possible using secure systems. Security and privacy are often related to each other, but they are not the same (Head and Yuan, 2001). On the one hand, security is the control of information in general (Camp, 1996). Thus, information is secure if the *owner* of the information can control it. On the other hand, information is private if the *subject* of information can control it. Thus, privacy requires security to control access and distribution of information (Garfinkel, 2009).

Agent platforms (APs) provide all the basic infrastructure (for message handling, tracing and monitoring, run-time management, and so on) required to create a MAS (Wooldridge, 2002). There are many APs developed by the agent community – for an overview of current APs and the features they provide refer to Alberola et al. (2010). As APs are in charge of executing MAS, they need to be concerned about basic security concepts. However, only a few of them currently take security concerns into account. For instance, Jade (JADE Board, 2005), Magentix (Such et al., 2011) (detailed in Chapter 3), AgentScape (Quillinan et al., 2008), SECMAP (Ugurlu and Erdogan, 2005), Tryllian ADK (Xu and Shatz, 2003), Cougaar (Newman, 2004), SeMoA (Roth and Jalali-Sohi, 2001), the one presented by Ismail (Ismail, 2008), and Voyager (Recursion Software Inc., 2008) are security-concerned APs.

There are security concepts that are necessary for preserving privacy, such as confidentiality (Gangopadhyay, 2001). All of the above APs use different mechanisms for providing confidentiality. Confidentiality is a security property of a system that ensures the prevention of unauthorized reading of information (Stamp, 2006). This involves both the control of access to information and its distribution. Confidentiality requires authorization mechanisms being in place as well as mechanisms for protecting the transmission of data over a network. They are key mechanisms for avoiding the leak of sensitive information, and, thus, protecting the subject of this information from a privacy breach.

Authorization is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity (Longstaff et al., 1997). The term access control is often used as a synonym for authorization. As an example of a scenario where authorization is necessary for preserving privacy, imagine that two agents A and B are running on the same host and represent two different principals. Agent A may contain a detailed profile about its principal and save this profile in a local file. In this situation, agent B must only be able to access this local file if it is authorized to do so. If agent B succeeds in accessing the file despite not being authorized to do so, this could represent a privacy breach for the principal of agent A.

Security-concerned APs use different authorization mechanisms. These mechanisms allow the specification of rights for agents to carry out activities ranging from traditional access control lists (Jade, Voyager) to other approaches, such as capability-based access control (Ismail, 2008), role-based access control (SeMoA, AgentScape, and Tryllian ADK), policy-based access control (Cougaar and SECMAP), and mandatory access control (Magentix). Most APs enforce the access rights that are defined using some of these approaches by means of sandboxing agents.

Confidentiality also implies the protection of transmitted data across a network (Stallings, 2010). In these situations, confidentiality usually means that sensitive information is encrypted into a piece of data so that only parties that are able to decrypt that piece of encrypted data can access the sensitive information. Current security-concerned APs provide confidentiality for the messages exchanged by the agents running on top of them. To this aim, APs use existing secure data transfer technologies such as Kerberos (Neuman et al., 2005), SSL (Frier et al., 1996), and TLS (Dierks and Allen, 1999). These technologies allow the encryption of messages before transferring them and the decryption of messages once they are received. As a result, if an agent A sends a message to an agent B, A is sure that B will be the only one able to read this message.

Without appropriate confidentiality mechanisms privacy cannot be preserved. All of the APs above provide the needed secure features to secure data transfer and storage. Therefore, they are suitable to prevent undesired collection of information. However, there are also security concepts that can represent by themselves an actual threat for privacy (Petkovic and Jonker, 2007), even though they are mandatory for the system to be secure. For instance, to achieve access control, each entity trying to gain access must first be authenticated so that access rights can be tailored to it (Stallings, 2010). As we will see in section 2.3.3, authentication can itself be a threat for privacy. Only some of the security-concerned APs provide mechanisms for preserving-privacy authentication. This should be considered when choosing an AP if there are privacy concerns to be considered.

2.2.3. Trusted Third Party Computation

Another approach to prevent information collection is based on third parties. Agents provide sensitive information to third parties that process this information and return the outcomes obtained to the intended destination agents. The agent-based information filtering community has developed some proposals that are based on trusted third parties (Cissée, 2003; Aïmeur et al., 2006; Cissée and Albayrak, 2007). Information filtering architectures take user profiles and generate personalized information based on them. User profiles usually contain information about preferences, rated items, etc. The resulting systems can be recommender systems, matchmaker systems, or can be a combination of both. The proposals we describe in this section enhance privacy by decoupling the three main parts in an information filtering architecture: users, service providers, and filters.

Aïmeur et al. (2006) present a software architecture that they call ALAMBIC. ALAMBIC considers three main parties: users, service providers, and the Still Maker. The Still Maker is a secure platform that generates mobile agents (with a unique public/private key pair) that migrate to service providers. These agents are in charge of filtering the information about users. The code of ALAMBIC agents is encrypted and obfuscated. Moreover, users' profiles are encrypted with the public key of the mobile agents before being transferred to the service provider. As a result, it is very difficult for service providers to obtain more information than the outputs of the filtering process that is carried out inside the mobile agents. However, according to Cissée and Albayrak (2007), this architecture addresses two aspects inadequately: the protection of the filter against manipulation attempts, and the prevention of collusion between the filter and the provider.

To overcome these aspects, Cissée and Albayrak (2007) propose separating the filter from the service provider. This proposal is based on the use of a trusted agent platform. Users, service providers, and filter entities (the party that provides filtering functionalities) can deploy agents in the trusted agent platform. In a nutshell, the information filtering process involves the following steps: (i) the filter entity deploys a temporary filter agent in the trusted agent platform; (ii) the user entity also deploys an agent, which is called relay agent, in the trusted agent platform; (iii) the relay agent establishes control of the temporary filter agent (by using mechanisms provided by the trusted agent platform) and sends the user profile to the temporary filter agent; (iv) the provider profile is propagated from the service provider to the temporary filter agent via the relay agent; (v) the temporary returns the recommendations to the service provider via the relay agent. The authors of this work assume that all providers of agent platforms are trusted. This assumption may be not valid in truly open Multi-agent systems in which there could be untrusted agent platforms.

2.3. Protection against Information Processing

Information processing refers to the use or transformation of data that has been already collected (Spiekermann and Cranor, 2009). Information processing usually involves various ways of connecting data together and linking it to the individuals to whom it pertains (Solove, 2006). For instance, a vendor could have a complete profile of a customer containing relevant data collected from the purchases made by the customer's agent. The vendor can then use information filtering techniques to obtain detailed information on the customer's tastes. Then, the vendor can infer which goods the customer is more willing to acquire and offer them in advance through personalized advertising. Moreover, the vendor could even incur in price discrimination practices, i.e., the vendor could charge different prices to different customers depending on the desire that the customer has to acquire a product according to their tastes.

Most of the work for protecting against the processing of information already collected is based on the principle of data minimization. Data minimization states that disclosed personal data should preserve as much unlinkability as possible (Pfitzmann and Hansen, 2010). This is a way to reduce the probability of different pieces of data being connected to each other and linked to an individual. Therefore, privacy threats are reduced while still allowing information to be collected.

Spiekermann and Cranor (2009) state that “Identifiability can be defined as the degree to which (personal) data can be directly linked to an individual”. The degree of privacy of a system is inversely related to the degree of user data identifiability (Pfitzmann and Hansen, 2010). The more identifiable data that exists about a person, the less that person is able to control access to information about herself/himself, and the greater the privacy risks. Identifiability ranges from completely identified to anonymous. Throughout this section, we survey different studies in MAS that prevent information processing through minimizing the collection of identifiable data.

Figure 2.3 depicts a conceptual map for all of the studied approaches that provide support for protecting against information processing.

2.3.1. Anonymity

Anonymity is the maximum degree of privacy, so it plays a crucial role in preserving privacy in agent technologies (Brazier et al., 2004). The main property of anonymity is that collected data cannot later be attributed to a specific individual. Anonymity is commonly defined in terms of a possible attacker. Pfitzmann and Hansen (2010) define anonymity as “Anonymity of a subject from an attacker's perspective means that the attacker cannot sufficiently identify the subject within a set of subjects, the

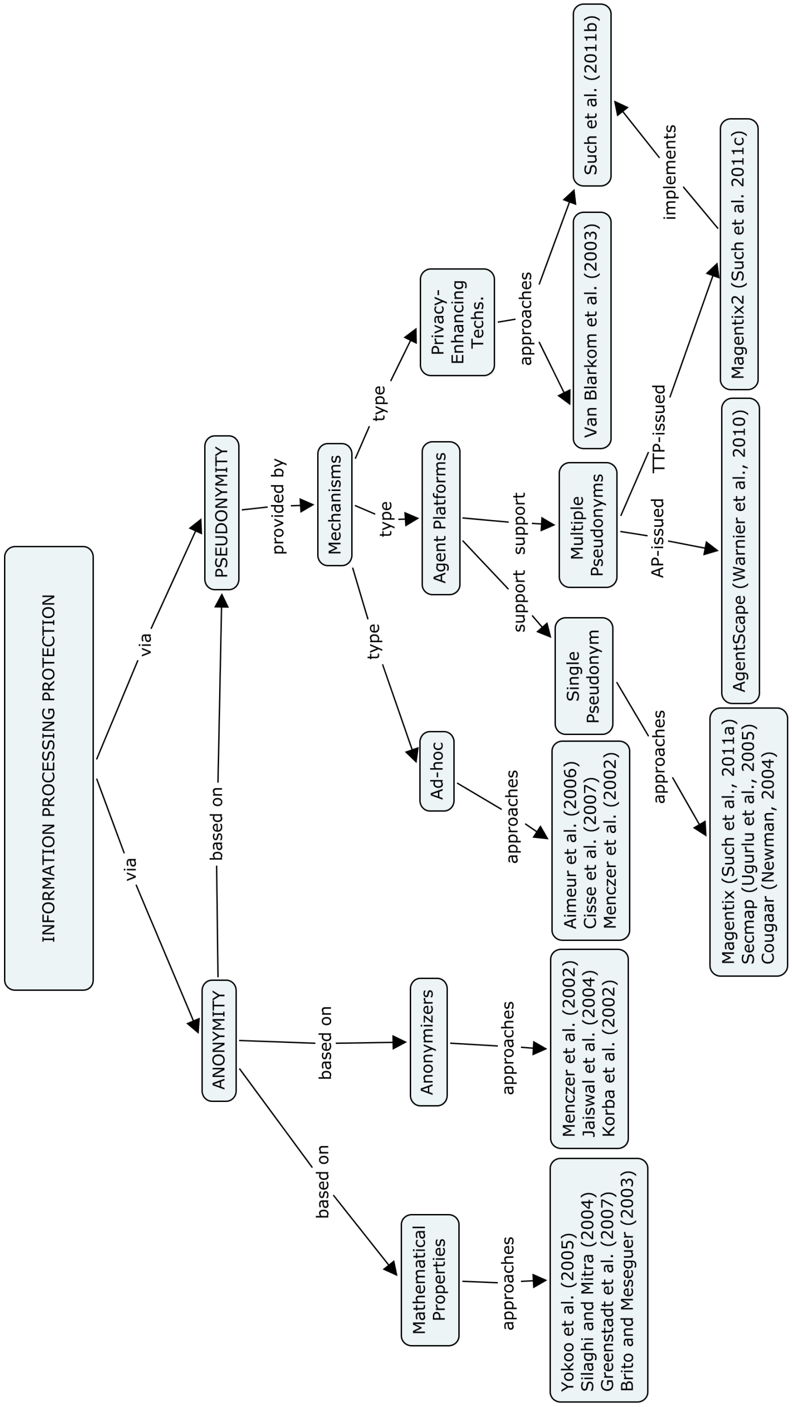

Figure 2.3: Information Processing Conceptual Map

anonymity set.”

Many anonymity systems can be modeled in terms of unlinkability (Diaz, 2006). Pfitzmann and Hansen (2010) define unlinkability as follows: “Unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an attacker’s perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not”. Anonymity can be achieved when a given IOI cannot be linked to a given subject. For instance, the sender of a message is anonymous if the message cannot be linked to a particular subject from a set of subjects that may have potentially sent the message (the anonymity set).

The agent community has developed algorithms that aim at preserving anonymity in Multiagent problem solving, including both distributed constraint satisfaction (DisCSP) and distributed constraint optimization (DCOP). In these problems, agents need to share information in order to solve a problem of mutual interest. The major concern in DisCSP and DCOP algorithms is that they usually leak information that can be exploited by some agents to infer private information of other agents (Greenstadt et al., 2006). The anonymity set here is the set of agents that share information. The main aim of these protocols is that shared information cannot be linked to the corresponding agent. A typical application is that of meeting scheduling in which agents arrange meetings according to their principals’ schedules. Private information in these problems usually refer to information about: (i) agent preferences (domain privacy), i.e., whether an agent can attend a meeting in each time slot in DisCSP or the utility valuations for each agent for each time slot in DCOP; and (ii) the assignment for each agent once a final solution is reached as well as partial solutions during the solving process (assignment privacy).

Yokoo et al. (2005), and Silaghi and Mitra (2004) present secure DisCSP algorithms based on multi-party computation. Multi-party techniques compute general functions with secret inputs. Therefore, these techniques allow the collection of information in a way that cannot be linked back to the agents. Theoretical proofs show that these secure DisCSP algorithms do not leak private information, i.e., there is no chance for either an agent or an external entity to link variable assignment to the agents taking part in the problem solving process. However, these approaches have a high computational cost. DisCSP algorithms require many comparisons and these protocols require exponentiation operations for each comparison. Therefore, these protocols should be used when privacy concerns are very high.

There are other computationally cheaper approaches, such as (Greenstadt et al., 2007) and (Brito and Meseguer, 2003). However, these approaches still leak information. These works try to reduce privacy loss of existing DisCSP/DCOP algorithms. Metrics based on the Valuations of Possible States (Maheswaran et al., 2006) framework are usually considered to quantify the reduction in privacy loss. Greenstadt et al. (2007) present the DPOP with Secret Sharing (SSDPOP) algorithm, which is an extension of DPOP based on the efficient cryptographic technique of secret sharing. Agents use secret sharing to send aggregate values, and, thus, they do not reveal their individual valuations. Brito and Meseguer (2003) present the Distributed Forward Checking (DisFC) algorithm, which is an approach without using cryptography. In DisFC, agents exchange enough information to reach a global consistent solution without making their own assignment public. To this aim, DisFC sends filtered domains (agent preferences) to other agents and replaces their own value by a sequence number.

Anonymizers

There are technologies developed outside the agent community called anonymizers. These anonymizers can be used to obtain communication anonymity. They hide the IP address and other whereabout information from the messages they receive and forward these messages (Menczer et al., 2002). If an agent sends a message to another agent using these anonymizers, the receiving agent is not able to identify the sender from the potential senders (the anonymity set in this case). Chaum (1981) first introduced MIX-networks as a means to counteract traffic analysis. A MIX-network is composed of a set of MIX nodes. Each MIX node receives a number of messages, modifies them (using some cryptographic transformation), and sends them randomly. Moreover, each MIX node in the network knows only the previous and next node in a received message's route. Therefore, an external observer is not able to correlate incoming and outcoming messages.

Onion routing (Goldschlag et al., 1999) is based on chaum's MIX-networks. While MIX nodes could store messages for an indefinite amount of time while waiting to receive an adequate number of messages to mix together, an onion router is designed to pass information with low latency. However, large message traffic is vital to strengthen anonymity in onion router networks. An example of implementation of onion routing is Tor (Dingledine et al., 2004).

Anonymizers also prevent an external observer from inferring a possible relationship between the sender and the receiver of a message (known as traffic analysis). Although authorization and confidentiality are ensured by secure APs (explained in section 2.2.2), a potential attacker could also gather information about who is communicating with whom. This is because the IP of both sender and receiver can be known even though the content of the message exchanged is encrypted. Moreover, this potential attacker could also know how often two agents communicate to each other, and then infer relationship patterns between the two agents.

There are some agent architectures and implementations that use anonymizers (Menczer et al., 2002; Jaiswal et al., 2004). IntelliShopper (Menczer et al., 2002) is an intelligent shopping agent that aids customers who are shopping for a product on an e-commerce site. This agent is in charge of monitoring e-commerce sites to notify the customer about updates related to products she/he is interested in. To this aim, the agent is able to collect information about the customer's activities at an e-commerce site to determine interesting products and decisions about whether or not she/he buys the products. Customers connect to IntelliShopper through an anonymizer so that IntelliShopper cannot know about the IP and other whereabout information of the requests it receives. The MAGNET architecture (Jaiswal et al., 2004) provides support for auction-based business-to-business marketplaces. MAGNET agents participate in auctions that are reverse, i.e., contracting agents present call for bids to supplier agents. MAGNET uses an anonymizer between the market and the bidders. This is intended to reduce market-supplier collusion by making the supplier's bids unlinkable until the end of auction.

There are also anonymizers specially developed for APs. Korba et al. (2002) present an alternate Onion Routing approach for Jade. Each Jade AP has several onion agents that provide an anonymous data forwarding service, and at least one onion monitor agent that keeps track of the location of all the other onion agents in the system. Onion monitor agents exchange information in order to maintain a valid topology of the complete onion agent network. The main drawback of this approach is that agents in Jade do not communicate directly with each other; instead, it is the *container* where agents live that finally sends the message over the network to the container where the recipient agent lives. Therefore, an external observer could track the path of a message through the containers and infer possible relations among agents living in these containers. The lower the number of agents in a container, the higher the probability for an external observer to link the message to one particular agent (as sender or receiver). Moreover, the Jade agent platform itself could monitor the path of a message from one agent to another. A possible solution for this may be that containers connect to each other through the general purpose anonymizers presented above.

2.3.2. Pseudonymity

Pseudonymity (Chaum, 1985) is the use of pseudonyms as identifiers. A pseudonym is an identifier of a subject other than one of the subject's real names (Pfitzmann and Hansen, 2010). Pseudonyms have been broadly used by human beings in the real world. For instance, in the 19th century when writing was a male-dominated profession, some female writers used male names for their writings. Nowadays, in the digital world, there are a great number of pseudonyms such as user-names, nicknames, e-mail addresses, sequence numbers, public keys, etc.

The most important trait of pseudonymity is that it comprises all degrees of identifiability of a subject (from identified to anonymous) depending on the nature of the pseudonyms being used. Complete identification is when the linking between a pseudonym and its holder is publicly known. Anonymity can be achieved by using a different pseudonym for each different transaction known as transaction pseudonyms (Chaum, 1985), unless the information contained in these transactions establishes linkability (Bhargav-Spantzel et al., 2007).

There are some agent-based approaches that implement ad hoc mechanisms for implementing pseudonymity. Some of these approaches have been proposed in the agent-based information filtering domain (Aïmeur et al., 2006; Cissée and Albayrak, 2007). Cissée and Albayrak (2007) provide an approach based on transaction pseudonyms. They aim at providing anonymity, i.e., the recommendations must not be linkable to the identity of the principal if the principal wants these recommendations to be unlinkable to himself. They propose that principals use a different agent each time they ask for a recommendation. Aïmeur et al. (2006) provide a completely different approach; the principal "identifies" herself/himself when she/he communicates with the service provider by using a pseudonym. Thus, the service provider can build a profile to better aid recommendations but without establishing linkability to the identity of the agent's principal. However, none of these approaches consider that the principal may want some recommendations to be unlinkable while others be linkable; instead they provide the use of either only one pseudonym or a different pseudonym for each recommendation.

Other approaches to pseudonymity come from the agent-based e-commerce domain (Menczer et al., 2002). Users connect to the IntelliShopper agent using a pseudonym to avoid the link between the profiles that IntelliShopper has about customers and their real identity. Moreover, users can use different pseudonyms for IntelliShopper to have separate profiles for separate activities. Therefore, users can decide whether or not to use a new pseudonym in each transaction, instead of forcing the same pseudonym for all transactions or a different pseudonym for each transaction (as in the approaches described in the above paragraph). However, the authors of this work leave the user with the responsibility of creating their pseudonyms and they do not provide any pseudonym management facility.

Another approach for providing general support for pseudonymity for agent technologies instead of ad-hoc solutions is to provide this support from APs. Thus, this support aids agent developers to use pseudonymity without having to implement their own solutions. However, only a few of the APs explained in section 2.2 implement some kind of support for pseudonymity. Magentix (Such et al., 2011) (detailed in

Chapter 3), Secmap (Ugurlu and Erdogan, 2005), AgentScape (Quillinan et al., 2008) and Cougaar (Newman, 2004) assign a unique identity for each agent that it can use to authenticate itself to other agents. Using this identity, agents can act pseudonymously, i.e., agents can act on behalf of their principal without using the identity of their principal. However, agents cannot hold more than one pseudonym, i.e., principals should use a different agent each time they want to use a different pseudonym (similarly to what is proposed by Cissée and Albayrak (2007) explained above).

Warnier and Brazier (2010) also present a mechanism for the AgentScape AP that offers pseudonymity by means of what they call *handlers*. Handlers are pseudonyms that agents can use to send/receive messages to/from other agents. At will, agents can request the AP for new handlers. Moreover, the AP is the only one that knows the association between handlers and GUIDs (global unique identities of the agents). An agent can also obtain anonymity by simply using a different handler for each transaction (transaction pseudonyms). AgentScape also offers an automatic anonymity service. Agents can send messages anonymously without having to manage pseudonyms. This service is provided by agents called *anonymizers*[3]. When an agent wants to send a message anonymously, this message is redirected to an anonymizer. Then, this anonymizer is in charge of removing the original handler of the sender from the message, replacing it with another (possibly new) handler, and sending the message to the intended recipient. If the intended recipient replies, this reply is forwarded to the sender of the original message. The original sender of the message must notify when a transaction ends. For each new transaction, the anonymizer generates a new handler.

APs that provide support for pseudonymity (e.g. by providing APIs to create and manage pseudonyms) do not consider that pseudonyms can be issued by external third parties. That is, APs themselves are in charge of issuing the pseudonyms. Thus, the AP itself (and the anonymizer agents for the case of AgentScape) must be trusted. This is because the AP knows the relation of pseudonyms to each other and to the principal involved. This usually implies that the organization or company that hosts the specific system (e.g. eBay in the case of an e-marketplace) knows the association of

[3] Note that these anonymizers are not the same as the ones presented in section 2.3.1.

pseudonyms to each other and to principals. Therefore, this organization or company can collect and process information about the principals that run their agents on the system.

Other more general approaches have been proposed to provide pseudonymity to agent technologies (van Blarkom et al., 2003). These approaches propose the integration of Privacy-Enhancing Technologies (PETs) (Senicar et al., 2003) into agent technologies. PETs can be defined as “a system of ICT measures protecting informational privacy by eliminating or minimising personal data thereby preventing unnecessary or unwanted processing of personal data, without the loss of the functionality of the information system” (van Blarkom et al., 2003).

Van Blarkom et al. (2003) propose the use of Identity Protectors. Identity Protectors are PETs that are in charge of converting the identity of the principal involved (the person whose data are being processed) into one or more pseudonyms. They propose that the Identity Protector is placed either between the principal and the agent or between the agent and the environment. The Identity Protector in an information system can take several forms: (i) a separate function implemented in the data system; (ii) a separate data system supervised by the individual; (iii) a data system supervised by a trusted third party. They present in which places of a specific agent architecture an Identity Protector can be placed. However, they do not provide any specific design or implementation of an Identity Protector and the integration of it into an agent architecture.

2.3.3. Implications in Security, Trust, and Reputation

As stated in section 2.2, security plays a crucial role in preventing undesired information collection. However, there are also security concepts themselves that can represent an actual threat to privacy (Petkovic and Jonker, 2007) even though they are mandatory for the system to be secure. These security concepts include authentication and accountability. Minimizing data identifiability may affect authentication and accountability if specific countermeasures are not considered.

Authentication binds a principal to a digital representation of her/his identity

(Bishop, 2002). To authenticate something on the Internet is to verify that its identity is as claimed (Jøsang et al., 2001). In the case of a message, the function of authentication is to assure the recipient that the message is from the source that it claims to be from (Stallings, 2010). For instance, if an agent A sends a message to an agent B, B should be able to *authenticate* A as the sender of the message. Authentication of the entities existing in an AP is the basis for confidentiality (Such et al., 2011), explained in section 2.2. All of the APs that provide some support for pseudonymity (Magentix (Such et al., 2011), Secmap (Ugurlu and Erdogan, 2005), AgentScape (Quillinan et al., 2008) and Cougaar (Newman, 2004)) support authentication based on pseudonyms.

The other security concept that has a direct impact on privacy (and vice versa) is accountability. Accountability refers to the ability of holding entities responsible for their actions (Bhargav-Spantzel et al., 2007). Accountability helps to promote trust in the system. This is because if an agent misbehaves and there are no accountability consequences, there is a sense of impunity that could even encourage abuse. This trust is crucial for systems in which users can be seriously damaged by losing money, such as agent-based e-commerce (Fasli, 2007b).

Accountability usually requires an unambiguous identification of the principal involved (Bishop, 2002). Then, this principal can be held liable for their acts. For instance, a customer agent pays a vendor agent for a good. The agent vendor commits to shipping the good to the customer agent's principal. In the event that the customer agent's principal does not receive the good, the vendor's principal[4] may be held liable for this. Although determining exactly who should be held liable for this depends on the applicable laws in the specific country, it usually requires the identification of the vendor's principal. Then, the vendor agent's principal can be sued for fraud.

Pseudonyms can be utilized to implement accountability (Hansen et al., 2004).

[4] Software entities (intelligent agents, virtual organizations, etc.) cannot have real identities because, until now, they could not be held liable for their acts in front of the law. However, this may change in the future if they finally achieve some kind of legal personhood, as suggested by Chopra and White (2004) and Balke and Eymann (2008). In this case, software entities may be provided with legal personhood to be (partially) held liable for their acts. The point is that according to the law, someone must be liable for frauds like this.

AgentScape and Magentix keep track of the association between principals and pseudonyms. Therefore, these two APs can disclose the principal behind the pseudonym, removing pseudonymity and producing identity and accountability as a result. The main drawback of this approach is that the AP itself (including the anonymizer agents for the case of AgentScape) must be trusted. This is because the AP knows the relation of pseudonyms to each other and to the principal involved. Although this is needed to ensure accountability (agent principals can still be held liable for their agents behavior even when pseudonyms are used), this usually implies that the organization or company that hosts the specific marketplace (e.g. eBay) knows the association of pseudonyms to each other and to principals. Therefore, this organization or company can collect and process information about the principals that run their agents in the marketplace.

In a Multi-agent System, agents usually need to assess trust towards other agents as well as their reputation. To this aim, the agent community has developed a vast number of trust and reputation models (Ramchurn et al., 2004; Sabater and Sierra, 2005). An agent can build a reputation by using the same pseudonym more than once. In the same way, an agent can be trusted by a transaction partner by using the same pseudonym for different transactions. Current Trust and Reputation models are usually based on the assumption that pseudonyms are long-lived, so that ratings about a particular entity from the past are related to the same entity in the future. However, when these models are actually used in real domains, this assumption is no longer valid. For instance, an agent that has a low reputation due to its cheating behavior may be really interested in changing its identity and restarting its reputation from scratch. This is what Jøsang et al. (2007) called the *change of identities* problem. This problem has also been identified by other researchers under different names, e.g., *whitewashing* (Carrara and Hogben, 2007).

Kerr and Cohen (2009) also point out the fact that entities could create new accounts (identity in the system) at will, not only after abandoning their previous identity but also holding multiple identities at once. This is known as the *sybil* attack (Jøsang and Golbeck, 2009). An example of this attack could be an agent that holds multiple identities in a marketplace and attempts to sell the same product through each of them, increasing the probability of being chosen by a potential buyer.

These vulnerabilities can be more or less harmful depending on the final domain of the application. However, these vulnerabilities should, at least, be considered in domains in which trust and reputation play a crucial role. For instance, in e-marketplaces, these vulnerabilities can cause users to be seriously damaged by losing money. Another example can be a social network like Last.fm[5] in which users can recommend music to each other. A user who always fails to recommend good music to other users may gain a very bad reputation. If this user creates a new account in Last.fm (a new identity in Last.fm) her/his reputation starts from scratch, and is able to keep on recommending bad music. Users may be really bothered by these recommendations and move to other social networks. In this case, the one seriously damaged is the social network itself by losing users.

A possible solution for these vulnerabilities is the use of *once-in-a-lifetime* pseudonyms (Friedman and Resnick, 1998). Agents can only hold one *once-in-a-lifetime* pseudonym in each marketplace. Therefore, they cannot get rid of the trust other agents have in them as well as the reputation they earned in the Multi-agent System. This model needs the existence of trusted third parties called Identity Providers to issue and verify pseudonyms. While this may not be a difficulty in networks such as the Internet, this may not be appropriate in environments with very scarce resources such as sensor networks.

There are also other solutions for identity-related vulnerabilities of trust and reputation models that can be used when trusted third parties cannot be assumed (Hoffman et al., 2009). Yu et al. (2006) present an approach based on social networks represented as a graph in which nodes represent pseudonyms and edges represent human-established trust relationships among them in the real world. They claim that malicious users can create many pseudonyms but few trust relationships. They exploit this property to bound the number of pseudonyms to be considered for trust and reputation. However, this approach is not appropriate for open MAS in which agents act on behalf of principals that may not be known in the real world. (Cheng and Friedman, 2005) have demonstrated several conditions using graph theory that must be satis-

[5] `http://www.last.fm`

fied when calculating reputation in order for reputation models to be resilient to sybil attacks. This approach needs a particular and specific way to calculate ratings about an individual. Thus, this approach cannot be applied to trust and reputation models that follow other approaches for managing trust and reputation ratings.

2.4. Protection against Information Dissemination

Information dissemination refers to the transfer of previously collected and possibly processed data to other third parties. It should be pointed out that protection against dissemination in an open environment such as open Multi-agent Systems is a very hard problem. This is mainly because when a sender agent passes information to a receiver agent, the former usually loses control over that information. Moreover, it is very difficult for the sender agent to verify whether or not the receiver agent passes this information to other third parties. In the following, we outline some approaches to protect against information processing based on concepts usually used in agent-based technologies: trust and reputation, and norms.

Figure 2.4 depicts a conceptual map for all of the studied approaches that provide support for protecting against information dissemination.

2.4.1. Based on Trust and Reputation Models

One approach to prevent information dissemination is based on trust and reputation models. There are works that assume that the reputation of another agent with regard to how they use the information they collect/process is available (Yassine and Shirmohammadi, 2009). Thus, agents can choose not to send information to agents that have a bad reputation. In this case, having a bad reputation means that the agent usually disseminates personal information about other agents. To measure the trustworthiness of agents regarding whether they disseminate personal information or not, one of the many trust and reputation models developed by the agent community could be used (refer to Ramchurn et al. (2004), and Sabater and Sierra (2005) for reviews on trust and reputation models).

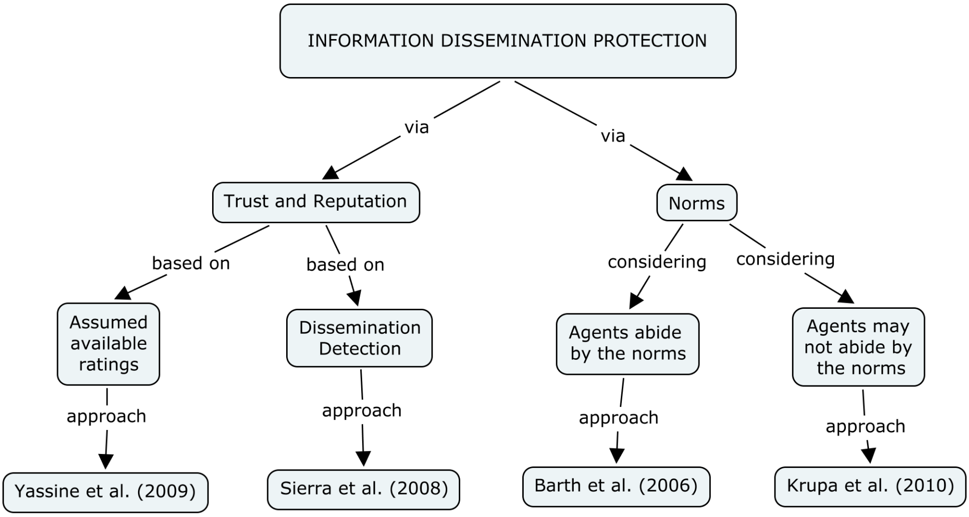

Figure 2.4: Information Dissemination Conceptual Map

These models usually need to verify the behavior of an agent in the past to predict their future behavior. However, how could an agent verify that another agent has disseminated information about it? The verification of whether or not an agent disseminates personal information about other agents or not is not straightforward. One approach could be that an external entity controls all the communications among agents. Thus, this external entity is able to know if an agent is disseminating information about another. This approach, however, cannot be applied due to its privacy implications (this external entity would act as a *big brother*). Instead, we envision Multi-agent Systems in which communications between each agent pair can be encrypted (by using mechanisms such as the ones presented in Section 2.2.2) to avoid undesired information collection by any other external entity.

Sierra and Debenham (2008) present a model for detecting undesired information dissemination based on information-theoretic measures. They consider that agents are uncertain about their world model. An agent estimates the amount of information that another agent possibly disseminated about the former agent from the information in the messages that the agent receives from other agents. To this aim, the agent sets update functions of its uncertain world model based on the messages received. For instance, if an agent A sends to agent B that A likes the color pink, agent A can set an update function of the messages received that scan for information related to the color pink. Thus, if an agent C sends agent A a message offering pink dresses, A could infer that B probably disseminated its color preferences to C. According to this, agent A can revisit the trustworthiness of agent B regarding information dissemination. This model only considers what an agent can observe by itself. However, other agents could also warn this agent about the fact that another agent is disseminating information about it. For instance, in the previous example, if agent A and agent C are known to each other, agent C may not take advantage of knowing A's color preferences. Instead, agent C can warn agent A that agent B disseminates information about it.

2.4.2. Based on Normative Multi-agent Systems

This approach is based on using norms for governing the dissemination of information in a so-called Normative Multi-agent System (Criado et al., 2011). According to Boella et al. (2006), a norm is "a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behavior". In this case, proper and acceptable behavior means that agents should not be able to disseminate sensitive information about other agents without their consent.

Barth et al. (2006) present a logical framework for expressing and reasoning about norms of transmission of personal information. This framework formalizes the main ideas behind contextual integrity. Contextual integrity (Nissenbaum, 2004) is a legal framework for defining the appropriateness of the dissemination of information based on the context, the role of the entities taking part in the context, and the subject of the personal information being transferred. In the framework of Barth et al. (2006), privacy norms are expressed as linear temporal logic (LTL) formulas. These formulas are used to define the permissions and prohibitions when disseminating private information about other agents. For instance, it can be expressed that, in a medical context, an agent playing the role of doctor is allowed to pass medical personal information to an agent only if this agent is the subject of the information and this agent is playing the role of patient. Note the difference to a role-based access control (RBAC) approach, which allows the definition of permission based on the roles of the entities taking part in the system. However, it cannot use the information about who is the subject of the information being transferred. Barth et al. (2006) assume a closed system in which all agents abide by the norms, so they do not provide any enforcement mechanism.

Krupa and Vercouter (2010) present a position paper that includes an initial proposal for controlling personal information dissemination in open Multi-agent Systems based on contextual integrity (explained above). They consider that agents may not abide by the norms. They propose five privacy-enforcing norms to promote privacy-preserving behaviors when disseminating information: (i) respect the appropriateness of the information to be transferred according to contextual integrity, i.e., agents should not transfer information that is not appropriate regarding the context, the roles of the agents involved, and the subject of the information; (ii) sign the transmission chain before sending the information, so agents that transmit information remain liable for this transmission; (iii) do not send information to violating agents, i.e., agents that do not abide by the norms; (iv) delete information received from violating agents so that this information is no longer transferred; and finally, (v) punish agents violating these norms (including this one) by sending *punishment messages* (messages that inform that a given agent has performed a violation) to the subject of the information and also other agents in the system.

Krupa and Vercouter (2010) suggest the use of trust models based on the punishment messages to isolate violating agents, i.e., if an agent is said to violate norms, other agents will not send personal information to it. Thus, trust and reputation models can be used based on these punishment messages. This is because, in this case, it is assumed that all of the agents will follow the norms, and in the event of not doing so, these punishment messages can act as the verification mechanism needed for trust and reputation models, as explained in Section 2.4.1. However, this work is at an initial stage. Specifically some major issues remain open (according to the authors): (i) the real connection of their proposal to trust models needs to be specified; (ii) two or more agents can easily collude by passing information to each other without other *benevolent* agents being aware of it; (iii) one agent can consider that another agent is not trustworthy according to its trust model, while another can consider it to be trustworthy (i.e., some transmissions can be viewed as appropriate by an agent and the same transmissions can be viewed as inappropriate by another agent); and (iv) the system can be subject to strategic manipulation, such as agents sending fake punishment messages that do not really correspond to real violations.

2.5. Conclusions

In this chapter, we introduced the issue of privacy preservation and its relation to Multi-agent Systems. We identified the possible privacy breaches that can occur in Multi-agent Systems. We also surveyed the state of the art on studies that fall on the intersection between privacy and Multi-agent Systems. Moreover, we classified these studies according to the information activity prevented and the approach followed to do so.

Although we have presented many studies that provide satisfactory solutions for some specific problems, we consider that research on privacy and Multi-agent Systems is still in its infancy. As pointed out during this survey, there are still a great number of possible research lines that remain unexplored. Some of them are addressed in this thesis, others are left as future work. Open challenges for future work are described in Chapter 7. In this thesis we focus on addressing:

- As pointed out in section 2.2.2, once an agent has decided what information to disclose to which other agent, this information must be protected from accesses from any other third party different from the agent to which the information is intended to. This includes parties from their local computer and network but also different locations, even across the Internet. We contribute, in Chapter 3, a secure AP that allow agents to communicate with each other in a confidential fashion, i.e., external third parties cannot access the information that two agents exchange. Moreover, this agent platform allows agents to communicate with each other without disclosing their principals' identities, which remain hidden. Although principals' identities are not known a priori, principals' identities can be obtained for accountability concerns (e.g. law enforcement).

- APs that provide support for pseudonymity (e.g. by providing APIs to create and manage pseudonyms) do not consider that pseudonyms can be issued by external third parties (refer to section 2.3.2). That is, APs themselves are in charge of issuing the pseudonyms. The main drawback of this approach is that the AP itself (and the anonymizer agents for the case of AgentScape) must be trusted. This is because the AP knows the relation of pseudonyms to each other and to the principal involved. This usually implies that the organization or company that hosts the specific system (e.g. eBay in the case of an e-marketplace) knows the association of pseudonyms to each other and to principals. Therefore, this organization or company can collect and process information about the principals that run their agents on the system. Moreover, as pointed out in section 2.3.3, using pseudonyms can have a direct impact on accountability, trust, and reputation. We propose, in Chapter 4, an identity management model for agents in a Multi-agent System. This model allows agents to hold as many identities as needed for minimizing data identifiability, i.e., the degree by which personal information can be directly attributed to a particular principal. Privacy is enhanced without compromising accountability and other crucial aspects for agents in a Multi-agent System, such as trust and reputation. In Chapter 5, we describe our implementation and integration of this model into the Magentix2[6] Agent Platform (AP).

- There are many cases where the direct benefit of disclosing personal information is not known in advance, as explained in section 2.2.1. This is the case in human relationships, where the disclosure of personal information in fact plays a crucial role in the building of these relationships (Green et al., 2006). These relationships may or may not eventually report a direct benefit for an individual. For instance, a close friend tells you what party he voted for. He may disclose this information without knowing (or expecting) the future gain in utility this may cause. Indeed, it might not report him any benefit ever. We propose in Chapter 6 a decision-making model for agents to decide whether or not to disclose personal information to other agents is acceptable or not. This model is based on psychological findings regarding how humans disclose personal information

[6] http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/index.php

in the building of their relationships. This model considers intimacy on the one hand and privacy loss on the other hand.

CHAPTER 3

A Secure Agent Platform

Contents

3.1. Introduction

As stated in (Petkovic and Jonker, 2007), privacy protection is only possible using secure systems. Security and privacy are often related to each other but they are not the same (Head and Yuan, 2001). On the one hand, security is the control of information in general (Camp, 1996). Then, information is secure if the *owner* of information can control that information. On the other hand, information is private if the *subject* of information can control that information. Therefore, privacy requires security to control access and distribution of information (Garfinkel, 2009).

Security related studies in the MAS research field have been increasing over the last few years due to the fact that an agent's incorrect or inappropriate behavior may cause non-desired effects such as money and data loss. Moreover, the lack of security in some current MAS-based applications is one of the reasons why MAS technology is barely used in industry.

As stated in (Longstaff et al., 1997), applications on the Internet need to be concerned about basic security concepts such as confidentiality, integrity, authentication and authorization. Confidentiality means that information is only read or copied by someone authorized to do so; integrity means that only authorized users[1] can modify information; authentication means proving that a user is who she claims to be; and finally, authorization is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity. As APs are the environment for running MASs, APs must be concerned about confidentiality, integrity, authentication and authorization taking into account the entities existing in an AP: agents and users (usually known as agent owners).

There are a lot of APs developed by the agent community, but currently only a few of them take security concerns into account. Although these security-concerned APs provide baseline security features such as authentication, authorization, integrity and confidentiality, they fail to support for preserving the privacy of the identity of the users that run their agents on such APs while preserving accountability. What is more, none

[1] In this chapter, we use the term user instead of principal. This is to differentiate from *Kerberos principals*, which can be seen as identities that Kerberos is able to authenticate.

of these APs provide mechanisms related to the sociability skills of agents, such as agent groups. In this chapter, we propose a security infrastructure (SI) for Magentix (Alberola et al., 2008). In addition to the concepts previously stated (authentication, authorization, integrity and confidentiality), this SI is focused on agent groups and user identity privacy. The SI presented is based on identities that are assigned to all of the different entities that can be found in Magentix, i.e., users, agents and agent groups. The authentication of these identities by any entity of the AP is the basis of the Magentix SI.

The rest of this chapter is organized as follows. Section 3.2 presents a state-of-the-art of security-concerned APs. Section 3.3 presents the security infrastructure for Magentix. Section 3.4 presents an evaluation of the SI, describing an application successfully built using Magentix and a performance evaluation of it. Finally, section 3.5 presents some concluding remarks.

3.2. Overview of Security-Concerned Agent Platforms

Current security-concerned APs provide secure message exchanges by means of integrity and confidentiality of the data exchanged. This is ultimately achieved by signing and ciphering messages before transferring them and checking signatures and deciphering messages when they are received.

Authentication and authorization are also provided, but in a different way. On the one hand, there are APs that authenticate and authorize agents regarding the identity of the users that create them (known as the agent owners). On the other hand, there are APs that authenticate and authorize agents using a unique identity associated to each agent.

Authentication of the agents in Jade (JADE Board, 2005), Semoa (Roth and Jalali-Sohi, 2001), Tryllian (Xu and Shatz, 2003), Voyager (Recursion Software Inc., 2008) and the one presented by Ismail (Ismail, 2008) are based on the identities of the agent owners or creators. In this sense, all of the agents owned by the same user share her identity. Whenever one agent is interacting with another, mutual authentication is needed so that the other agent is always aware of the former's owner's identity. This may be a privacy problem, because all of the secure interactions between two agents require the disclosure of their respective owner's identity.

Regarding authorization, APs that authenticate agents based on their owner's identity do not include a fine-grained access control to resources based on agents, i.e., a compromised agent would gain all of its owner permissions. Hence, there is no way of somehow "confining" an agent to only being allowed to access the resources she needs but not all of the resources her owner can access.

Secmap (Ugurlu and Erdogan, 2005) and Cougaar (Newman, 2004) assign a unique identity for each agent. In this sense, agent interactions can be carried out without the need to disclose agent owner identities. Therefore, in these two APs the privacy of the owner identities is preserved. However, there are some scenarios where either the identity, or at least a credential (e.g. credit card number), of the agent owner is needed. For instance, an agent can be bidding for a good in an auction. If she commits to buying the good she must buy it, and in the event of her not buying the good, someone (maybe her owner) should be held liable for this.

Authorizing agents by means of a unique identity per agent allows a fine-grained access control, but in turn, security policies become more complex because more entities are taken into consideration.

In order to improve user identity privacy, agents should have a unique identity. In this way, all of the interactions among agents that do not specifically need the identity of their owner can be carried out preserving user identity privacy. Moreover, the underlying infrastructure must relate agent identities with their owners' identities (managing a private log for instance) so that accountability is not lost. The identity of an agent owner is then known in case an agent behaves inappropriately so that an agent owner can be punished if necessary.

Regarding agent groups, none of the APs presented above support agent groups except Jade (Baldoni et al., 2010). However, the security mechanism and the group mechanism of Jade are not related to each other. Therefore, an integration between both mechanisms to allow secure groups is not provided. Other APs like MadKit

(Gutknecht and Ferber, 2000) provide agent groups but do not take security into consideration. Not including security when supporting agent groups can cause multiple security flaws, e.g. an agent could act on behalf of a group without being part of that group.

The SI presented in the following section takes into consideration agent groups, which are defined in terms of their structure and interactions. In this way, the security infrastructure supports agent groups to be shaped as teams, coalitions, hierarchies, federations, congregations and other well-known agent organizations in a secure fashion.

3.3. Magentix Security Infrastructure

Magentix[2] AP (Alberola et al., 2008) is a distributed AP composed of a set of computers executing the Linux OS. It is implemented in the C language and using the Linux OS services. There are two kinds of agents in Magentix: platform and user agents. Platform agents are in charge of providing the basic services that the AP offers while user agents make up the MAS being developed. The basic services provided by Magentix are briefly summarized bellow (for an in-depth description refer to Alberola et al. (2008)).

The **Magentix (MGX)** service is in charge of managing the topology of the AP, i.e., which hosts make up the AP.

The **Agent Management System (AMS)** service is defined by FIPA (FIPA, 2001a) and offers the white pages functionality. This service stores the information regarding the user agents that are running on the AP.

The **Directory Facilitator (DF)** service is also defined by FIPA and offers the yellow pages functionality. This service stores the information regarding the services offered by user agents. The *DF* service allows user agents to register services offered by them, deregister these services or look up user agents that offer a specific required

[2] `http://www.dsic.upv.es/users/ia/sma/tools/Magentix/index.html`

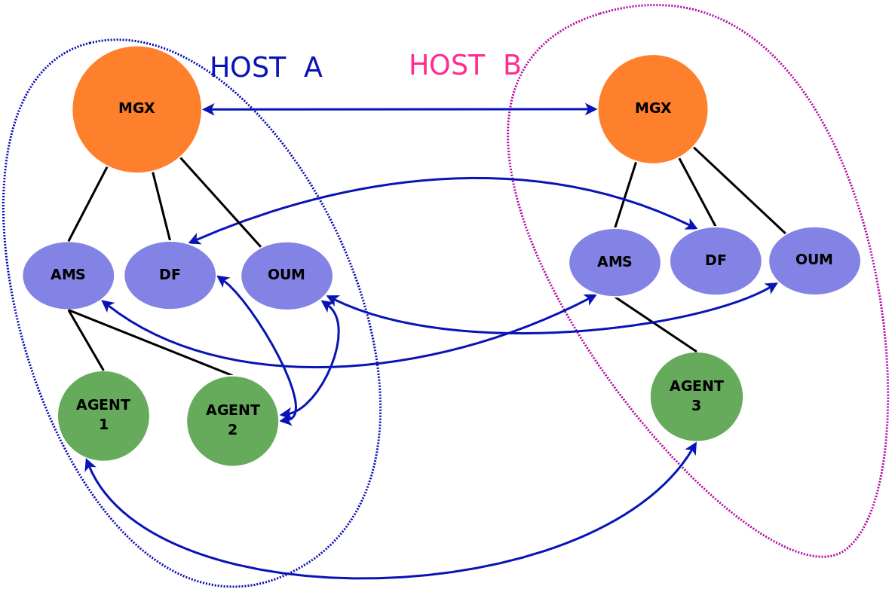

Figure 3.1: Magentix structure

service.

The **Message Transport Service (MTS)** is the only service which has not been developed as a platform agent but as a function library. Magentix agents (both platform and user agents) use this library to carry out their communications.

The **Organizational Unit Manager (OUM)** service provides support oriented to agent-group communication. Once a group is created, the user is able to specify one or more agents to receive the messages addressed to the group (called *contact agents*). The user can also specify the way in which these messages have to be delivered to the *contact agents* according to one of the *routing types* available.

The **Security Infrastructure (SI)** is based on identities that are assigned to all of the different entities that can be found in Magentix, i.e., users, agents and agent groups. Agents are created by users and can be grouped into agent groups. Each entity in the AP is represented by its identity. The authentication of these identities by any other entity of the AP is the basis for achieving the other desired features. An identity in Magentix is a Kerberos[3] (Neuman et al., 2005) principal and follows the form principal@MAGENTIX. From this moment on, the terms identity and principal are used indistinctly.

Kerberos is a network authentication protocol. It provides applications based on client/server paradigm with strong authentication using symmetric cypher encryption. Kerberos protocol uses strong cryptography allowing a client to prove its identity to a server (and vice versa) through an insecure network connection. When both the client and the server prove their identity, they can also encrypt all the communications in order to assure exchanged data confidentiality and integrity. Kerberos makes use of a trusted third party, termed a Key Distribution Center (KDC). Kerberos works on the basis of *tickets* which serve to prove the identity of the entities. The KDC maintains a database of secret keys; each entity on the network shares a secret key known only to itself and to the KDC. Knowledge of this key serves to prove an entity's identity. For communication between two entities, the KDC generates a session key which they can use to secure their interactions.

Entities in Magentix (users, agents and agent groups) and how the SI treats them is described in the following sections.

3.3.1. Magentix Users

In Magentix, there is the user concept. These users match a Kerberos *principal* and follow the form user@MAGENTIX. Do not confuse the MAP users with the local users of a Unix machine. Therefore, a MAP user has to login in the system in the conventional way, and then authenticate itself to Magentix (authenticating to the KDC) running the program mgx_login that is a wrapper for the kinit of the Kerberos distribution. For instance, let us have a Linux machine with a user bob. bob sits in front of the Linux machine, logs in the system an starts using it. When he needs to launch an agent in the running Magentix MAP on the local host, he has to login first executing the mgx_login program using its Kerberos *principal* (for instance, bobby@MAGENTIX).

[3] http://web.mit.edu/Kerberos/

There are two different kinds of users in secure Magentix:

- **Administrator**. The administrator of a Magentix MAP. It has the following permissions:
  - Create and delete system users *principals*.
  - Create and delete MAP services *principals*.
  - Platform launching.
- **Regular Users**. Users that are allowed to launch agents in a Magentix MAP. In order to do this, the administrator has to create a *principal* per each user that needs to launch agents. At any moment the administrator can remove an agent launching right from a user by simply removing its *principal*.

3.3.2. Magentix Agents

Two kinds of agents can be part of Magentix: platform and user agents. Platform agents offer the basic services that user agents need (they are part of the AP itself) while user agents make up the MAS being run on top of the AP.

When Magentix is launched, an identity (kerberos *principal*) for each platform agent is created. For instance, if `pc.example.com` is going to be a part of Magentix, the following *principals* are created: `mgx/pc.example.com@MAGENTIX`, `ams/pc.example.com@MAGENTIX`, `df/pc.example.com@MAGENTIX` and `oum/pc.example.com@MAGENTIX`.

Agent Launching

User agents are created after the AP startup process and their identities follow the form `agentname@MAGENTIX`. The process for creating a user agent assures that: firstly, only AP users can launch user agents; secondly, resources from agents launched cannot be accessed by agents owned by a different user; and thirdly, the creator user is known and can be punished if necessary (the AMS maintains a log relating agent identities to user identities).

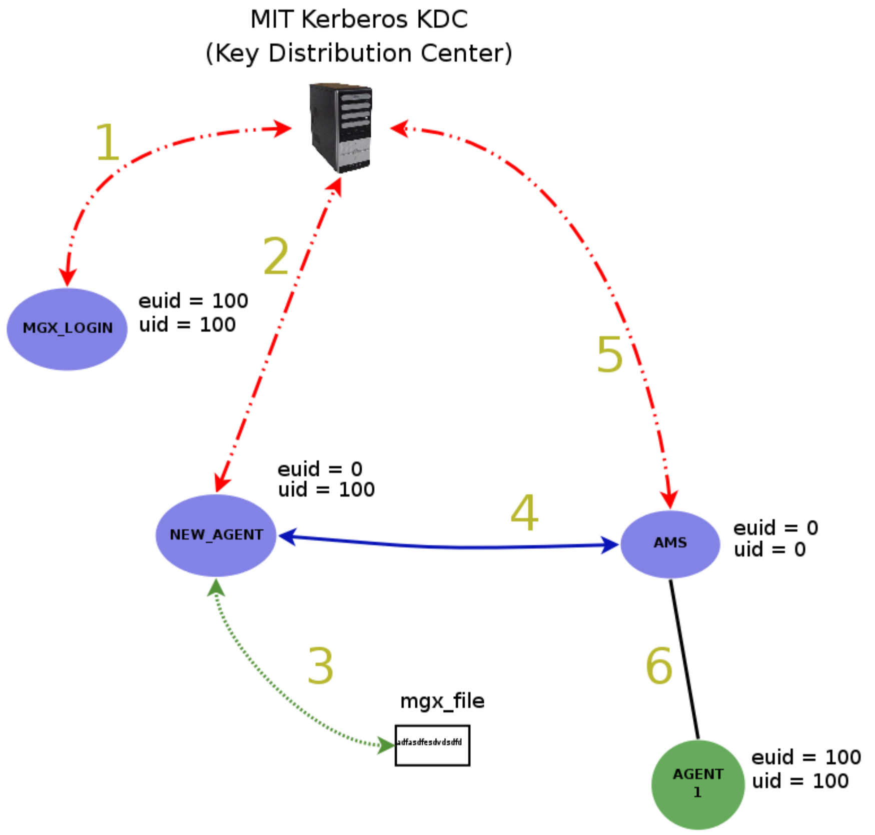

Figure 3.2: Agent Launching.

Figure 3.2 shows the process with its stages numbered. The stages are:

1. User authenticates to the KDC with the `mgx_login` program using its AP identity (Kerberos *principal*).

2. User launches `new_agent` program that has its `setuid` active and runs with effective `uid` (`euid`) as `root`. Then, `new_agent` asks the KDC for a ticket to communicate with the `ams` service using the AP identity of the user.

3. `new_agent` reads the key generated by the `ams` when the AP is launched to ensure that the `new_agent` implementation is the one expected. The file containing this key (named `mgx_file`) can only be read by `root`, that is the reason why `new_agent` has its `setuid` active.

4. A security context is created between the `ams` and `new_agent` using the AP identity of the ams and the AP identity of the user that has launched the `new_agent`. Then, `new_agent` sends the request to create a new agent. Although the request is not generated by the AP administrator, the `ams` accepts it in order to allow all AP users to launch agents. The request contains: the name of the agent to be created, the Linux `uid` and `gid` of the requesting user, the key generated by the `ams` when the AP is launched, the route of the agent binary and the arguments for the agent.

5. The `ams` asks the KDC to create an identity for the new agent by means of using the `kadmin` program of the Kerberos distribution. An agent AP identity (Kerberos *principal*) follows the form `agentname@MAGENTIX`.

6. The `ams` launches the binary of the agent setting its `uid` and `gid` to the `uid` and `gid` of the Linux user that has launched the `new_agent`. Therefore, the agent created can only access the same Linux local resources as its owner.

Finally, when an agent dies, the `ams` removes its *principal*.

Agent Communication

Regarding agent communication, both platform and user agents communicate in the same way. When an platform/user agent requires communication with another agent, a security context is established as client with the identity (kerberos *principal*) of the requesting agent and as server with the identity of the destination agent. Kerberos is told to cipher the communications so that integrity and confidentiality of all of the data exchanged between these two agents is assured. Kerberos also prevents replay attacks and attacks due to the clock. Kerberos security contexts expire (they are not valid infinitely), so an agent can discover that a security context is no longer valid when trying to encrypt or decrypt data. Then, a new security context has to be negotiated with her conversation partner.

An agent always uses its identity when interacting with other agents. In this sense, the privacy of its owner's identity is preserved. For accountability concerns, the AMS service has a log relating **Agent** identities to **User** identities in the form of $\{agent, owner\}$ tuples. This log can only be accessed by the AP administrator user. Thus, when an agent behaves wrongly, the user that is liable for the agent's behavior is known (and can be punished if necessary).

There are some scenarios where the **Agent** identity is not enough in order for two agents to interact with each other. The **User** identities of the owners of the agents interacting are needed. Agents are then allowed to disclose their **User** identity to the other interacting agent.

In order for agents to act on behalf of the agent group they represent, they are also given the **Group** identity of an agent group (which is unique and created when the group is so). Thus, an agent has access to all of the **Group** identities of the agent groups represented by herself. Agent groups in the SI are explained in more detail later on.

The Magentix SI agents manage three identity types when communicating with other agents:

- **Agent** identity. The unique identity of an agent.
- **User** identity. The identity of its owner, i.e, the identity of the user that created the agent.
- **Group** identity. The identity of each agent group of which the agent is a *contact* agent(explained later on).

A Magentix agent is provided with more than one identity, so a way to indicate to the Magentix communication module which Kerberos credentials it has to use when sending a message is needed. This is done with a new field in the message header. If this field is in the message header of a message to be sent, the communication module tries to use the identity chosen; otherwise the corresponding **Agent** identity is used. If the Kerberos credentials associated to the identity that the agent is requesting are not available, and the agent is trying to use an identity that it does not own for instance, the sending of the message fails.

Agent Confinement

We use AppArmor[4] (Bauer, 2006) in order to provide Magentix agents with a fine-grained access control and restricted privilege usage. In this sense, each Magentix agent has an AppArmor profile that restricts her behaviour in the system following the principle of least privilege. These profiles describe which files an agent can access and which capabilities an agent can have.

Platform Agents are run as `root` user because: they need some privileges; and as a way to ensure that the administrator of a computer is the only user that can manipulate them. Using AppArmor we can confine platform agents to only have the privileges that are strictly necessary. AMS needs `CAP_SETUID` and `CAP_SETGID` in order to execute user agents and change its effective uid and gid to uid and gid of the user that wants to create the agent. Both MGX and AMS platform agents can terminate platform and user agents respectively (e.g. when shutting down the platform), so `CAP_KILL` is also needed for these two agents. DF and OUM do not need any special capability. Hence, an AppArmor profile is given for each platform agent containing only the capabilities that each platform agent needs, restricting the overall damage in the event of any of them being compromised.

Figure 3.3 shows the AppArmor Profile of the MGX platform agent. When Magentix is installed in a host the platform agent binaries are copied to `/usr/lib/-magentix`. As discussed before, MGX is given with `CAP_KILL` capability. MGX also needs to read from `/etc/krb5.keytab`. Moreover, MGX can read (`r`) and execute[5]

[4] `http://www.novell.com/linux/security/apparmor/`

[5] Execute access requires a modifier: `p` instructs AppArmor to transition to a different profile when a process performs on exec() with the named program. `u` instructs AppArmor to unconfine the process when it performs an exec() with the named program. `i` instructs AppArmor to keep the current profile, even if a profile exists for the named program.

```
/usr/lib/magentix/mgx {

 capability kill,

 /etc/krb5.keytab r,
 /usr/lib/magentix/ams rpx,
 /usr/lib/magentix/df rpx,
 /usr/lib/magentix/oum rpx,

}
```

Figure 3.3: AppArmor profile of the MGX platform agent.

(`px`) AMS, DF and OUM binaries. Using this profile for the MGX platform agent, the damage to the overall system is limited in case the MGX agent is compromised. In such an event, although the attacker would act as `root` user, she could only have the privilege to kill processes, affecting the system's availability. Integrity of the data in the system is preserved because the MGX agent is not allowed to write in any file. Finally, only the confidentiality of files `/etc/krb5.keytab`, `/usr/lib/magentix/ams`, `/usr/lib/magentix/df` and `/usr/lib/magentix/oum` is compromised due to the read permission. The rest of the files in the system remain confidential.

User Agents have a default profile which restricts them to no permission in the system. Therefore, user agents with a default profile can only interact with other agents by means of the communication mechanisms explained previously. For this reason, user agents with a default profile cannot access any file or perform any action on the system except the sending and receiving of messages to and from the rest of the agents running on Magentix.

At launch time, users can provide a custom profile allowing an agent to use some of the user permissions (e.g. access to a database). However, these permissions can only be a subset of the permissions that the user already has. AppArmor does not bypass the Linux traditional access control mechanism, it only allows a more fine-grained control.

3.3.3. Magentix Agent Groups

The SI allows agents to organize themselves into groups These groups can be both static and dynamic, i.e., they can be specified before running the application and they can also emerge at runtime when an agent decides to create one. The designer of the final application can define the groups and which agents make them up. The designer can also define the behaviour of the agents, and at runtime they can group themselves in order to fulfill their design objectives.

Apart from structuring purposes, both agent to group and group to group interactions are allowed. Interactions are therefore described by defining the *contact* agents and the *routing type* of a group. Routing types are as follows:

- Unicast: The messages addressed to the unit are delivered to a single agent which is responsible for receiving messages. This type is useful when we want a single message entrance to the group. It could be useful if the group presents for example, a hierarchical structure, where the supervisor receives every message and distributed them to its subordinates.

- Multicast: Several agents can be appointed to receive messages. When a message is addressed to the unit, this message is delivered to any contact agent of the unit. It could be useful if we want to represent an anarchic scenario, where every message needs to be known by every agent without any kind of filter.

- Round Robin: There can be several agents appointed to receive messages. But each message addressed to the unit is delivered to a different contact agent, defined according to a circular policy. This type of routing messages it is useful when some agents offers the same service but we want to distribute the incomming requests to avoid the bottlenecks.

- Random: Several agents can be defined as contact agents. But the message is delivered to a single one, according to a random policy. As the previous type, it is useful for distributing the incomming requests, but it is not specified any kind of order for attending these requests.

- Sourcehash: Several agents can be the contact agents. But any given message is delivered to one of the agents responsible for receiving messages, according to the host where the message sender is situated. It is a load-balancing technique.

The SI secures agent group authentication and agent group interactions with agents and other agent groups (only agents with permission can act on behalf of an agent group). An agent cannot use its identity when it is communicating on behalf of an agent group, because in such a case the destination agent would know the identity of the former agent and no hiding would be possible. Moreover, the destination agent is able to check that the agent that has sent a message on behalf of an agent group is a member of the group. That way, the SI also takes into account the **Group** identity concept, so that all of the agent groups created in the platform have a way of proving themselves when using the support provided by the Magentix SI.

When an agent decides to create a new agent group, it makes a request to the OUM, which creates the agent group. From this moment on, the agent that created the agent group becomes the manager of the agent group. Regarding access control in agent groups, due to the way agent groups are implemented in Magentix, only the agent that is the manager of the group decides if a new agent can join the group. Moreover, this decision can be based on the identity of the agent that is requesting to become a member of the agent group, because the agent identities are assured. Moreover the manager can also decide to throw out an agent which is not acting as expected.

When the OUM creates a new agent group, it also creates a new identity (A Kerberos *principal* following the form `groupname@MAGENTIX`) for the new agent group, as the AMS creates a new agent identity when it creates a new agent. Moreover, the credentials of the new agent group are given to the manager of the agent group. From this moment on, when an agent makes a request to join the agent group as a *contact* agent, if the manager accepts the request, it also gives the **Group** identity to the *contact* agent so that the latter can act on behalf of the agent group. Agents that are *members* of a group but not *contact* cannot act on behalf of this agent group to interact with other agents and agent groups. Thus, an agent has access to all of the **Group** identities of the agent groups in which she is a *contact* agent.

The SI allows agent groups to be shaped like some well-known agent organizations (Horling and Lesser, 2004), such as teams, coalitions, hierarchies, federations, etc. This is due to the fact that the SI allows: firstly, the definition of the structure and interactions of an agent group; and secondly, it secures agent group authentication and interactions with other agents and agent groups.

3.4. Evaluation

In this section, we provide an evaluation of the SI presented. To carry out the evaluation, we developed an application on top of Magentix to assess the feasibility of developing secure MAS in Magentix. This application is a capture-the-flag game as described in section 3.4.1. In this sense, we describe the security benefits the application has when running on top of Magentix with the SI enabled.

Based on this application, we present a performance evaluation in section 3.4.2 which compares the performance of the application running on top of Magentix without the SI versus Magentix with the SI.

3.4.1. Capture-the-flag Game Application

A Capture the Flag (CTF) game (Barella et al., 2006) involves agents that are grouped into two teams (allies and axis). The allies must go to the axis base, capture the flag, and take it to their base, in order to win the game. The axis agents must defend their flag against the allies and, if the flag is captured, they must return it to their base. There is a time limit for the allies to bring the flag to their base. If the time limit expires, the axis team wins the game.

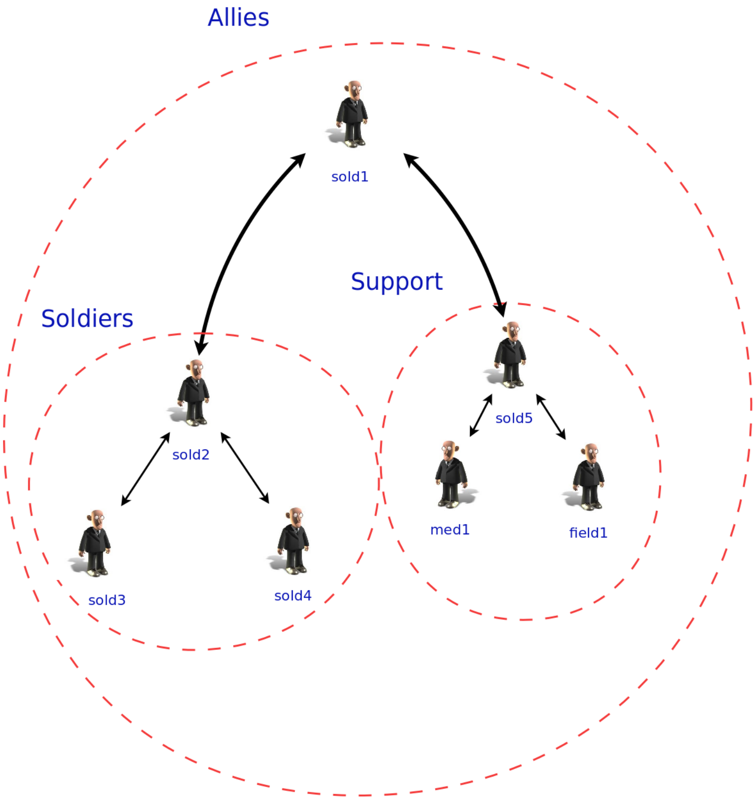

Figure 3.4: Allies team

There are three kinds of agents which provide specific services: Soldiers are the main agents in the game, Medics heal Soldiers and FieldOps provide Soldiers with munitions. In the remainder of this section, we detail an example of the CTF game in which the $Allies$ and the $Axis$ groups are composed of five Soldiers, one Medic and one FieldOps each one. We focus on the implementation of the agents using Magentix. How the agents are rendered in a Virtual 3D world, in which they move and can capture the flag, is out of the scope of this chapter. To know more about an example of how agents can be rendered in a Virtual 3D world, please refer to (Barella et al., 2006).

The $Allies$ group is defined as a two-level hierarchy shown in Figure 3.4. It is composed of the agent $sold1$ and the simple hierarchies $Soldiers$ and $Support$. The

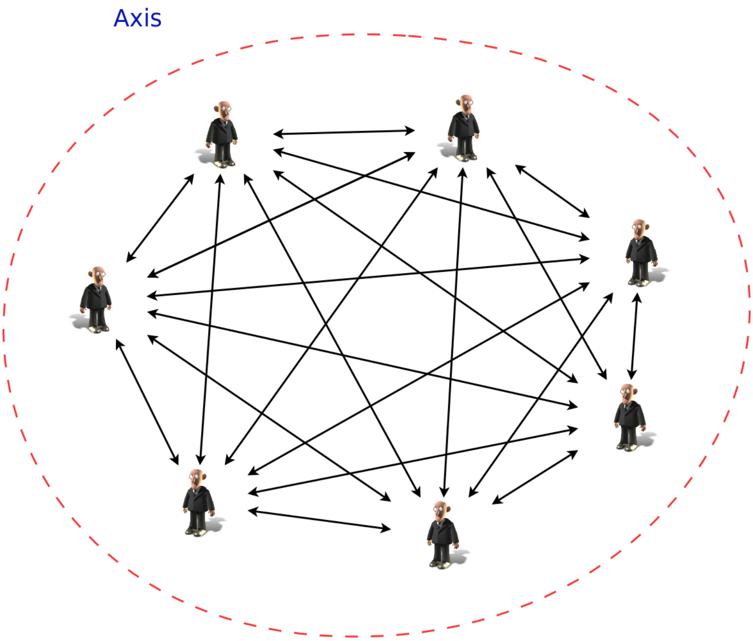

Figure 3.5: Axis team

supervisor (the only one that can act on behalf of $Allies$) is the agent $sold1$, and $Soldiers$ and $Support$ are the subordinates.

The $Soldiers$ and $Support$ groups are created as a simple hierarchies. For the group $Soldiers$ the supervisor is $sold2$ and agents $sold3$ and $sold4$ his subortinates. Therefore, at some point of the code of agent $sold2$, he has to create the corresponding Magentix $Soldiers$ group as follows:

```
mgx_new_group("Soldiers",UNICAST);
mgx_new_member("Soldiers","sold2");
mgx_new_member("Soldiers","sold3");
mgx_new_member("Soldiers","sold4");
mgx_new_contact_agent("Soldiers","sold2");
```

Note that the $Soldiers$ unit is created with the routing type `UNICAST` and $sold2$ is defined as the unique `contact_agent` so that $sold2$ is the unique agent allowed to interact on behalf of *Soldiers*. Similar code is needed in agent *sold*5 in order to create the *Support* group. Finally, the code for creating the *Allies* group is in the *sold*1 agent and is as follows:

```
mgx_new_group("Allies",UNICAST);
mgx_new_member ("Allies","sold1");
mgx_new_member ("Allies","Soldiers");
mgx_new_member ("Allies","Support");
mgx_new_contact_agent ("Allies", "sold1");
```

Axis group is defined as a team (Figure 3.5) so that all of its members are equally important and can interact with each other. In order to carry out the implementation of the *Axis* group, one of the agents that make up the group must create it. In our example, *sold*6 is in charge to create the *Axis* group as follows:

```
mgx_new_group("Axis",MULTICAST);
mgx_new_member ("Axis","sold6");
mgx_new_contact_agent ("Axis", "sold6");
                  .
                  .
                  .
mgx_new_member ("Axis","field2");
mgx_new_contact_agent ("Axis", "field2");
```

Note that *Axis* is defined with the routing type `MULTICAST`, and all of the agents of *Axis* are also defined as `contact_agents`.

All the agents are implemented using the Magentix API[6] in the same way without taking into account whether Magentix has the SI enabled or not. This is, indeed, one of the main advantages of the SI presented, it is almost transparent for agent developers. In this example, there is no extra coding needed for the agents to run in a secure fashion. Then, the Magentix administrator can choose to enable or disable the SI at Magentix's launching time. What are the advantages of enabling the SI in

[6] In oder to learn how to develop Magentix agents refer to the Magentix documentation available at `http://users.dsic.upv.es/grupos/ia/sma/tools/Magentix/documentation.html`

this CTF application? There are some advantages that we describe in the following paragraphs.

At runtime, the Magentix SI assures the message exchanges and the identities of both agents and groups. As the agent and group identities are assured, an agent can avoid cheating agents that try to act as members of the allies team but are really members of the axis team. The message exchanges are also secured, assuring their confidentiality and integrity. In this sense, the Magentix SI avoids that agents from a team (allies/axis) can overhear conversations of agents from the other team (axis/allies).

Although Magentix allows the creation and management of groups, it only can assure that groups are shaped like some well-known agent organizations (Horling and Lesser, 2004) if the SI is enabled. This is due to the fact that the SI allows: firstly, the definition of the structure and interactions of an agent group; and secondly, it secures agent group authentication and interactions with other agents and agent groups. For instance, in the $Allies$ group, the unique agent that is the supervisor is agent $sold1$. Moreover, the unique supervisor of the $Soldiers$ group, which is a subordinate of the $Allies$ group, is agent $sold2$. In order to preserve the chain of command, $sold1$ must be sure that he is talking to the supervisor of the group of subordinates $Soldiers$.

By assuring identities and message exchanges, the CTF game implemented on top of Magentix may be used as an online game with multiple players each one acting as a CTF agent (Soldier, Medic or FieldOps) avoiding cheating agents. This is a key feature for online games, because cheating agents could discredit the game, discouraging users from playing it. What is more, in this kind of games, the anonymity of the players will be preserved, because each of their agents will have an identity for their own. However, if an agent misbehaves seriously and breaks Magentix security, the identity of her owner is recorded by the AMS so that her owner could be punished by the law.

Finally, we stated that the SI is almost transparent for agent developers. In this sense, all the agents will have the default AppArmor profile, i.e., they cannot do anything more in the local machine where they are running rather than communicating with other agents. However, the agent developer may need to modify this default profile to allow agents to have access to more resources. For instance, if an agent needs to read a database where there are previously finished CTF games to learn playing better for the current game, the agent developer must create an special AppArmor profile for this agent to allow her to read the database.

3.4.2. Performance Evaluation

We present a set of tests in order to measure the differences in efficiency and scalability of both Magentix without the SI and Magentix with the SI enabled. For the sake of the clarity of the figures presented, we will denote Magentix without the SI as simply Magentix and Magentix with the SI enabled as Magentix-S. We have previous experiences in evaluating the efficiency and scalability of APs (refer to (Mulet et al., 2006)) which have helped in the design and execution of the tests.

We used 7 PCs Intel(R) Core(TM) 2 Duo CPU @ 2.60GHz, 1GB RAM, Ubuntu 8.10 and Linux Kernel 2.6.27. The computers are connected to each other via a 100Mb Ethernet hub. The SI has been configured so that Kerberos uses the AES algorithm with 128-bit keys to encrypt and SHA-1 hash function with 96-bit keys to perform HMAC computations.

We first check how Magentix and Magentix-S perform when the number of agents in the system is increased (simulating an on-line game with a big amount of players). In this sense, we launch the allies and the axis teams with the goal of exchanging a big amount of messages (a lot of interactions among agents). We increase the number of agents of each team so that both the number of agents and the messages exchanged increase. Each agent (being Medic, Soldier or FieldOp) communicates with the rest of the agents of the group in a sequential way. Each agent sends and receives m messages. Moreover, if an agent receives a message from other agent, she answer the message. We measure the time elapsed between the first message is sent by the first agent and the last message is received by the last agent. We start the experiment with 100 agents in the system (50 agents in each group) and we increase this number up to 1000. The number of messages sent by each agent is specified to $m = 1000$.

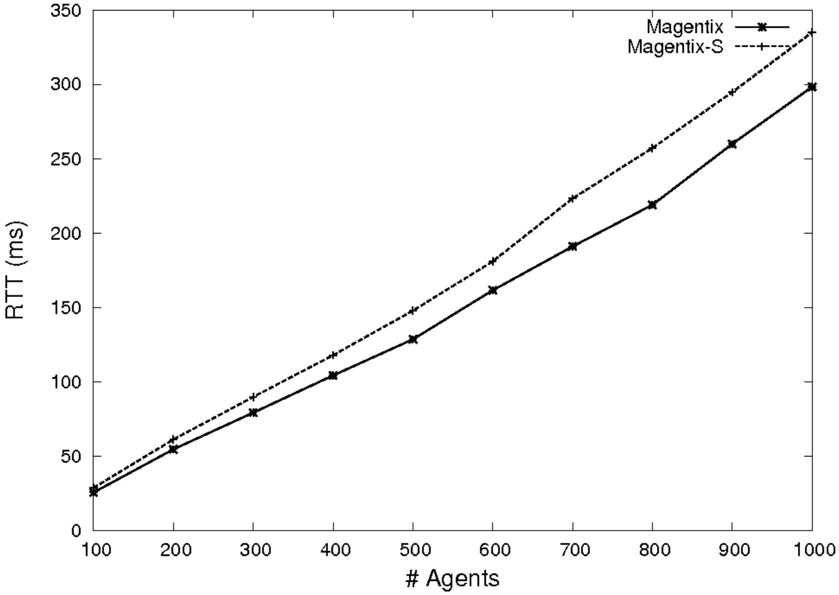

Figure 3.6: Results for Test 1

Figure 3.6 shows the results obtained. The differences between Magentix and Magentix-S are slight. However, we can see in the results that when increasing a lot the number of agents per host, the differences are not constant but directly related to the number of agents. This is due to the fact that there are a lot of agents in the same host requiring cryptographic computations, and they have to share the computation power available in each of the hosts.

Another typical scenario that may arise in a CTF game is the massive message sending to a specific agent. In this second test, we measure the ability of the frameworks when a lot of agents send messages to a single one. This specific agent could become a bottleneck in the system when multiple messages are addressed to her. This scenario appears, for example, when soldiers are requesting the same medic agent to heal them. This agent has to serve every received request. As the number of incoming requests is increased, the time for processing these requests may also increase.

In order to simulate this situation, we present a test in which a medic agent is

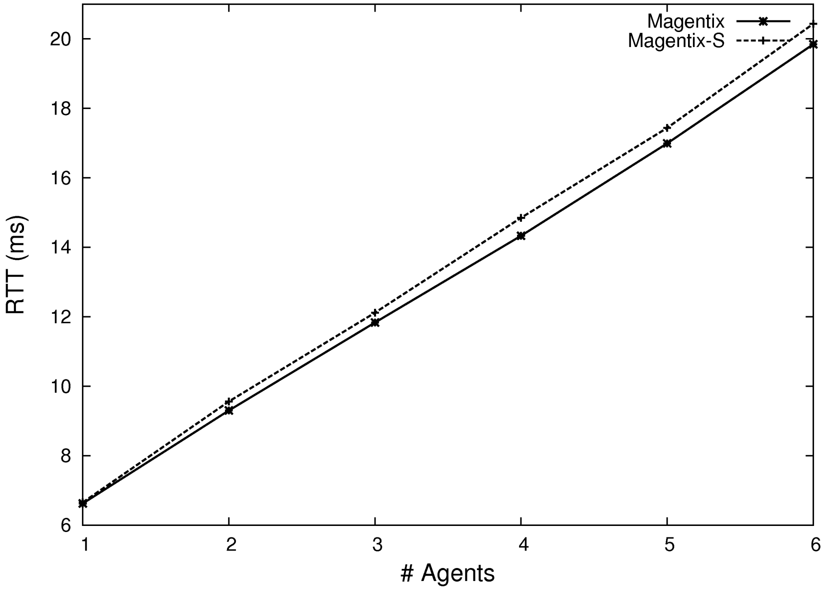

Figure 3.7: Results for Test 2

requested by a lot of agents of the same team. In this test, we launch a single medic agent and n soldier agents whose goal is to send m messages to the medic agent. The elapsed time between when the Medic agent receives the first message and when she answer the nxm messages can be seen in Figure 3.7. In this experiment we launch an allies team composed by the Medic and one Soldier agent ($n = 1$) and we increase the number of Soldier agents up to 6. Each soldier agent sends an amount of 10000 messages ($m = 10000$) to the Medic agent.

We can see that the elapsed time increases in both approaches as the number of requests increase. The differences in this test between Magentix and Magentix-S are almost negligible. As in the first test, this differences can be attributed to the needed cryptographic computations needed for each message exchange. In this sense, the differences are smaller because there are less agents per host, so that the cryptographic computations distributed among the hosts.

With the results provided in these tests, we can conclude Magentix-S performs slightly worse than Magentix. In these tests, we have simulated two typical scenarios that may arise in the CTF game. Moreover, these tests represent typical scenarios in any MAS: first, a huge number of agents that must be able to communicate to each other; and second, an agent who provides a service very requested by other agents. These tests represent critical situations so that we can see more clearly the degree of performance difference between Magentix and Magentix-S.

This little difference between the performance of Magentix and Magentix-S means that all of the security features explained at the end of the section 3.4.1 come at a bearable performance degradation. Therefore, Magentix-S allows the development and execution of secure MAS introducing only a little performance penalty with respect to non-secure MAS.

3.5. Conclusions

With the addition of the SI presented in this chapter, Magentix provides baseline security features (confidentiality, integrity, authentication and authorization) plus user identity privacy and agent group support. Although we implemented the SI in Magentix, this SI could also be generalized to be applied in other APs.

User identity privacy is achieved by providing a unique identity for each agent so that user identities are only disclosed when strictly necessary. Agents interact with each other maintaining the anonymity of their owners whenever possible. In order to ensure accountability, Magentix manages a log relating agent and user identities so that when an agent behaves wrongly her owner is known and can be punished if necessary.

The SI presented provides support for agent groups. It allows agents to organize themselves into groups. This can be viewed as a pre-support for agent organizations because it allows the definition of organizational structures (such as hierarchies, teams, etc.) by defining the group structure and its permitted interactions.

Including security features obviously makes Magentix perform worse because expensive cryptographic computations are required in order to assure integrity and confidentiality of the messages exchanged among agents. However, as shown in section 3.4, the performance degradation introduced by the SI is absolutely bearable. Therefore, the whole security features obtained when using Magentix with the SI enabled make this degradation almost negligible.

Another important feature of the SI presented is that it is almost transparent for agent developers. In this sense, the Magentix administrator can choose to enable or not the SI at Magentix's launching time without requiring changes in the agent code.

Magentix, as presented in this chapter, provides two main features for preserving privacy: (i) confidentiality in the messages two agents exchange; and (ii) hiding the user identity during agent interactions. These two features are crucial to preserve privacy, but may not be sufficient. For instance, minimizing data identifiability as pointed out in Chapter 2 may require agents to change their pseudonym (pseudonyms in this case match Magentix agent identities). Agents in Magentix cannot hold more than one identity, i.e., users should use a different agent each time they want to use a different pseudonym. Moreover, in order to avoid a lack of accountability that could cause a sense of impunity and encourage abuse, Magentix keep track of the association between users and pseudonyms (Magentix identities). The main drawback of this approach is that the Magentix itself must be trusted. This is because Magentix knows the relation of pseudonyms to the user involved. Although this is needed for ensuring accountability (users can remain liable for their agents behavior even when pseudonyms are used), this usually implies that the organization or company that hosts the specific system (e.g. eBay) knows the association of pseudonyms to users. Therefore, this organization or company can collect and process information about the users that run their agents on the system.

The next chapter (Chapter 4) proposes a model for agent identity management that considers that agents can hold more than one pseudonym, and that APs should not keep track of pseudonym-user relations while still allowing accountability. Later on in Chapter 5 a possible implementation of this model is presented.

CHAPTER 4

Privacy-Enhancing Agent Identity Management

Contents

4.1. Introduction

As pointed out in Chapter 2, information processing refers to the use or transformation of data that has been already collected (Spiekermann and Cranor, 2009). For instance, a vendor could have a complete profile of a customer containing relevant data collected from the purchases made by the customer's agent. The vendor can then use information filtering techniques to obtain detailed information on customer's tastes. Then, the vendor can infer which goods the customer is more willing to acquire and offer them in advance through personalized advertising.

The privacy risks of information processing can be alleviated by minimizing data identifiability. Spiekermann and Cranor (2009) states that "Identifiability can be defined as the degree to which (personal) data can be directly linked to an individual". The degree of privacy of a system is inversely related to the degree of user data identifiability (Pfitzmann and Hansen, 2010). The more identifiable data that exists about a person, the less she is able to control access to information about herself, and the greater the privacy risks. Identifiability ranges from complete identified to anonymous.

Pseudonymity (Chaum, 1985) can be used for minimizing data identifiability. Pseudonymity is the use of pseudonyms as identifiers. A pseudonym is an identifier of a subject other than one of the subject's real names (Pfitzmann and Hansen, 2010). The most important treat of pseudonymity is that it comprises all degrees of identifiability of a subject (from identified to anonymous) depending on the nature of the pseudonyms being used. Complete identification is when the linking between a pseudonym and its holder is publicly known. Anonymity can be achieved by using a different pseudonym for each different transaction known as transaction pseudonyms (Chaum, 1985), unless the information contained in such transactions establishes linkability (Bhargav-Spantzel et al., 2007).

Minimizing data identifiability may have a direct impact on trust and reputation. This is because current Trust and Reputation systems are based on the assumption that identities are long-lived, so that ratings about a particular entity from the past are related to the same entity in the future. However, when such systems are actually used in real domains this assumption is no longer valid. For instance, an entity which has a low reputation due to its cheating behavior may be really interested in changing her identity and restarting her reputation from scratch. This is what Jøsang et al. (2007) called the *change of identities* problem. This problem has also been identified by other researchers under different names (e.g. *whitewashing* (Carrara and Hogben, 2007)).

The work of Kerr and Cohen (2009) shows that Trust and Reputation Systems exhibit multiple vulnerabilities that can be exploited by attacks performed by cheating agents. Among these vulnerabilities, the *re-entry* vulnerability exactly matches the *change of identities* problem exposed by Jøsang et al. They propose a simple attack that takes advantage of this vulnerability: An agent opens an account (identity) in a marketplace, uses her account to cheat for a period, then abandons it to open another.

Kerr and Cohen (2009) also point out the fact that entities could create new accounts (identity in the system) at will, not only after abandoning their previous identity but also holding multiple identities at once. This is known as the *sybil* attack (Jøsang and Golbeck, 2009). An example of this attack could be an agent that holds multiple identities in a marketplace and attempts to sell the same product through each of them, increasing the probability of being chosen by a potential buyer.

It is worth mentioning that this is not an authenticity problem. Interactions among entities are assured, i.e, an agent holding an identity is sure of being able to interact with the agent that holds the other identity. However, there is nothing which could have prevented the agent behind that identity from holding another identity previously or holding multiple identities at once. For instance, let us take a buyer agent and a seller agent in an e-marketplace. The buyer has an identity in the e-marketplace under the name of *buy1* and the seller two identities in the e-marketplace *seller1* and *seller2*. Authentication in this case means that if *buy1* is interacting with *seller1* she is sure that she is interacting with who she wants. However, *buy1* has no idea that *seller1* and *seller2* are the same entity.

These vulnerabilities can be more or less harmful depending on the final domain of the application. However, these vulnerabilities should be, at least, considered in domains in which trust and reputation play a crucial role. For instance, in e-marketplaces these vulnerabilities can cause users being seriously damaged by losing money. Another example can be a social network like Last.fm[1] in which users can recommend music to each other. A user who always fails to recommend good music to other users may gain a very bad reputation. If this user creates a new account in Last.fm (a new identity in Last.fm) her reputation starts from scratch, and she is able to keep on recommending bad music. Users may be really bothered with such recommendations and move to other social networks. In this case, the one seriously damaged is the social network itself by losing users.

In this chapter, we propose an identity management model for agents in a Multi-agent System. This model enhances privacy by allowing agents to hold as many identities as needed for minimizing data identifiability, i.e., the degree by which personal information can be directly attributed to a particular principal. Privacy is enhanced without compromising trust and reputation. To this aim, the model proposes a solution for the well-known identity-related vulnerabilities of trust and reputation models. Otherwise, these vulnerabilities can be exploited through whitewashing and sibyl attacks.

This model also considers that agents should be able to selectively disclose parts (attributes) of their identity. As detailed later on in this chapter, agent identities are usually composed of not only pseudonyms but also other attributes describing the agent in a given context (e.g. roles, location, etc.). To enhance privacy, agents should be able to disclose these attributes at will, i.e., they should have control about which attributes are disclosed to which other agents.

The reminder of the chapter is organized as follows. We introduce in the next section the concept of partial identity and relate this concept to trust and reputation later on in sections 4.3 and 4.4. In section 4.5 we introduce what we call the *Partial Identity Unlinkability* Problem (PIUP), which is a generalization of the whitewashing and sybil attacks. A privacy-preserving solution to PIUP is proposed in section 4.6, taking into consideration partial identities and their relation to privacy, trust and reputation. Section 4.8 presents an implementation of a prototype of our solution to PIUP and an application scenario. Finally, section 4.9 presents some concluding remarks.

[1] `http://www.last.fm`

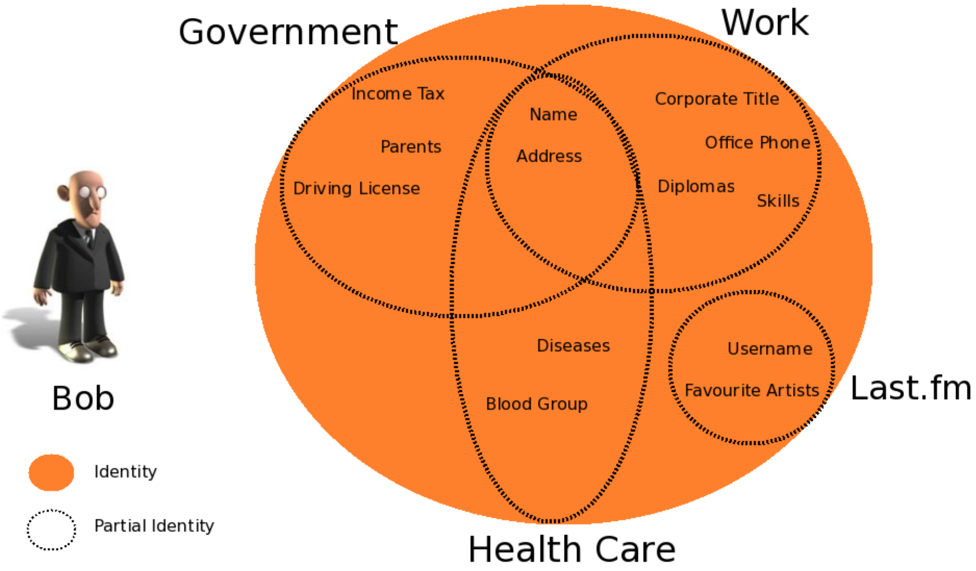

Figure 4.1: Identity and Partial Identities of Bob

4.2. Identity and Partial Identities

The identity and partial identity terms are broadly used in identity management literature, such as Clauβ et al. (2005), Pfitzmann and Hansen (2010) and Rannenberg et al. (2009). However, there is a lack of clear and formal definitions of these two terms. In this section, we propose formal definitions of both identity and partial identity.

We assume that an entity can be: a legal person (a human being, a company, etc.) or a software entity (an intelligent agent, a virtual organization, etc.).

We also assume that entities are described by attributes attached to them. Attributes can describe a great range of topics (Rannenberg et al., 2009). For instance, entity names, biological characteristics (only for human beings), location (permanent address, geo-location at a given time), competences (diploma, skills), social characteristics (affiliation to groups, friends), and even behaviors (personality or mood).

Definition 1 *Given a finite set of attributes* $A = \{a_1, \ldots, a_n\}$ *each one with a finite* *domain* $V_{a_i} = \{v_1, \ldots, v_k\}$*, a set of entities* E *and the entity* $e \in E$*, a* partial identity *of the entity* e *is a vector* $I_e = (i_1, \ldots, i_n)$*, satisfying* $i_j \in V_{a_j}$ *and* $\forall d[d \in E \setminus \{e\} \rightarrow \forall I_d(I_d \neq I_e)]$.

The set of attributes A, the set of values for each attribute a denoted as V_a and the set of entities E are context-dependent. Therefore, a partial identity I_e of an entity $e \in E$ sufficiently identifies (represented by the second constraint in the definition) the entity e within the set E considering A and V_a. For instance, let a human being be registered with a given profile in the Last.fm social network. This profile is a partial identity because it does sufficiently identify the human being among all of the different entities registered in Last.fm.

Although each partial identity usually identifies the entity in a specific context or role, the same partial identity can identify the entity in *different contexts*. For instance, a driver license identifies an entity in the context of operating a motorized vehicle but it also identifies an entity in the context of accessing a disco only for adults.

A partial identity is usually composed of a pseudonym that is unique within a context and other attributes that describe the entity within that context (roles, location, preferences, etc.).

Definition 2 *The* identity *of an entity* e *is* $I_e = \bigcup_j I_e^j$.

The identity I_e of an entity e is the union of all of the partial identities I_e^j of e. In this sense, an identity of an entity is composed of many partial identities. In order for the reader to better understand the identity and partial identity concepts, Figure 4.1 shows the identity and some of the partial identities of an individual person called Bob. Four partial identities are shown regarding four contexts: government, work, health care and social networking (Last.fm). For the sake of clarity, we only show some attributes that make up each of the partial identities represented. It is easily observed that the name and address of Bob are shared by three partial identities but are not used in the partial identity he uses in Last.fm.

4.2.1. Real Identities

We also consider an special type of partial identities: real identities. A real identity is a partial identity that sufficiently identifies an entity within the set of all of the *legal persons* — entities that can be liable for their acts in front of the law, such as human beings, companies, etc. As described later on in section 4.6, we use real identities for accountability concerns such as law enforcement. For this reason, real identities are restricted to only legal persons. A real identity would be for example: *Bob Andrew Miller, born in Los Angeles, CA, USA on July 7, 1975*.

Software entities (intelligent agents, virtual organizations, etc.) cannot have real identities because, up to now, they cannot be liable for their acts in front of the law. However, this may change in the future if they finally achieve some kind of legal personhood, as suggested by Chopra and White (2004), and Balke and Eymann (2008). In this sense, they may be part of the set of all of the legal persons and will have a real identity.

4.3. Trusting Entities through Partial Identities

In this section, we propose partial identities as a foundation to build trust relationships. In this sense, we first introduce the concept of trust.

According to Gambetta (1990), trust is ”*the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends*“.

Most of the trust models proposed by the agent community are based on Gambetta’s definition and treat trust as a probability. Different grounding theories are used to build these models. Although most of them are based on Game Theory (for a survey refer to (Sabater and Sierra, 2005)) there are other probabilistic approaches like (Sierra and Debenham, 2005), in which Sierra and Debenham use Information Theory.

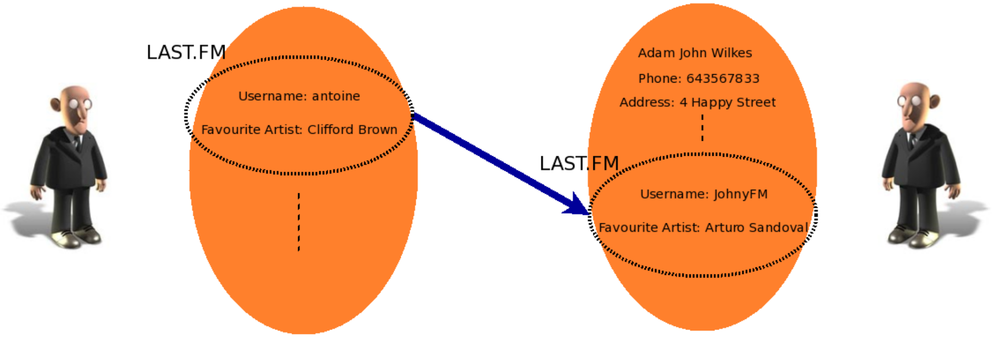

Figure 4.2: Trust Through Partial Identities

Agent community has also developed *cognitive* models which treat trust differently. For instance, Castelfranchi and Falcone (1998) define trust as "*a mental state, a complex attitude of an agent x towards another agent y about the behaviour/action relevant for the result (goal) g*".

Both probabilistic and cognitive models share that trust is established from a *trustor* (the one who trusts) to a *trustee* (the one who is trusted). Thus, we focus on trust as a directed relationship between two entities. In this sense, a primary requirement is that the trustor is able to recognize the trustee when they interact with each other.

In the real world, an individual can recognize other individuals by means of identity documents such as a passport. However, inter-personal meetings are also carried out without the needing for such documents. For instance, a trustor is able to recognize a trustee from past interactions by recognizing her face.

In the digital world there is no physical contact, all of the interactions between entities are carried out through online networks and most of them across the Internet. The increase in global connectivity increases the number of entities taking part in the digital world and also the number of interactions they carry out. In this scenario, recognizing an entity in an interaction usually means authenticating it using technologies like Kerberos (as detailed in Chapter 3), OpenID[2], and others. Entities are authenti-

[2] http://openid.net/

cated using such technologies according to a partial identity that they hold.

We propose trust relationships to be established between two entities through some of their partial identities. Moreover, these partial identities represent part of the context where the trust relationship is established.

Partial identities are key parts in order to build trust relationships. There are attributes of a partial identity of an entity that clearly describe important features of an entity. For instance, a corporate title (such as chief executive officer) is an attribute which is part of the partial identity of an employee of a company. When this employee interacts with other entities in a business context, his corporate title is an important attribute that the rest of the entities in that context will consider valuable to trust in him.

Figure 4.2 shows an example of a trust relationship established between two entities through partial identities. The entity with the username antoine trusts (represented as a directed arrow) the entity with the username JohnyFM (Adam John Wilkes). This trust relationship is contextualized in Last.fm. Moreover the favorite artist of both partial identities plays a crucial role in the trust relationship. In this sense, JohnyFM has as favorite artist Arturo Sandoval and antoine has Clifford Brown as his favorite artist. Both Arturo Sandoval and Clifford Brown are trumpet players. By knowing this, antoine may consider music recommendations from JohnyFM to be relevant for him, because they like the same kind of music players.

4.4. Reputation through Partial Identities

In the previous section, we stated how trust relationships can be built through partial identities. In this section, we state how partial identities relate to reputation.

We understand reputation in the same way as Sabater et al. in their Repage Model (Sabater-Mir et al., 2006). In this sense, reputation is a social evaluation of a target entity attitude towards socially desirable behavior which circulates in the society (and can be agreed on or not by each one of the entities in the society).

Reputation, just like trust, is known to be context dependent (Sabater and Sierra, 2005). For instance, a lawyer can have a great reputation defending digital criminals while having a bad reputation making cakes.

Unlike trust, reputation also relates to anonymity. The anonymity concept is defined by Pfitzmann and Hansen (2010) as: *”Anonymity of a subject means that the subject is not identifiable within a set of subjects“*. Reputation, as a social evaluation circulating in the society, is *anonymously* assigned to an entity. Therefore, the social evaluation any entity has about other entities remains private (whenever she does not communicate her social evaluation to others in a non-anonymous fashion).

The anonymous nature of reputation is sometimes not taken into account, which leads to some problems. For instance, the eBay reputation system is not anonymous which leads to an average 99% of positive ratings (Resnick and Zeckhauser, 2002). This is due to the fact that entities in eBay do not negatively rate other entities for fear of retaliations which could damage their own reputation and welfare.

We consider reputation as an anonymous social evaluation of an entity in a given context through one of its partial identities. In this sense, the partial identity of the entity reputed is needed to define the context of a reputation. Moreover, if an entity has a reputation in a given context, all of the entities interacting with this entity in the same context can be aware of her reputation through her partial identity.

4.5. The Partial Identity Unlinkability Problem

After the definition of the partial identity concept and its relationships to trust and reputation has been given, we are now in a position to define what we call the *partial identity unlinkability problem* (PIUP).

In section 4.1 we described two vulnerabilities that affect trust and reputation systems: the multiple identities and the change of identities problems. As far as we are concerned, these two vulnerabilities are closely related to the *unlinkability* concept described by Pfitzmann and Hansen (2010). They define *unlinkability* as ”*Unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an* *attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not*".

We use this definition of *unlinkability* made by Pfitzmann and Hansen and our definition of partial identity to formulate the PIUP:

Definition 3 *The* partial identity unlinkability *problem (PIUP) states the impossibility that an entity, which takes part in a system, is able to sufficiently distinguish whether two partial identities in that system are related or not.*

It is easily observed that the *change of identities* problem is an instantiation of PIUP, i.e., an entity with an identity by which she is known to have a bad reputation, acquires another identity with a fresh new reputation so that other entities are unable to relate the entity to its former reputation. In a similar way, if an entity does not trust another entity, the latter can change her identity. Therefore, the former entity is unable to notice that the same entity which he used to trust (distrust) is behind the new identity, so the trust relationship is restarted.

Regarding multiple identities, a similar instantiation can be made, so that an entity holds several identities and has different reputations with each of them. Thus, another entity is unable to relate the different reputations that the entity has because it is unaware of all of the identities the entity has. PIUP relates to trust in the same way when multiple identities are considered. An entity can believe that she trusts multiple entities in a given system (such as a specific marketplace), but she may be trusting the same entity with different identities without being aware of it.

4.5.1. The Straightforward Solution

PIUP is obviously solved by forcing the entities taking part in a system to use their real identity. Historically, a real identity has been used to uniquely identify persons (Rannenberg et al., 2009).

If an entity is not allowed to change its identity, then trust and reputation assessments of this identity cannot be removed. Although the changing of real identities has always been possible as a way of erasing reputation, these changes are not cost-free and do not completely erase the reputation. For instance, there are some companies that change their name in order to erase their previous reputation. However, a link with the previous reputation can be made (e.g. looking at its employees in order to find employees of the former company).

Due to the impossibility of completely erasing reputation, new online services are emerging related to the management of the online reputation of an entity with a real identity. For instance, ReputationDefender[3] and Mamba IQ[4] provide services to report the online reputation of an entity with a real identity (individuals or companies). These services usually find information related to an entity searching in blogs, social networks, and audio and video pages. These services also give the entities advice on improving their online reputation.

However, the solution of forcing entities to use their real world identities exposes a great disadvantage: privacy loss. Fischer-Hübner and Hedbom (2008) define *privacy* as "*the right to informational self-determination, i.e. the right of individuals to determine for themselves when, how, to what extent and for what purposes information about them is communicated to others*“.

Nowadays, in the era of global connectivity (everything is inter-connected anytime and everywhere) privacy is a great concern regarding identity management in the digital world. While in the real world everyone decides (at least implicitly) what to tell other people about themselves (after considering the situational context and the role each person plays), in the digital world users have more or less lost effective control over their personal data. Users are therefore exposed to constant personal data collection and processing without being aware of it (Fischer-Hübner and Hedbom, 2008).

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

[4] `http://www.mambaiq.com`

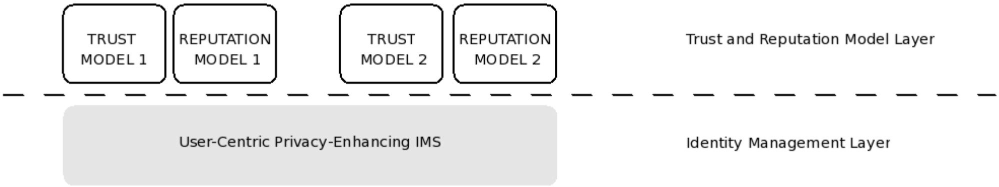
TRUST MODEL 1
REPUTATION MODEL 1
TRUST MODEL 2
REPUTATION MODEL 2
Trust and Reputation Model Layer
User-Centric Privacy-Enhancing IMS
Identity Management Layer

Figure 4.3: Two-layer architecture for Trust and Reputation without PIUP

4.6. A Privacy Preserving Solution for PIUP

After the definition of PIUP and the privacy issues of the straightforward solution, we provide a privacy preserving solution to PIUP so that trust and reputation systems can be used without PIUP and preserving users' privacy. Figure 4.3 shows our proposed architecture for trust and reputation systems. There are two layers that make up the architecture: the identity management layer and the trust and reputation model layer. The identity management layer is in charge of providing the entities taking part in a trust and reputation system with partial identity management. The trust and reputation model layer is in charge of providing the actual trust and reputation models being deployed in the system.

We assume that entities communicate to each other following a secure connection by using technologies such as Kerberos (as detailed in Chapter 3) or TLS (Dierks and Allen, 1999). Therefore, the data they exchange in their interactions is provided with basic security features such as integrity and confidentiality.

4.6.1. Identity Management Layer

The technical systems supporting the process of management of partial identities are known as Identity Management Systems (IMSs) (Rannenberg et al., 2009). User-centric privacy-enhancing IMSs are supposed to enable a user to control the nature and amount of personal information disclosed (Clauβ et al., 2005). These infrastructures are usually composed of three main parts:

- *Identity Service (IdS)* is composed of two kinds of services: Identity Providers (IdPs), that issue partial identities and validate these identities to the RPs; and Relying Parties (RPs), that are a set of APIs that allows services to check the identity of the entities that interact with them.

- The *Identity Selector (IS)* provides a simple way to manage partial identities and choose which partial identity to be used in a given context.

- *Attribute Service (AS)* include services that allow an entity to determine the access control rights of every other entity when accessing each attribute of each partial identity she holds. Attributes can be managed and self-issued. Managed attributes are verified by IdPs and are reliable (and provable) information about an entity. Self-issued attributes contain information about what an entity claims about itself. IdPs can only verify that self-issued attributes are what the entities claim about themselves.

Our solution to PIUP is based on *once-in-a-lifetime* partial identities (Friedman and Resnick, 1998). We propose that IdPs issue two kinds of partial identities: permanent partial identities (PPIs) and regular partial identities (RPIs). Entities can only hold one PPI in a system. RPIs do not pose any limitation. Although both kinds of partial identities enable trust and reputation relationships, only PPIs guarantee that PIUP is avoided. Then, entities will choose to establish trust and reputation through PPIs if they want to avoid PIUP. Our proposed identity management layer considers three main parties:

PIdP. The Permanent Identity Provider is an IdP (or a federation of IdPs[5]) that issues PPIs to the entities taking part in the specific system. Entities must register using a real identity which the PIdP will not reveal to others. The PIdP is also in charge of forcing one entity to only hold a PPI in this specific system.

[5] IMSs support the federation of IdPs that belong to the same and also different remote security domains across the Internet. A PIdP, then, can be implemented as a federation of IdPs instead of only one IdP, minimizing the typical drawbacks of a centralized trusted third party, such as being a single point of failure (SPOF) and a possible efficiency bottleneck. Examples of identity federation standards are the Liberty Alliance Identity Federation Framework `http://projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications/` and WS-Federation `http://www.ibm.com/developerworks/library/specification/ws-fed/`.

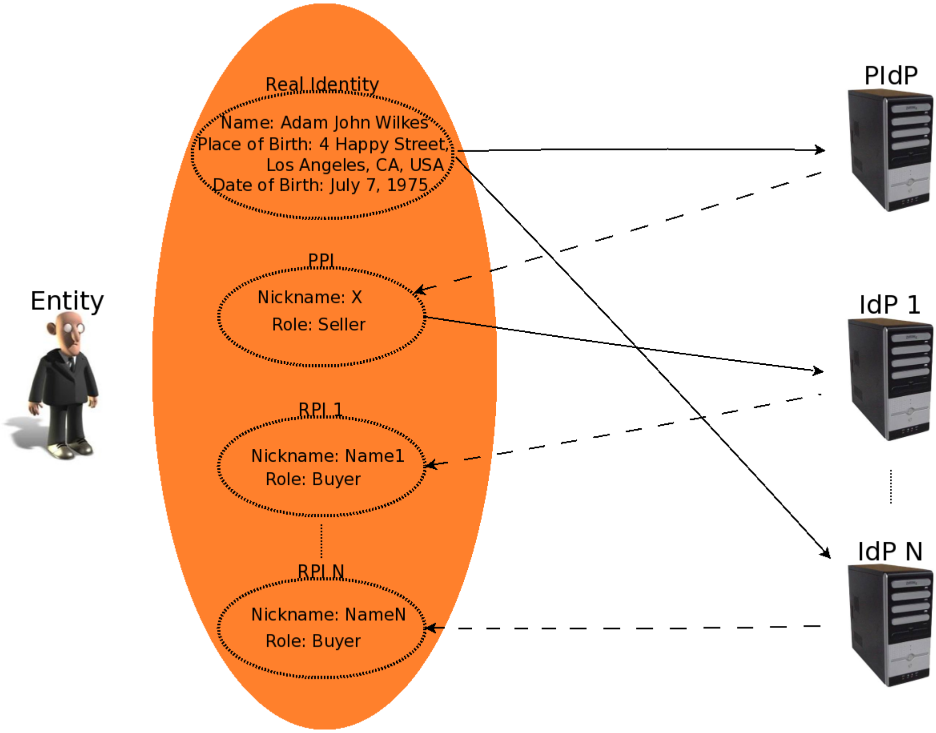

Figure 4.4: An example of an Entity as seen by the Identity Management Layer

IdPs. IdPs issue RPIs to the entities taking part in the specific system. Entities request RPIs providing either a real identity, or a PPI that IdPs will not reveal to others. There is no limitation in the number of IdPs per system as well as in the number of RPIs per entity and per system.

Entities. Entities, which are in a given trust and reputation system, select and manage their own partial identities using the IS. Moreover, entities also act as RPs that validate the partial identities of other entities through the PIdP and the IdPs. Entities use the AS to access attributes of other entities' partial identities. Entities also use the AS to set access control policies to their own partial identity attributes.

Figure 4.4 shows an example of an entity and its partial identities for a given system. The entity has the real identity with an attribute name *Adam John Wilkes*. Using this real identity the entity has obtained a PPI from the PIdP that includes two attributes: name and role. This entity has also obtained N RPIs from N different IdPs. Some of the RPIs are obtained providing its PPI (such as RPI 1) and some other using its real identity (such as RPI N).

The identity management layer provides the following main features from the point of view of security and privacy:

- *Authentication of Partial Identities*. Entities use RP APIs in order to authenticate the partial identities of the other entities taking part in the trust and reputation system. Therefore, entities are allowed to recognize to each other from interaction to interaction and establish trust and reputation relationships.
- *PIUP avoidance*. Only the PIdP is allowed to issue PPIs for a given trust and reputation system. The PIdP avoids that a previously registered entity (using a real identity) is able to obtain a new PPI. There is no chance for an entity in a trust and reputation system to have two different PPIs. Therefore, trust and reputation relationships built through PPIs avoid PIUP.
- *Multiple RPIs*. Entities can hold multiple RPIs in a system. There are many situations in which entities could be interested in using multiple RPIs. For instance, multiple RPIs can play a crucial role for preserving privacy. In order to avoid buyer profiling, entities could use a different RPI for each interaction with another entity (Warnier and Brazier, 2010).
- *Hiding of original partial identities*. IdPs (including PIdP) act as independent third parties that must be trusted by the entities taking part in the trust and reputation system. For obtaining new partial identities (PPIs or RPIs), entities must provide a real identity, or a PPI to IdPs. IdPs do not make the original partial identities available. Therefore, the rest of the entities in the trust and reputation system are, a priori[6], not able to link a partial identity used in the system to the corresponding original real identity, PPI, or RPI.
- *Entity control over partial identity attributes*. ASs allow entities to determine the access control rights over each attribute of a partial identity they hold. Entities

[6] Note that if the attributes between two partial identities of the same entity are similar enough, another entity could infer that these partial identities correspond to the same entity.

are able to choose to hide some of the attributes of a partial identity in a system as long as the resulting set of attributes is still a partial identity, i.e., it sufficiently identifies the entity among the set of entities in that system.

- *Entity accountability*. Under special circumstances, such as law enforcement, the real identity of a misbehaving entity can be known. If an entity misbehaves when using its PPI, the PIdP can disclose its real identity if required by a court. If an entity misbehaves when using one of its RPIs, IdPs can disclose the real identity or the PPI that the entity used to obtain a RPI. In case the entity used a PPI to obtain such RPI, then the PIdP can use this PPI to finally disclose the real identity of the entity. Therefore, accountability is assured and entities can be punished if necessary. This leads entities to be liable for their acts and they will take this into consideration before misbehaving.

4.6.2. Trust and Reputation Model Layer

On the top of the identity management layer, we find the trust and reputation model layer. This layer is the one which implements the actual trust and reputation models being used in the system.

Trust and reputation models in this layer are based on the definitions of identity and partial identity and their relationship with trust and reputation detailed in sections 4.2, 4.3 and 4.4. In this sense, partial identities act as a foundation for the establishment of trust and reputation among the entities taking part in the system.

The concept of partial identity is totally independent from the trust and reputation model being used. Therefore, a privacy preserving solution to PIUP is provided without the needing of re-designing the trust and reputation models. However, as explained in sections 4.3 and 4.4, partial identities are part of the context in which trust and reputation take place. Therefore, trust and reputation models must be aware of partial identities in order to extract the information they need to compute trust and reputation.

In this sense, partial identities can be used by trust and reputation systems for identifying an entity from interaction to interaction and building trust based on past interactions with her. For instance, Urbano et al. propose the SinAlpha (Urbano et al., 2009) model for trust. This model is based on past experiences (successful or not) which are converted into a measure of trust in [0,1]. 0 means no trust and 1 means completely trust. They recognize entities from interaction to interaction by using the name of each entity. Therefore, the only adaptation needed by this model is to use partial identities as sets of only one attribute: the name of each entity.

Another example of a trust and reputation model which can be built using our two-layer architecture is Fire developed by Huynh et al. (2006). This model takes into account not only past experiences but also other sources of information to assess trust and reputation. Concretely, Fire uses the role that an entity is playing in an institutional structure as a mechanism to assign default reputation to the entities. In this sense, the role of the entities can be extracted from their partial identity (whenever entities decide to make it accessible to other entities).

Finally, the trust and reputation model layer also allows heterogeneous trust and reputation systems. In this sense, there is nothing that prevents different entities from using different trust and reputation models in the same trust and reputation system. Entities are not forced to use a concrete particular trust and reputation model in a system. They could choose the trust and reputation model they prefer for a given system. Indeed, this fact opens the possibility of having multiple vendors of trust and reputation models to be used for different entities in the same system.

4.7. Prototype Implementation

We implemented one PIdP and one IdP both as webapps running on top of the Tomcat[7] web application server. Both are developed using Axis2[8]. PIdP and IdP are implemented as secure web services using the API provided by the Axis2 security

[7] `http://tomcat.apache.org/`

[8] `http://ws.apache.org/axis2/`

module Rampart[9]. Rampart complies with the OASIS WS-Security[10] and WS-Trust[11] standards.

We considered two kinds of entities: legal persons and agents. In this way, we implemented agents as simple Java objects that interact to each other by sending and receiving messages using object method calls inside the same JVM. These agents act on behalf of legal persons. Agents also use Axis2 and Rampart APIs to call the services offered by the PIdP and the IdP. These services (following the WS-Trust standard) include the issuance, renewal, cancellation and validation of partial identities in the form of SAML2[12] security tokens. Entities can, then, use these SAML2 security tokens to prove their partial identities to other agents. Moreover, agents can choose which attributes of the attributes in a partial identity to include in each security token. Thus, they have the control over what attributes are disclosed to what other agents.

An agent calls the services of the PIdP using WS-Security with X.509 certificates to obtain a PPI. These X.509 certificates contains the real identity of the legal person that an agent is acting on behalf of. We considered the set of legal persons in Spain. Thus, the PIdP requires X.509 certificates issued by either the Spanish Electronic Identification[13] (DNIe) or the *Fábrica Nacional de Moneda y Timbre*[14] (FNMT).

The PPIs issued by the PIdP can contain attributes that an agent chooses for itself (self-issued) or can contain attributes from the real identity (managed) of the legal person the agent is acting on behalf of. The important point is that once the PIdP issues a PPI, the PIdP keeps track of what real identity holds what PPI and will always issue the same PPI to the same real identity in a given system. Thus, the PIdP avoids that an agent can have more than one PPI in a given system. PPIs can also contain attributes that the PIdP verified considering the real identities behind them. For instance, an entity can be willing to include an attribute in its PPI stating that it is over 18 years old. Then, the PIdP verifies it against the birth date in the X.509

[9] http://ws.apache.org/rampart/

[10] http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[11] http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html

[12] http://saml.xml.org/saml-specifications

[13] http://www.dnielectronico.es/

[14] http://www.fnmt.es/

certificate, and if it is true the PIdP includes the attribute in the PPI issued. Afterwards, the entity is able to prove to other entities that it is over 18 years old without disclosing its birth date.

Entities call the services of the IdP using WS-Security with SAML2 tokens representing its PPI to obtain a RPI. After that, the entity is able to prove that it holds the RPI to other entities. Agents can obtain as many as RPIs they desire. The IdP only keeps track of which PPI is associated to which RPIs for accountability concerns in case of law enforcement.

4.8. Evaluation

An application Scenario for our proposed solution to PIUP is an agent-mediated e-commerce (Sierra, 2004) application. Agent-mediated electronic commerce refers to electronic commerce in which agent technologies are applied to provide personalized, continuously running, semi-autonomous behavior. In agent-mediated electronic commerce applications security, privacy, trust, and reputation play a crucial role (Fasli, 2007b).

We describe an electronic market where seller agents and buyer agents trade wines. This is based on the experimental setting of the approach described in (Aydoğan and Yolum, 2010) that provides a method for seller agents to learn buyer agents' preferences from previous negotiations with the same buyer.

In the following sections we describe the evaluation that we carried out of our proposal based on this scenario. Specifically, we show the experiments we performed and the results we obtained with respect to: to what extent changing RPIs can reduce information collection, and to what extent agents can build trust and reputation models without PIUP.

We assume that in this scenario payments are carried out using some kind of anonymous payment mechanism and that deliveries are carried out using some anonymous delivery system. Hence, credit card numbers and delivery addresses do

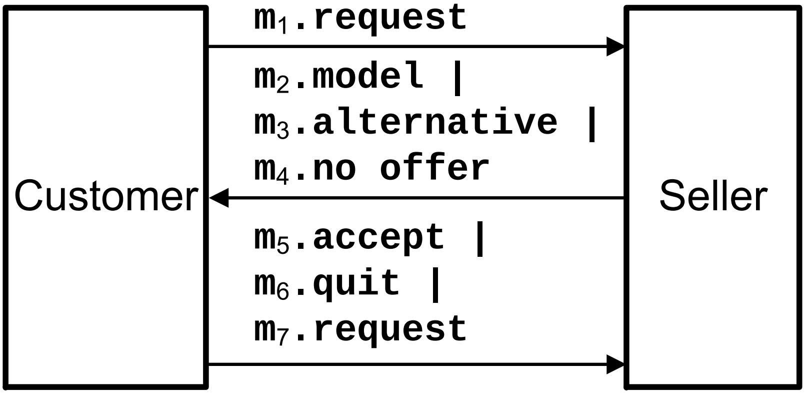

Figure 4.5: Negotiation Protocol for the Wine e-marketplace scenario.

not need to be disclosed when an agent acquires a product. For instance, the untraceable electronic cash presented by Chaum et al. Chaum et al. (1990) can be used for anonymous payments. For anonymous deliveries, the privacy-preserving physical delivery system presented by Aïmeur et al. Aïmeur et al. (2006) can be used.

4.8.1. Avoiding Information Processing

Privacy can be a great concern in this scenario. This is because buyer agents are subject to possible information processing. Specifically, seller agents could perform what is commonly known as buyer profiling (Hildebrandt and Gutwirth, 2008), in which seller agents obtain detailed profiles of their customers and tailor their offers regarding customer's tastes. Seller agents could even charge buyer agents different prices for the same wine according to the customers' profiles (Odlyzko, 2003), i.e., if a vendor knows that some wine is of great interest to one customer, the vendor could charge this customer more money for this wine than other customers for the same wine. An example of price discrimination occurred in 2000, Amazon charged customers different prices for the same DVD titles (Spiekermann, 2006).

Our aim in this section is to experimentally demonstrate that information processing, and thus, its possible undesired effects, can be minimized by changing RPIs. To this aim, we designed an experiment in which buyer agents negotiate/purchase wines with/from a seller agent. The primary objective is for buyer agents to avoid that the seller agent is capable of obtaining a preference model from them.

Attribute	**Values**
Color	red, rose, white
Body	light, medium, full
Flavor	delicate, moderate, strong
Sugar	dry, offDry, sweet
Country	France, Portugal, Spain, Italy, USA, Germany, Australia, NewZeeland

Table 4.1: Considered Wine Attributes.

Parameter	**Description**	**Value**
Ni	# of interactions per negotiation	10
Nn	# of negotiations	100
Nre	# repetitions of the experiment	100
Nbu	# of buyer agents	10
Nse	# of seller agents	1
Alg	Learning algorithms used	J48, NNge, BayesNet

Table 4.2: Parameters used in the privacy preservation experiments.

We assume that seller agents follow an approach to build buyer agents' preference models similar to Serrano et al. (2011). Agents in the e-marketplace follow the negotiation protocol depicted in Figure 4.5. A buyer agent makes a request to buy a bottle of wine with a *request* message. This message can be replied by the seller agent with either a *model* message (which means that the requested wine is available and includes its price) or an *alternative* message (which means that the requested wine is not available but there is another one that is very similar). Then, the buyer agent can reply to both messages with: an *accept* message (which means that the buyer agent accepts the wine offered), a *quit* message (which means that the negotiation was broken by the buyer agent), or a *request* message (which means that the agent request a new different bottle of wine).

We also assume the following attributes to describe wines: color, body, flavor, sugar, and country. The possible values for each of these attributes are shown in Table 4.1. We based on the wine attributes considered in the preference modeling approach described in (Aydoğan and Yolum, 2010). However, the main difference with this work is that we do not consider any ontological relation between attribute values for the sake of simplicity.

For our experiment, we consider 10 buyer agents and 1 seller agent with the parameters that we sum up in Table 4.2. Each buyer agent has different preferences with respect to the wines that it likes. The specific preferences for each agent are shown in Table 4.3. According to that preferences, each buyer agent performs 100 different purchases of a bottle of wine. Each purchase involves a negotiation with a seller agent to get the desired wine. We assume that negotiations are always successful. However, we consider that negotiations can randomly involve from 1 up to 10 rounds of the protocol. That is, we simulate negotiations on which a buyer agent and a seller agent perform a maximum number of 10 rounds of the protocol. Based on this, a seller agent marks wines that are not accepted by a buyer agent as a negative instance (class ”-”), while a seller agent marks the wine of the last step in the protocol (i.e., the wine that the buyer agent accepts to buy) as a positive instance (class ”+”). After the 100 purchases, the seller agent use all the collected instances about a buyer agent to train a classifier. Thus, the resulting trained classifier models the buyer agent's preferences with a given accuracy, which we calculate as the percent of correctly classified instances from an extra set of test instances (positive and negative) that we generate according to the buyer preferences.

The aim of the experiment is to demonstrate that changing RPIs can significantly reduce the accuracy of the preference models obtained by the seller agent. Therefore, seller agents cannot take advantage of these models to abuse buyer agents, e.g., performing price discrimination on buyer agents. In order to prove that changing RPIs can reduce information processing, we performed our experiment in which buyer agents repeat the 100 negotiations with a varying number of RPI changes. That is, buyer agents start with 100 negotiations and without any RPI change and end up using a different RPI for each of the 100 negotiations. For each number of RPI changes we

Agent	Preferences
1	$(body = light \wedge flavor = delicate) \vee$ $(sugar = dry \wedge country = Portugal)$
2	$(color = red \wedge body = full \wedge flavor = strong) \vee$ $(color = white \wedge body = light \wedge flavor = moderate) \vee$ $(sugar = offDry \wedge country = Germany)$
3	$(sugar = sweet \wedge country = France) \vee$ $(sugar = dry \wedge country = Spain) \vee$ $(color = rose \wedge flavor = moderate)$
4	$(color = red \wedge body = medium \wedge flavor = moderate) \vee$ $(color = rose \wedge body = light \wedge sugar = dry) \vee$ $(color = white \wedge body = full \wedge sugar = dry)$
5	$(color = red \wedge body = medium \wedge flavor = moderate) \vee$ $(color = rose \wedge body = full \wedge country = Italy)$
6	$(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$ $(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA)$
7	$(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$ $(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA) \vee$ $(sugar = sweet \wedge color = rose \wedge body = medium \wedge country = Portugal)$
8	$(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$ $(sugar = offDry \wedge color = white \wedge body = medium \wedge flavor = delicate \wedge country = Australia)$
9	$(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$ $(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA) \vee$ $(sugar = sweet \wedge color = rose \wedge body = medium \wedge country = Australia)$
10	$(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$ $(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA) \vee$ $(sugar = sweet \wedge color = rose \wedge body = medium \wedge country = Portugal) \vee$ $(sugar = offDry \wedge color = red \wedge body = full \wedge flavor = strong \wedge country = NewZeeland)$

Table 4.3: Buyer agent's preferences

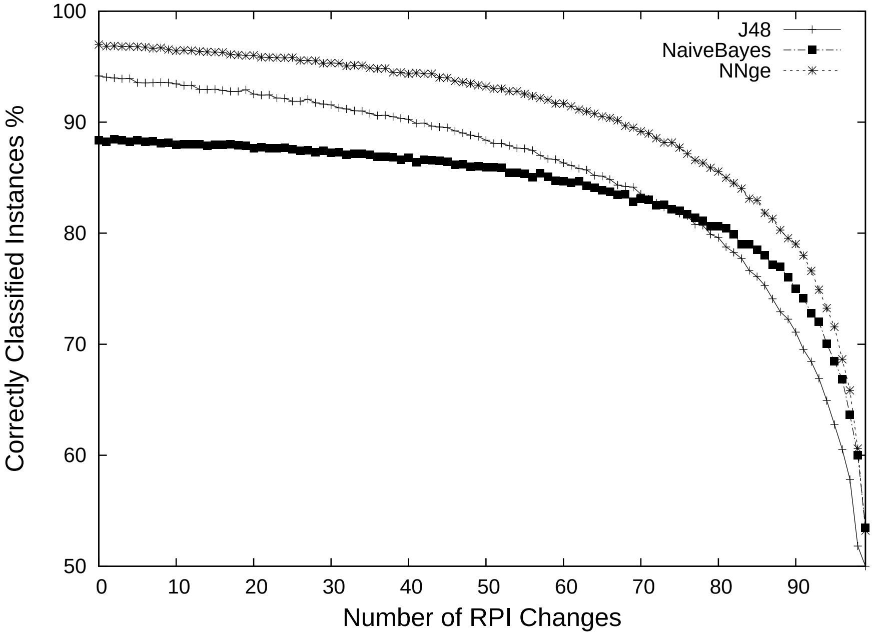

Figure 4.6: Privacy Preservation when changing RPIs

calculate the accuracy of the resulting classifier.

We implemented seller agents so that they use a different classifier to obtain a model of the buyer agents' preferences based on the previous negotiations with them. In this way, we repeat the overall experiment to obtain the results regarding three differnt classifiers. Specifically, we consider the same classifiers as in Serrano et al. (2011): the J48 decision tree algorithm (an implementation of the C.45 algorithm), the NNge classification rules algorithm (Nearest neighbor like algorithm using non-nested generalized exemplars) and the BayesNet classifier that is a classifier based on Bayesian networks.

Figure 4.6 shows the results obtained for our experiment. We can see that the percent of correctly classified instances behaves very similar regardless of the learning algorithm used. As expected, the more a buyer agent changes its RPI, the more inaccurate the learning algorithms become. In other words, the more a buyer agent changes its RPI, the less the seller agent is able to obtain a preference model of the buyer agent. Therefore, buyer agents can avoid that the seller agent performs information processing so that it cannot make any secondary use of the processed data.

Another remarkable phenomenon is that it seems that there is a threshold in the number of RPI changes ($\approx$ 90 RPI changes) from which the accuracy of learning algorithms decreases at a faster rate. This reinforces the thesis of some privacy-enhancing technologies researchers that encourage users to change their identities as often as possible. Moreover, it is clear that the maximum privacy preservation is achieved when buyer agents change their RPI for each new interaction, which is known as transaction pseudonyms in the privacy-enhancing technologies literature.

4.8.2. Avoiding Trust and Reputation Vulnerabilities

Trust and reputation also play a crucial role in this scenario. Buyer agents must be able to choose among seller agents which sell the same wines. One of the important dimensions that a buyer will take into account in her decision is the trust she has in each seller agent. This trust can be based on successful previous interactions with the same seller agent. A buyer agent can trust in a seller agent regarding past interactions by measuring: whether or not the seller agent provisioned the wine, the overall quality of the wine bought, if there were hidden costs, etc. A buyer agent can also trust in a seller agent regarding some attributes of the seller agent's partial identity in the electronic market: registration date, corporate title, skills, etc.

Another important dimension that a buyer agent will take into account in her decision buying a service is the reputation of the seller agent. In this case, it is not what an agent thinks of a given seller agent but what it is generally said about the seller agent in the electronic market.

For the sake of simplicity, we assume that seller agents do not provide a service until they are paid. Therefore, the reputation of buyer agents and the trust other buyer and seller agents have in them are not treated.

In this scenario the PIUP is a great concern. Seller agents should not be able to get rid of their trust and reputation assessments. This could cause important money loss. For instance, a seller agent can be cheating buyer agents by getting paid for a service which will never be delivered. This obviously decreases the trust and reputation that buyer agents have in this seller agent. Hence, this seller agent decides to quit the electronic market and re-entry into it with a new fresh identity, restarting her trust and reputation assessments from scratch. Another example would be a seller agent which sell the same service under different partial identities. In this sense, the probability that a buyer agent chooses one of their partial identities as the provider of the service increases.

We implemented one seller and three buyers. Each buyer uses its own trust and reputation machinery to model the trustworthiness of the sellers based on previous interactions and personal attributes of the sellers. The PPIs issued by the PIdP take values for two attributes: name and role. Both sellers and buyers register into the system using the PPI that the PIdP issued for them — so that the system does not know the real identity of the legal person that agents are acting on behalf of. In this way, buyers are able to identify providers from previous interactions and build their own trust and reputation models being sure that the seller will not be able to hold any other PPI.

The seller follows a normal distribution with a mean of 0 and standard deviation of 1 to model whether it carries out the service requested in the way consumers expect it. In this sense, when a buyer requests a service to the seller, if the value returned is in the interval [-1,1], the buyer considers that the seller performed as expected. If the value returned is out of this interval the buyers consider that the seller did not perform as expected. When the seller performs as expected, buyers rate them with 1. When the seller does not perform as expected, buyers rate them with 0. These ratings are inputs of the trust and reputation model each buyer has.

Each buyer runs a different trust and reputation model that is fed using past interactions with sellers and attributes from sellers' partial identities. We implemented three models (each one for each buyer), one simply using a mean of all the previous performances to compute a trust value, one using the SinAlpha trust model that con-

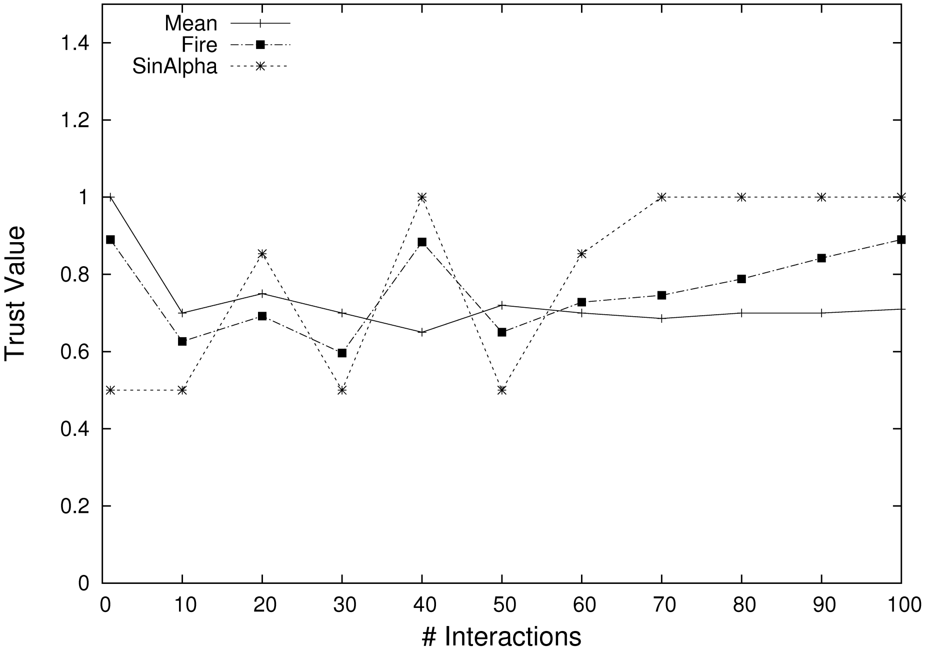

Figure 4.7: Buyer Trust Values

siders previous interactions, and finally, one using the Fire trust and reputation model which uses, among other information, previous interactions and the role of the entities to be trusted. Figure 4.7 presents trust values for each buyer after 100 interactions with the seller. These trust values are the result of each buyer's trust and reputation model given the results of the interactions with the seller.

4.8.3. Discussion

To sum up, the application scenario benefits from the following features that our proposal provides:

- *Multiple RPIs*. Buyers can hold multiple RPIs and use a different one for each interaction with the seller. Therefore, they are able to avoid that the seller performs buyer profiling, as it has been shown in section 4.8.1.

- *Authentication of Partial Identities*. Buyers and sellers are able to authenticate their partial identities (both PPIs and RPIs). Therefore, they are allowed to recognize to each other from interaction to interaction and establish trust and reputation relationships as shown in section 4.8.2.

- *PIUP avoidance*. There is no chance for a buyer or a seller to have two different PPIs. Therefore, trust and reputation relationships built through PPIs avoid PIUP.

- *Hiding of original partial identities*. Both the PIdP and the IdP do not make the partial identities needed to obtain a PPI or a RPI available. Therefore, the rest of the agents are a priori not able to link a partial identity used to the corresponding original real identity or PPI.

- *Entity accountability*. If an agent misbehaves when using its PPI, the PIdP can disclose its real identity if required by a court. If an entity misbehaves when using one of its RPIs, IdPs can disclose the PPI that the entity used to obtain a RPI. Then the PIdP can use this PPI to finally disclose the real identity of the entity.

4.9. Conclusions

In this chapter, we propose formalized definitions of partial identities and their relationship to trust and reputation. Partial identities are a key concept for identifying entities. Moreover, they play a crucial role in trust and reputation, modeling part of the context where trust and reputation take place. In this sense, both trust and reputation are established through partial identities.

We also define the *partial identity unlinkability* problem (PIUP) based on partial identities. PIUP can be more or less harmful depending on the final domain of the application using trust and reputation models. In domains where users can be seriously harmed (e.g. in an e-marketplace by losing money) PIUP needs, at least, to be considered.

We finally propose a privacy-preserving solution to PIUP. An agent can create as many RPIs as needed to avoid information processing. Otherwise, an agent can use a PPI if it is interested in building trust and reputation. Thus, other agents can trust in this agent while being sure that it cannot perform whitewashing and sibyl attacks.

We implemented a prototype to validate our solution to PIUP. However, further research is needed in order to integrate our proposal into an agent platform. Such an integration would result in a complete architecture for deploying agent-based trust and reputation systems without PIUP and respecting privacy concerns. The following chapter (Chapter 5) includes the design of this architecture and its implementation and integration into the Magentix2 agent platform.

CHAPTER 5

Privacy-enhancing Agent Platform

Contents

5.1. Introduction

In this chapter, we propose the implementation and integration of the model presented in Chapter 4 into the Magentix2 agent platform. We also propose a secure agent communication mechanism for Magentix2. As a result, Magentix2 enhances the privacy of the agent-based applications built on top of it. Specifically, Magentix2 provides mechanisms to alleviate two information-related activities that can represent a major threat for privacy: information collection and information processing (Rannenberg et al., 2009).

Information collection refers to the process of gathering and storing data about an individual. For instance, an attacker can be listening to the messages that two agents exchange over the network and simply gather the information that is in the content of these messages. Applications need to be secure to avoid undesired information collection (Garfinkel, 2009). Information collection is alleviated in Magentix2 by providing confidentiality in agent communications. Confidentiality prevents sensitive personal information from being accessed by any other third party that is different from the agent to which the information is directed to.

Information processing refers to the use or transformation of data that has already been collected Spiekermann and Cranor (2009), even though this information has been collected by mutual consent between two parties. In the previous chapter (Chapter 4), we proposed an identity management model for agents in a Multi-agent System. This model alleviates the problem of information processing by allowing agents to hold as many identities as needed to minimize data identifiability, i.e., the degree by which personal information can be directly attributed to a particular principal. We described a prototype to validate our model in Chapter 4. However, further research is needed in order to integrate our proposal into an AP. In this chapter, we describe the integration that we carried out of our proposal into the Magentix2 AP.

In our proposed model in Chapter 4 information processing is alleviated without compromising accountability, trust, and reputation. Accountability refers to the ability to hold entities responsible for their actions Bhargav-Spantzel et al. (2007). Accountability usually requires an unambiguous identification of the principal involved. Thus, this principal can be liable for her/his acts. Commercial systems emphasize accountability because, in these environments, principals can be subject to serious losses such as money loss. Moreover, the sense of impunity generated by the lack of accountability could even encourage abuse. Accountability is of crucial importance for agent-based technologies because it helps to promote trust in agent-based applications, which is needed for principals to be willing to engage with and delegate tasks to agents (Fasli, 2007a). Morevoer, there is usually the need to equip agents with models to reason about and assess trust towards other agents in an agent-based application (Fasli, 2007b). These models allow agents to select the best and most reliable partnership in a specific situation and to avoid partners of previous unsuccessful transactions. However, minimizing data identifiability may also have a direct impact on trust and reputation models. The ability to hold multiple pseudonyms (as is sometimes required to minimize data identifiability) causes the well-known identity-related vulnerabilities of most current trust and reputation models (Carrara and Hogben, 2007). These vulnerabilities can place the system in jeopardy, causing significant money loss.

The remainder of this chapter is organized as follows. Section 5.2 gives a brief overview of the Magentix2 AP. Section 5.3 presents the Magentix2 agent identity management support. Section 5.4 presents the secure mechanism for agent communication. Section 5.5 presents an application scenario. Finally, Section 5.7 presents some concluding remarks and future work.

5.2. Magentix2 Agent Communication

The Magentix2 AP focuses on providing support for open MAS. Magentix2 uses AMQP[1] Vinoski (2006) as a foundation for agent communication. This standard facilitates the interoperability between heterogeneous entities. Magentix2 allows heterogeneous agents to interact with each other via messages that are represented following the FIPA-ACL FIPA (2001b) standard, which are exchanged using the AMQP standard.

[1] `http://www.amqp.org/`

Magentix2 uses the Apache Qpid[2] open-source implementation of AMQP for Agent Communication. Apache Qpid provides two AMQP servers, implemented in C++ (the one we use) and Java. Qpid also provides AMQP Client APIs that support the following languages: C++, Java, C# .NET, Ruby, and Python. Qpid allows distributed applications made up of different parts written in any of these languages to communicate with each other. What is more, any client that is developed using one of the Qpid Client APIs is able to communicate with any client that is developed using any other AMQP-compliant API via any AMQP server implementation, as long as both server and clients implement the same version of the AMQP standard.

Figure 5.1 shows an overview of the Magentix2 agent communication architecture. Magentix2 is composed by one or more (in this case federated) AMQP Servers (QPid brokers). Magentix2 agents act as AMQP Clients (using Qpid Client APIs) that connect to the Qpid broker and are then able to communicate with each other. Magentix2 agents can be located in any Internet location, they only need to know the host on which the Qpid broker (or one of the federated Qpid brokers) is running.

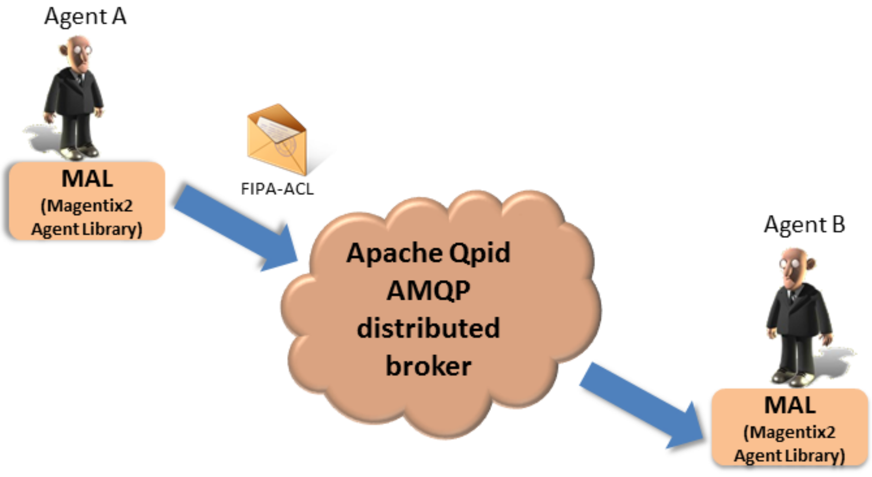

Figure 5.1: Magentix2 Agent Comunication Architecture

[2] http://qpid.apache.org/

Magentix2 provides a Java library, which is called the Magentix2 Agent Library (MAL), to facilitate the development of agents. This API allows agent programmers to specifically focus on creating FIPA-ACL messages and sending and receiving them, without dealing directly with the Qpid Client Java API. Currently, this API is only written in Java, but the existence of multiple QPid Client APIs for several programming languages enables the development of agents written in different programming languages. What is more, any proprietary implementation that follows both AMQP and FIPA-ACL standards would be interoperable with Magentix2 agents.

5.3. Magentix2 Agent Identity Management

Magentix2 implements the agent identity management model presented in Chapter 4. This model is based on the concept of partial identity. A partial identity can be seen as a set of attributes that identifies an entity in a given context. They are composed of a pseudonym that is unique within a context and other attributes that describe the entity within that context (roles, location, preferences, etc.).

This model considers two kinds of partial identities: permanent partial identities (PPIs) and regular partial identities (RPIs). A PPI must contain a permanent pseudonym (*once-in-a-lifetime* pseudonym) for a given system. Thus, agents can only hold one PPI in this given system. A RPI can contain a regular pseudonym that does not pose any limitation on the number of these pseudonyms per agent and per system. Although both kinds of partial identities enable trust and reputation relationships, only PPIs guarantee that identity-related vulnerabilities are avoided. Therefore, agents will choose to establish trust and reputation through PPIs if they want to avoid identity-related vulnerabilities. If they want to avoid information processing, they can use as many RPIs as needed. For instance, an agent can use a different RPI for each different transaction (transaction pseudonyms).

This model also considers the concept of real identities. Real identities identify entities that can be liable for their acts in front of the law, such as human beings, companies, etc. Real identities are used for accountability concerns such as law enforcement. For this reason, real identities are restricted to only legal persons. A real identity, for example, would be: *Bob Andrew Miller, born in Los Angeles, CA, USA on July 7, 1975*. Software entities (intelligent agents, virtual organizations, etc.) cannot have real identities because, up to now, they cannot be liable for their acts in front of the law[3].

Magentix2 complies with the client part of the Identity Metasystem Interoperability standard[4]. This standard specifies the interfaces for the secure web services provided by User-Centric Privacy-Enhancing Identity Management Systems (Clauβ et al., 2005). These systems support the process of management of partial identities. They provide the following facilities:

- *Identity Providers (IdPs)*, which issue partial identities and validate these identities to other Relying Parties.
- *Relying Parties*, which are a set of APIs for verifying partial identities against an Identity Provider.
- *Identity Selectors*, which provide a simple way to manage partial identities and choose which partial identity to use in a given context.
- *Attribute Services*, which allow the specification of access control rights of relying parties over the attributes in a partial identity.

Figure 5.2 shows an overview of the Magentix2 agent identity management support. The Magentix2 Management Service (MMS) is a secure web service that acts as a Relying Party, i.e., it is able to request IdPs to verify partial identities. The MMS is in charge of dynamically signing digital certificates for agents to communicate securely in Magentix2 (as described in section 5.4). Agents request the signing of digital certificates to the MMS using one of their partial identities. The MMS must verify the partial identity that the agent used before signing the digital certificate.

[3] This may change in the future if they finally achieve some kind of legal personality, as suggested by Balke and Eymann (2008). In this case, they may have a real identity for accountability concerns as well.

[4] `http://docs.oasis-open.org/imi/identity/v1.0/identity.html`

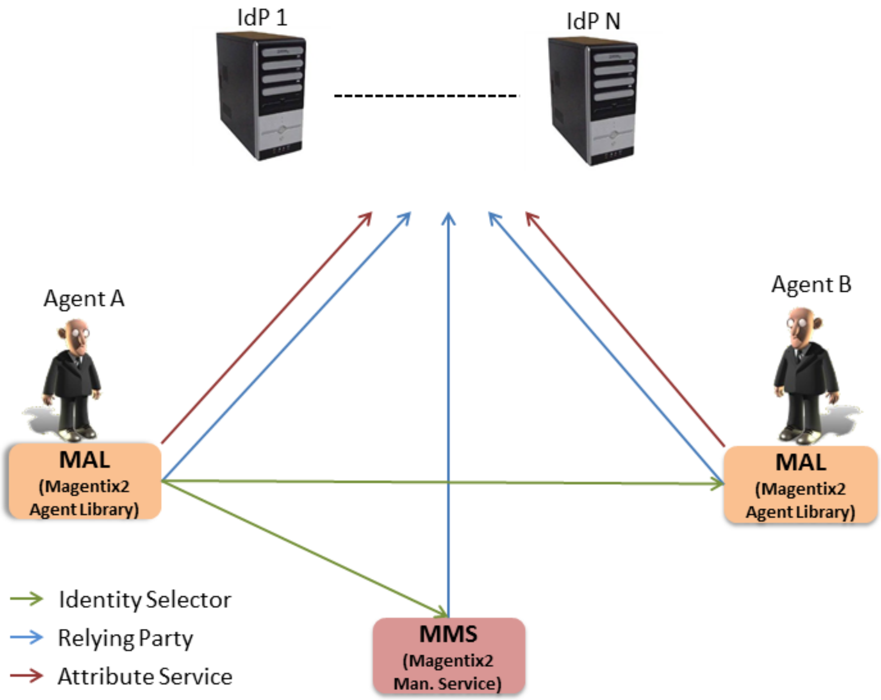

Figure 5.2: The Magentix2 agent identity management support.

The Magentix2 Agent Library (MAL) implements clients for Identity Selectors, Relying Parties, and Attribute Services. Therefore, agents in Magentix2 can select the partial identity to use in a given transaction, verify the partial identities of other agents, and specify access control for attributes in their partial identities.

IdPs are classified according to the type of partial identities they issue. The Permanent Identity Provider (PIdP) is an IdP (or a federation of IdPs[5]) that issues PPIs to the agents taking part in the specific system. Agents must register using a real

[5] User-Centric Identity Management Systems support the federation of IdPs that belong to the same and also different remote security domains across the Internet. Therefore, a PIdP can be implemented as a federation of IdPs instead of only one IdP, minimizing the typical drawbacks of a centralized trusted third party, such as being a single point of failure (SPOF) and a possible efficiency bottleneck. Examples of identity federation standards are the Liberty Alliance Identity Federation Framework http://projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications/ and WS-Federation http://www.ibm.com/developerworks/library/specification/ws-fed/.

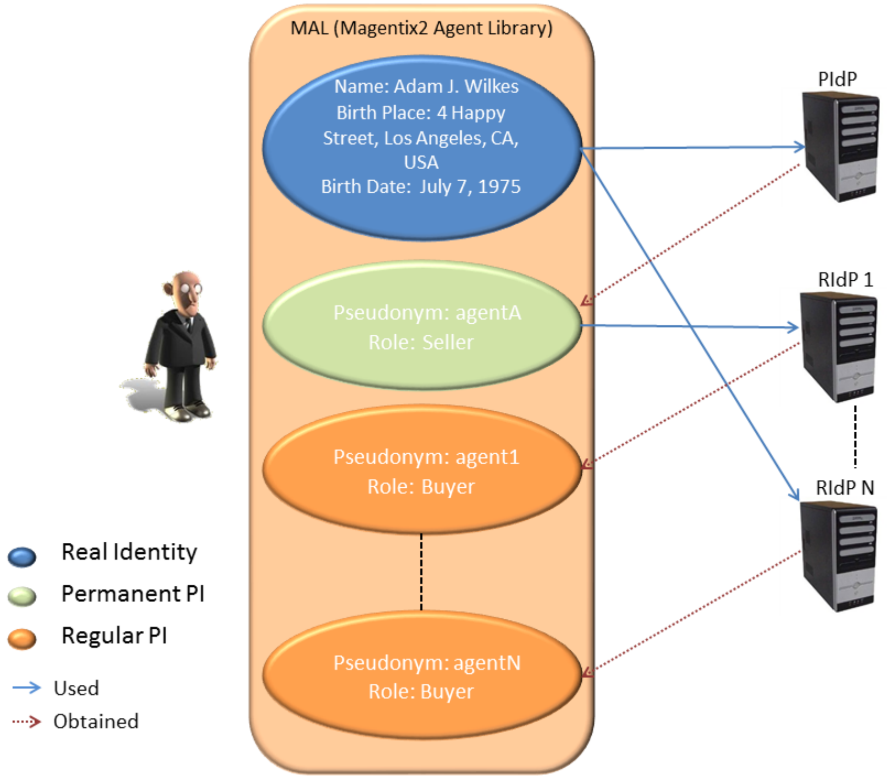

Figure 5.3: An example of the Partial Identities of an agent.

identity that the PIdP will not reveal to other agents or to Magentix2. The PIdP is also in charge of forcing agents to only hold a single PPI in this specific system.

Regular Identity Providers (RIdPs) issue RPIs to agents. Agents request RPIs by providing either a real identity, or a PPI that RIdPs will not reveal to others. There is no limitation in the number of RIdPs per system or in the number of RPIs per agent and per system.

Figure 5.3 shows an example of an agent and its partial identities. The agent's principal has a real identity with an attribute name *Adam John Wilkes*. Using this real identity, the agent has obtained a PPI from the PIdP that includes two attributes: name and role. This entity has also obtained N RPIs from N different IdPs. Some of the RPIs are obtained by providing a PPI (such as RPI 1) and other RPIs are obtained using a real identity (such as RPI N).

5.4. Magentix2 Secure Agent Communication

Agent communication in Magentix2 is based on AMQP. The AMQP standard specifies secure communication by tunneling AMQP connections through SSL Frier et al. (1996) (so-called amqps). Apache Qpid implements SSL support for AMQP. SSL authenticates communicating parties based on digital certificates. Thus, it needs a configured Public Key Infrastructure (PKI). The Magentix2 PKI is set during installation time. Firstly, the Magentix2 certificate authority (MCA) is created. Secondly, certificates for the Magentix2 Management Service (MMS) and the Qpid Broker are created using this certificate authority. Digital certificates for agents are created automatically by the MAL and dynamically signed by the MCA through the MMS at execution time (as described below).

The MMS is a front-end of the MCA. It is implemented as a secure web service. The MMS is in charge of dynamically signing digital certificates for agents, which can use these certificates to communicate securely. The MMS service needs two inputs: the agent pseudonym and a non-signed digital certificate (both represented as a blue arrow in Figure 5.4). The first input is the pseudonym in the permanent or regular partial identity (issued by a permanent or regular IdP) that the agent uses to invoke the MMS. The second input is a non-signed certificate that contains the agent's public key (this is the certificate that is to be signed). The agent key pair (private and public key) and this certificate are created by the MAL *locally* for each agent and for each new partial identity.

The MMS produces one output: the digital certificate *signed* by the MCA (represented as a red doted arrow in Figure 5.4). The MMS produces this output after: (i) verifying that the pseudonym is the same as the one in the partial identity used to invoke the secure web service; (ii) verifying the partial identity against the IdP that issued it; (iii) and finally signing the certificate using the MCA. Agents can then use this signed certificate to communicate to other Magentix2 agents. Figure 5.4 shows an example of an agent with pseudonym A that obtains a certificate from the MMS. From this moment on, agent A can communicate securely with agent B.

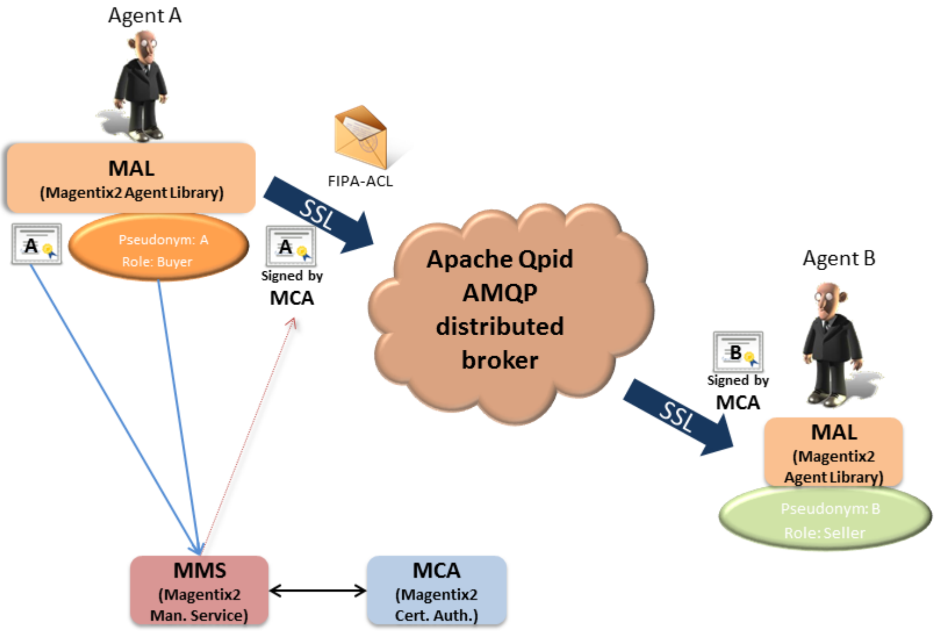

Figure 5.4: Secure Agent Communication in Magentix2.

The AMQP connection of every agent to the Qpid broker is tunneled through SSL. Hence, the communication between two Magentix2 agents is provided with confidentiality and integrity out of the box. To ensure the authenticity of the sender pseudonym in a FIPA-ACL message (recall that in Magentix2 FIPA-ACL messages are encapsulated into AMQP messages), an agent must verify that the pseudonym of the sender in the AMQP sender message field is the same as the pseudonym of the sender in the FIPA-ACL sender message field upon receiving a new message. This is performed automatically by the Magentix2 agent library.

5.5. Application Scenario

We describe a Business-to-Consumer (B2C) electronic marketplace where seller agents retail medicines to buyer agents. Privacy can be of great concern in this scenario. A principal may need to acquire different medicines but does not want these medicines to be linked to her/him. For instance, there are medicines that are only prescribed for one specific illness, such as asthma. Therefore, buying these medicines automatically discloses the illness that the principal is suffering from. A principal may prefer to conceal his/her real identity when acquiring such medicines. This is because she/he is probably concerned about her/his illnesses being in the public domain and affecting other aspects of her/his life such as finding a job.

The principal can instruct her/his buying agent to obtain a partial identity that is different from her/his real identity before entering the marketplace. IdPs act as independent third parties that must be trusted by both Magentix2 and the agents. To obtain new partial identities (PPIs or RPIs), agents must provide a real identity, or a PPI to IdPs. IdPs do not make the original partial identities available. Therefore, the rest of the agents in the marketplace and Magentix2 itself are, a priori[6] , not able to link a partial identity to the corresponding original real identity or PPI.

Moreover, some asthma medicines may require the principal to be of legal age. The agent then asks a RIdP for a RPI containing a pseudonym (e.g. a random number) and containing an attribute that states that the agent's principal is of legal age. The RIdP can check this by verifying the birth date in the real identity of the agent's principal. The agent can show this attribute when purchasing medicines that require being of legal age and concealing this attribute otherwise (e.g. when purchasing medicines for a cold).

Moreover, seller agents could construct a detailed profile on the medicines needed

[6] We assume that payments are carried out using some kind of anonymous payment mechanism and that deliveries are carried out using some anonymous delivery system. Hence, credit card numbers and delivery addresses do not need to be disclosed when an agent acquires a product. For instance, the untraceable electronic cash presented by Chaum et al. Chaum et al. (1990) can be used for anonymous payments. For anonymous deliveries, the privacy-preserving physical delivery system presented by Aïmeur et al. Aïmeur et al. (2006) can be used.

by the principal. This allows seller agents to practice price discrimination. For instance, seller agents could infer that the buyer agent periodically purchases such medicines. Thus, they could charge a slightly increasing cost for each new transaction. The principal can instruct her/his buyer agent to use a different new RPI each time it purchases asthma medicines in order to avoid this. Thus, it is difficult for a seller agent to be aware that different transactions were performed by the same buyer agent under different RPIs.

Buyer agents are able to choose among seller agents that sell the same medicines. One of the important dimensions that buyers will take into account in their decisions is the trust that they have in each seller agent. This trust can be based on successful previous interactions with the same seller agent. A buyer agent can trust in a seller agent in regard to past interactions by measuring: whether or not the seller agent shipped the product in time, the overall quality of the product bought, if there were hidden costs, etc. If the buyer agent has no previous interactions with a seller agent, the buyer agent can also consider the reputation of the seller agent in the marketplace.

In this scenario, identity-related vulnerabilities are a great concern. Seller agents should not be able to get rid of their trust and reputation ratings. This could cause important money loss. For instance, a seller agent could be cheating buyer agents by shipping medicines with a quality that is lower than expected. This obviously decreases the trust and reputation that buyer agents have in this seller agent. Hence, this seller agent decides to quit the electronic market and to reenter it with a new fresh partial identity, restarting its trust and reputation ratings from scratch. Another example would be a seller agent that sells the same medicine under different partial identities. This way, the probability of a buyer agent choosing one of its partial identities as the seller of the product increases.

If a buyer agent (and by extension its principal) wants to avoid identity-related vulnerabilities, it should only consider seller agents with a permanent partial identity (PPI). Thus, the buyer agent can use its own trust and reputation machinery to model the trustworthiness of these sellers and be sure that whitewashing and sibyl attacks are avoided.

Finally, accountability also needs to be considered. For instance, there may be seller agents that sell medicines illegally. For these cases, the real identity of the principal behind a seller agent that sells medicines illegally can be known. A court could require the PIdP to disclose the real identity behind a PPI. As a result, the principal holding this real identity could be sued for selling medicines illegally. The final punishment may depend on the applicable laws for such a case.

5.6. Performance Evaluation

In chapter 4 we described an experiment that demonstrates that changing RPIs can minimize information processing. However, changing RPIs could also have costs associated to the change, e.g., a temporal cost. In this section, we carry out an experiment to evaluate the temporal cost of changing RPIs in the Magentix2 agent platform.

We performed a similar experiment as the one presented in Chapter 4, in which agents change their RPI a number of times in order to reduce information processing. In this case, we focus on only two agents, one buyer agent and one seller agent. Moreover, we do not calculate the accuracy of the preference model that the seller obtains but we calculate the temporal cost for the buyer to change its RPI a specific number of times. This is in order to ascertain whether or not it is temporally feasible for a buyer agent to change their RPI as many times as needed to prevent the seller agent from constructing a detailed model on the buyer agent's preferences.

We perform a simulation in which the buyer agent carries out 100 different purchases of a bottle of wine. Each purchase involves a negotiation with the seller agent to get the desired wine. We assume that negotiations are always successful. However, we consider that negotiations can randomly involve from 1 up to 10 rounds of the protocol. That is, we simulate negotiations on which a buyer agent and a seller agent perform a maximum number of 10 rounds of the protocol. Specifically, the buyer agent repeats the 100 purchases with a varying number of RPI changes. That is, the buyer agent starts with 100 purchases and without any RPI change and end

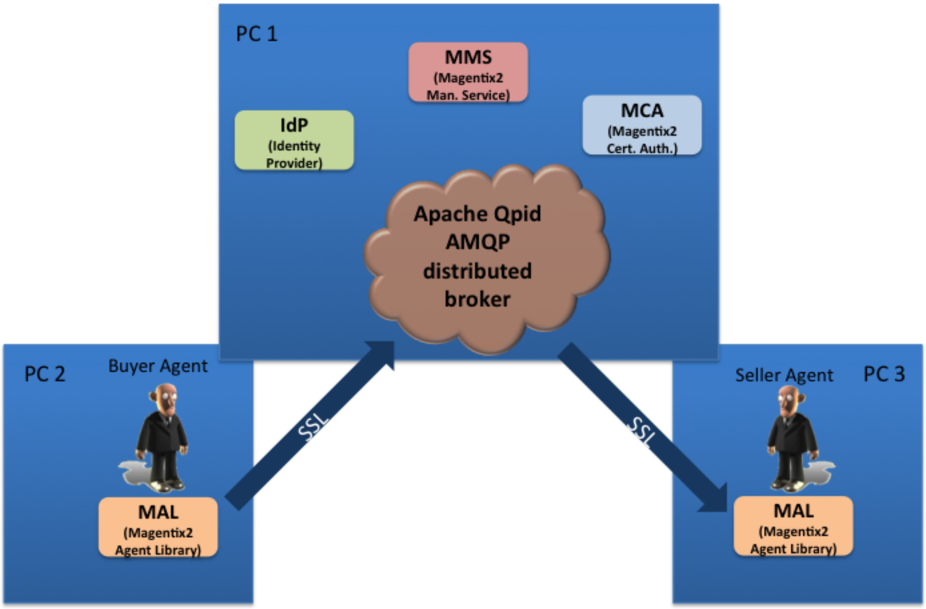

Figure 5.5: Location of the different components.

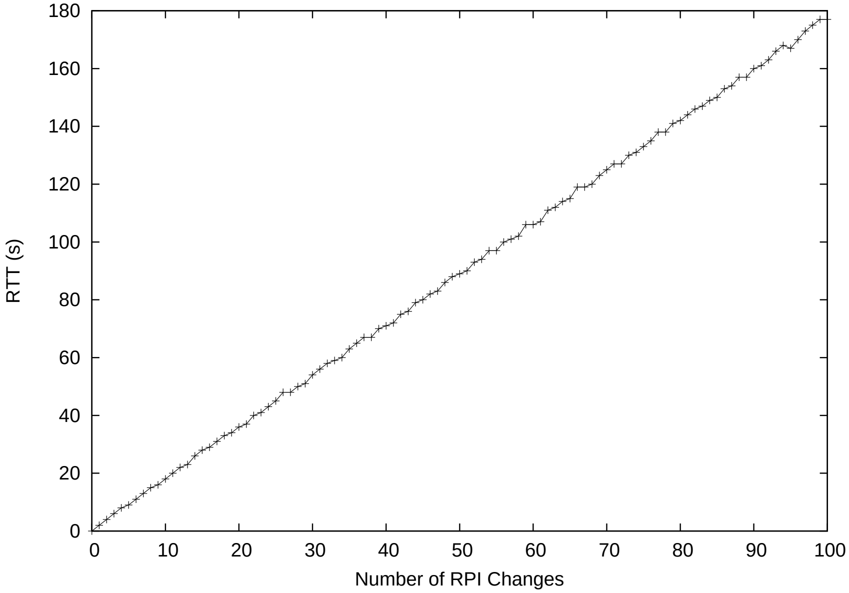

Figure 5.6: Performance per number of RPI changes

up using a different RPI for each of the 100 purchases (this is known as transaction pseudonyms). For each number of RPI changes we calculate the RTT time of the messages exchanged between the buyer agent and the seller agent.

In order for the experiment to be in an absolutely controlled environment, we do not use any external IdP but we use the prototype described in Chapter 4 as both the PIdP (the IdP that issues PPIs) and the RIdP (the IdP that issues RPIs). Moreover, we used 3 PCs Intel(R) Core(TM) 2 Duo CPU @ 2.60GHz, 1GB RAM, Ubuntu 11.04 (x86_64) and Linux Kernel 2.6.38. The computers are connected to each other via a 100Mb Ethernet switch. The security parameters are the following: certificate keys are 1024 bits RSA keys, SHA-1 hash function with 96-bit keys to perform HMAC computations, and the saml2 tokens to be issued by the IdP contain keys of 256 bits. The location of the different components is shown in Figure 5.5: PC number 1 runs the Qpid Broker, the MMS, the MCA, and the IdP; PC number 2 runs the buyer agent; and finally, PC number 3 runs the seller agent.

Figure 5.6 shows the results obtained. These results mean that changing a RPI has a temporal cost that is linear with the number of changes to be made. These results also mean that the temporal cost of a single change is constant, and thus, it is not related to the number of previously performed changes. Therefore, we can claim that agents developed in Magentix2 can minimize information processing about their principals' data (as shown in Chapter 4) without incurring in a not affordable temporal cost. Moreover, as the cost of single change is constant, a buyer agent can predict in advance the temporal cost of the changes it requires to reduce seller agents from processing its data, and then, decide if it performs the required changes or not.

5.7. Conclusions

In this chapter, we present the privacy-enhancing support that Magentix2 provides. This privacy-enhancing support also avoids identity-related vulnerabilities of trust and reputation models as well as the lack of accountability of the principals involved. All these features are crucial for encouraging principals' trust in agent-based applications.

Agents running on Magentix2 can use these features at will depending on their principals' needs. An agent can create as many RPIs as needed to avoid information processing. Otherwise, an agent can use a PPI if it is interested in building trust and reputation. Thus, other agents can trust in this agent while being sure that it cannot perform whitewashing and sibyl attacks.

We carried out a performance evaluation to validate our implementation. The experimental results we obtained suggest that changing a RPI has a temporal cost that is linear with the number of changes to be made. Therefore, agents developed in Magentix2 can minimize information processing about their principals' data by changing their RPIs as much as needed (as shown in Chapter 4) without incurring in a not affordable temporal cost.

As we have detailed in this chapter, Magentix2 implements the client part of the Identity Metasystem Standard. We have focused on describing how Magentix2 an the agents running on top of it make use of three parts of the standard: Identity Providers, Relying Parties, and Identity Selectors. However, we have not provide any insight on the remaining part of the standard, i.e., Attribute Services. Magentix2 agents can use Attribute Services to specify access control rights of relying parties over the attributes in a partial identity. The problem that arises when considering Attribute Services is: how could an agent decide whether or not to grant access to an specific attribute in one of its partial identities to other agents? In the next chapter (Chapter 6), we propose a model for agents to decide when disclosing an attribute is acceptable or not.

CHAPTER 6

Self-disclosure Decision Making

Contents

6.1. Introduction

An autonomous agent usually encapsulates personal data attributes (PDAs) describing its principal (Fasli, 2007b). PDAs can describe a great range of topics (Rannenberg et al., 2009). For instance, names (real names, pseudonyms), physical characteristics, competences, preferences, roles in organizations and institutions, social characteristics (affiliation to groups, friends), location (permanent address, geolocation at a given time), and even behaviors (personality, mood). When agents carry out interactions on behalf of their principals, they usually exchange PDAs. Hence, they play a crucial role to safeguard and preserve their principals' privacy (Fasli, 2007b).

Westin (1967) defined privacy as a "personal adjustment process" in which individuals balance "the desire for privacy with the desire for disclosure and communication". Humans have different general attitudes towards privacy that influence this adjustment process (Olson et al., 2005; Ackerman et al., 1999; Westin, 1967): privacy fundamentalists are extremely concerned about privacy and reluctant to disclose PDAs; privacy pragmatists are concerned about privacy but less than fundamentalists and they are willing to disclose PDAs when some benefit is expected; and finally, privacy unconcerned do not consider privacy loss when disclosing PDAs. In online interactions, just 10% of users are unconcerned (Westin, 2003). Therefore, privacy is of actual concern to most users in the digital world.

Westin proposed his definition for privacy long before the explosive growth of the Internet. As far as we are concerned, it also applies to autonomous agents that engage in online interactions that require the disclosure of their principals' PDAs. Agents, then, should be able to autonomously balance their desire for privacy and their desire for disclosure and communication. Thus, they need to incorporate self-disclosure[1] decision-making mechanisms allowing them to autonomously decide whether disclosing PDAs to other agents is acceptable or not.

As pointed out in Chapter 2, current self-disclosure decision-making mechanisms are based on the privacy-utility tradeoff (Krause and Horvitz, 2008; Lebanon et al.,

[1] We consider self-disclosure as the process by which individuals *disclose* PDAs about themselves to others (Green et al., 2006).

2006). This tradeoff considers the direct benefit of disclosing a PDA and the privacy loss it may cause; for instance, the tradeoff between the reduction in time to perform an online search when some PDAs (e.g. geographical location) are given and the privacy loss due to such disclosure (Krause and Horvitz, 2008).

There are many cases where the direct benefit of disclosing PDAs is not known in advance. This is the case in human relationships, where the disclosure of PDAs in fact plays a crucial role in the building of these relationships (Green et al., 2006). These relationships may or may not eventually report a direct benefit for an individual. For instance, a close friend tells you what party he voted for. He may disclose this information without knowing (or expecting) the future gain in utility this may cause. Indeed, it might not report him any benefit ever. In this chapter, we present two self-disclosure decision-making mechanisms based on intimacy and privacy measures to deal with these situations. These mechanisms consider psychological findings regarding how humans disclose personal information in the building of their relationships, such as the well-studied *disclosure reciprocity* phenomenon (Green et al., 2006). This phenomenon is based on the fact that one person's disclosure encourages the disclosure of the other person in the interaction, which in turn, encourages more disclosures from the first person.

We use these self-disclosure decision-making mechanisms to model privacy pragmatist and fundamentalist agents. We claim that, privacy pragmatist agents lose less privacy than unconcerned agents in order to achieve the same intimacy level. We also claim that privacy fundamentalist agents lose less privacy than both pragmatist and unconcerned agents but are unable to achieve the same intimacy. To prove these claims, we first present metrics grounded on information theory to measure the intimacy and the privacy loss between two agents; second, we present self-disclosure decision making mechanisms based on these metrics; and third, we present experiments performed comparing agents using these self-disclosure decision-making mechanisms with privacy unconcerned agents that do not use them.

The remainder of the chapter is organized as follows. Section 6.2 introduces Uncertain Agent Identities (UAIs), which is a formalism for describing agents based on PDAs. Section 6.4 presents a measure for the degree of intimacy between two agents based on UAIs. Section 6.5 presents a model for measuring the privacy loss of PDA disclosures based on UAIs. Section 6.6 proposes self-disclosure decision-making mechanisms for autonomous agents based on intimacy and privacy loss. Section 6.7 presents the experiments we carried out. Finally, Section 6.8 presents some concluding remarks.

6.2. Uncertain Agent Identities

We assume a Multiagent System composed of a set of intelligent autonomous agents $Ag = \{\alpha_1, \ldots, \alpha_M\}$ that interact with each other through message exchanges. Agents in Ag are described using the same finite set of PDAs, $A = \{a_1, \ldots, a_N\}$. Each PDA $a \in A$ has a finite domain of possible values $V_a = \{v_1, \ldots, v_{K_a}\}$.

Each agent $\alpha \in Ag$ has values for their PDAs that are not known by the other agents in Ag. Agents are able to disclose PDA values to others, but the values of the PDAs disclosed may not be true. Thus, agents are uncertain about the PDA values of the other agents. Moreover, agents may not even be absolutely certain about the specific values for their own PDAs (e.g. an agent could be uncertain about whether it is competent in performing a given task). Therefore, agents maintain *uncertain agent identities* (UAIs) modeling their own PDAs and the PDAs of the rest of the agents in Ag.

Definition 4 *Given a set of PDAs $A = \{a_1, \ldots, a_N\}$, each one with domain $V_a = \{v_1, \ldots, v_{K_a}\}$, an uncertain agent identity $I = \{P_1, \ldots, P_N\}$ is a set of discrete probability distributions P_i over the values V_{a_i} of each PDA a_i.*

We thus denote P_a as the probability distribution of a over V_a and $p_a(\cdot)$ as its probability mass function, so that $p_a(v)$ is the probability for the value of a being equal to $v \in V_a$.

An agent $\alpha \in Ag$ manages its own UAI and two UAIs associated to each agent $\beta \in Ag \setminus \{\alpha\}$. We will refer to the UAI of an agent α as I_α. We denote $I_{\alpha,\beta}$ as the UAI that α believes that β has, i.e., what α knows (or thinks it knows) about I_β. Moreover, it is crucial for an agent α to also have UAIs modeling what the other agents in Ag may know about its own UAI I_α for measuring privacy loss (as explained in section 6.5). We denote $I_{\alpha,\beta,\alpha}$ as the UAI that α believes that β believes that α has[2].

UAIs may be initialized regarding the actual knowledge that an agent has for the probability distributions of each of the PDAs. For instance, if the agent is completely uncertain about the distribution of a PDA a, then, its probability distribution $P_a \in I$ may be initialized to a uniform, i.e., each $p_a(v)$ may be initialized to $\frac{1}{|V_a|}$ for each $v \in V_a$.

6.2.1. Uncertainty Measures

An agent needs to measure how much uncertainty there is in the probability distribution of a PDA. Taking into account this uncertainty, the agent may decide, for instance, whether or not to take specific actions to reduce this uncertainty under a desired threshold.

A well-known measure of the uncertainty in a probability distribution is Shannon (1948) entropy:

$$H(P_a) \quad = \quad -\sum_{v \in V_a} p_a(v) \log_2 p_a(v) \tag{6.1}$$

The entropy of each probability distribution in an UAI provides a measure of the uncertainty for each PDA. However, as an UAI can span over several PDAs, a method for aggregating the uncertainties of all of the probability distributions in an UAI is needed. In this chapter, we use a simple computational method that is the mean of the uncer-

[2] Subindexes of an UAI should be read from left to right, starting with *"the UAI"* and adding an *"that agent believes"* for each agent that appears separated by a semicolon, except for the agent in the last position which is read as *"that agent has"*. For instance, I_α should be read as *"the UAI that* α *has"*, $I_{\alpha,\beta}$ should be read as *"the UAI that* α *believes that* β *has"* and $I_{\alpha,\beta,\alpha}$ should be read as *"the UAI that* α *believes that* β *believes that* α *has"*.

tainties in each of the probability distributions in an UAI:

$$H(I) = \frac{1}{|A|} \sum_{a \in A} H(P_a) \tag{6.2}$$

With this measure an agent is able to know how certain it is about an UAI. We assume that at initialization time the entropy of an UAI I is the highest possible, i.e., the uncertainty in I will decrease as the agent obtains more information related to the PDAs being modeled.

6.3. Updating UAIs

UAIs are supposed to be dynamic, i.e., they may change as time goes by. These changes will potentially reduce the uncertainty in an UAI. An agent α may update the UAIs that it manages as it gets more information about the probability distributions for the PDAs in these UAIs. In this section, we provide a method for updating the two UAIs that α has per each agent in Ag.

PDA values are private to each agent. We assume that α *discloses* its PDA values for a to β by sending a message[3] $\mu = \langle \alpha, \beta, \langle \alpha, a, P_a \rangle \rangle$, where α represents the sender, β represents the receiver, and $\langle \alpha, a, P_a \rangle$ represents the claim "the probability distribution for the PDA a of α is P_a".

UAIs are updated with the disclosures that agents carry out. The update process of an UAI has two steps: (i) updating the probability distribution of the PDA being disclosed; and (ii) inferring updates of probability distributions of other PDAs based on the PDA being disclosed and other information already known. We denote that an UAI I is updated with a message μ as I^{μ}. Moreover, we denote that an UAI I is updated sequentially and in order considering a tuple of messages $M = (\mu_1, \ldots, \mu_P)$ as I^M.

We now detail how and which UAIs should be updated when receiving and when sending a message.

[3] In this chapter, we use the terms message and disclosure as equivalents because we only consider messages that involve a PDA disclosure.

6.3.1. Receiving a Message

If α receives $\mu = \langle \beta, \alpha, \langle \beta, a, Q_a \rangle \rangle$ from β, then α can update $I_{\alpha,\beta}$ – the UAI that α believes that β has. The resulting UAI is denoted as $I^{\mu}_{\alpha,\beta}$.

Update Given $\mu = \langle \beta, \alpha, \langle \beta, a, Q_a \rangle \rangle$, $P_a \in I_{\alpha,\beta}$, and $r_{\alpha,\beta}$ (which is the reliability that α assesses to β and is explained below), let $S^{\mu}_a = r_{\alpha,\beta} \cdot Q_a + (1 - r_{\alpha,\beta}) \cdot P_a$. Then, we update $P^{\mu}_a \in I^{\mu}_{\alpha,\beta}$ as:

$$P^{\mu}_a = \begin{cases} S^{\mu}_a & \text{if } H(S^{\mu}_a) < H(P_a) \\ P_a & otherwise \end{cases}$$

P_a is only updated if the message produces an information gain, i.e. resulting probability distribution S^{μ}_a is more certain than P_a.

Reliability A model for reliability may be based on the difference between the values that agents claim for their PDAs – the disclosures they send to other agents – and the values observed for these PDAs by other agents. We assume that α builds another UAI $O_{\alpha,\beta}$ that is different from I_α, $I_{\alpha,\beta}$ and $I_{\alpha,\beta,\alpha}$ based on observations. $O_{\alpha,\beta}$ contains probability distributions that α has inferred from the observation of β's behavior. An example of observation may be the following. Let $competentTaskA$ be a PDA with domain $\{true, false\}$. If β discloses $\langle \beta, \alpha, \langle \beta, competentTaskA, \{true \rightarrow 1, false \rightarrow 0\} \rangle \rangle$, α may request β to perform this task. Then, α can *observe* the result of the task to assess whether or not β is actually competent in carrying out the task and may infer the probability distribution for $competentTaskA$ as being $\{true \rightarrow 0.8, false \rightarrow 0.2\}$.

α may measure the reliability of β as follows. Let a be a PDA β disclosed to α, let $P_a \in I_{\alpha,\beta}$ be the probability distribution that α believes that β has (from what β disclosed to α), and let $O_a \in O_{\alpha,\beta}$ be the probability distribution that α has observed for the PDA a of β. Then, the reliability of β as seen by α is:

$$r_{\alpha,\beta} = \frac{1}{|A|} \sum_{a \in A} \frac{1}{1 + \mathsf{KL}(O_a \parallel P_a)}$$

Where $\mathsf{KL}(O_a \parallel P_a)$ is the Kullback-Leibler divergence (Kullback and Leibler, 1951) that measures the distance between two probability distributions:

$$\mathsf{KL}(O_a \parallel P_a) \quad = \quad \sum_{v \in V_a} o_a(v) log_2 \frac{o_a(v)}{p_a(v)}$$

If all the probability distributions that α observed for all the disclosed PDAs from β are close to the probability distributions for these PDAs in $I_{\alpha,\beta}$, then KL values will be close to 0 and $r_{\alpha,\beta}$ will be close to 1. If all the probability distributions α observed for all the disclosed PDAs from β are far from the probability distributions for these PDAs in $I_{\alpha,\beta}$, then KL values will be high and $r_{\alpha,\beta}$ will be close to 0.

Inference The rest of the probability distributions of PDAs not yet disclosed from β to α may be inferred considering the PDAs that have already been disclosed. The inference model that we consider in this chapter is based on the existence of conditional probabilities $\mathsf{Pr}(b \mid a)$[4], considering a as a PDA β disclosed to α and b as the PDA to be inferred. Thus, if Q_b is a probability distribution defined as:

$$q_b(u) \quad = \quad \sum_{v \in V_a} \mathsf{Pr}(b = u \mid a = v) p_a(v)$$

then,

$$P_b^{\mu} = \begin{cases} Q_b & \text{if } H(Q_b) < H(P_b) \\ P_b & otherwise \end{cases}$$

A simple method based on frequencies for estimating these conditional probabili-

[4] More sophisticated methods, e.g. based on bayesian networks (He et al., 2006), could be used. The important point is that inference should be considered when dealing with the disclosure of PDAs.

ties may be:

$$\text{Pr}(b = u \mid a = v) =$$
$$\frac{\left|\{\beta \mid \beta \in Ag \text{ and } P_a, P_b \in I_{\alpha,\beta} \text{ and } p_a(v) > \varepsilon \text{ and } p_b(u) > \varepsilon\}\right|}{|Ag| - 1}$$

This method averages the number of UAIs that α believes that other agents in Ag have in which the probabilities for a and b to be v and u are higher than a threshold ε. This is a simple method for estimating if the values v and u of PDAs a and b are commonly related to each other for agents in Ag. This method requires a minimum knowledge about the other agents in Ag.

6.3.2. Sending a Message

α discloses the probability distribution for its PDA a to β by sending a message $\mu = \langle \alpha, \beta, \langle \alpha, a, Q'_a \rangle \rangle$ to β so that KL$(Q'_a \parallel Q_a)$ determines the level of sincerity of α to β, considering $Q_a \in I_\alpha$. Then, α may update $I_{\alpha,\beta,\alpha}$ – the UAI that α believes that β believes that α has. The resulting UAI is denoted as $I^{\mu}_{\alpha,\beta,\alpha}$.

α updates $P_a \in I_{\alpha,\beta,\alpha}$ replacing it with Q'_a, i.e., α assumes that β believes the probability distribution for its PDA a is Q'_a from this moment on. α may also update the probability distributions of PDAs that α has not yet disclosed to β, which could be inferred from PDAs that α has already disclosed to β using the inference method explained in the above section.

6.4. Intimacy

According to Miller et al. (2007), intimate human partners have extensive personal information about each other. They usually share information about their PDAs, including preferences, feelings, and desires that they do not reveal to most of the other people they know. Indeed, self-disclosure and partner disclosure of PDAs play an important role in the development of intimacyGreen et al. (2006).

An agent α could simply count the number of PDAs disclosed to β, count the number of PDAs that β disclosed to α to estimate its intimacy to β. However, as explained in section 6.3, when disclosing PDAs, it may be the case that more information is being disclosed without explicitly disclosing it. Therefore, PDAs not yet disclosed may be inferred from PDAs already disclosed so that α is actually giving β more information than just the PDAs explicitly disclosed to β.

Uncertainty and information are closely related to each other Klir (2006). The amount of information obtained by an action can be measured by the reduction of uncertainty due to that action. Thus, information may be measured by the difference between the a priori uncertainty – uncertainty before the action – and the a posteriori uncertainty – uncertainty after the action. For instance, as stated in Sierra and Debenham (2007a), if the action is the sending/reception of a message, the information gain that a message provides may be measured by the difference in uncertainty before sending/receiving the message and the uncertainty after sending/receiving the message.

Definition 5 *Given an UAI I and a message μ, the information gain of message μ is:*

$$I(I,\mu) = H(I) - H(I^{\mu})$$

α may measure the amount of information it has about β by measuring the information gain of all the messages received from β. α may measure the amount of information β has about it by measuring the information gain of all the messages that α sent to β.

Definition 6 *Given an UAI I and a tuple of messages M, the information gain of M is:*

$$I(I,M) = H(I) - H(I^{M})$$

Sierra and Debenham Sierra and Debenham (2007b) defined the intimacy between α and β considering the amount of information that α knows about β and vice versa. We adapt this definition for the case of UAIs. Thus, we define intimacy as follows.

Definition 7 *Given the UAIs $I_{\alpha,\beta}$ and $I_{\alpha,\beta,\alpha}$, a tuple of messages M from β to α and a tuple of messages M' from α to β, the intimacy between α and β is:*

$$\mathcal{Y}_{\alpha,\beta} \quad = \quad I(I_{\alpha,\beta}, M) \oplus I(I_{\alpha,\beta,\alpha}, M')$$

Where $\oplus$ is an appropriate aggregation function. $\mathcal{Y}_{\alpha,\beta} = 0$ means that there is no intimacy between α and β from the point of view of α. The higher the $\mathcal{Y}_{\alpha,\beta}$, the more intimacy between α and β from the point of view of α. It is worth noting that the intimacy measure, as we define it, is not necessarily symmetric, i.e., $\mathcal{Y}_{\alpha,\beta}$ may be different from $\mathcal{Y}_{\beta,\alpha}$.

6.5. Privacy Loss

Disclosing PDAs always comes at a loss of privacy because personal information is made known. Therefore, it is crucial for agents to estimate the privacy loss that a disclosure may imply before deciding whether they actually carry it out.

Privacy loss is defined in previous works (Lebanon et al. (2006),Li et al. (2007)) taking into account the sensitivity of the PDAs to be disclosed under a successful identification. As explained in Section 6.2, each agent $\alpha \in Ag$ has its own UAI I_α that is not known by the other agents in Ag. Moreover, α has UAIs that it believes that other agents in Ag believe that α has, i.e., what other agents in Ag may know about I_α. In this sense, α could estimate (from its point of view) the extent to which β knows I_α by measuring the distance between I_α and $I_{\alpha,\beta,\alpha}$. α can calculate this distance by measuring the distance between each probability distribution for each PDA in these UAIs.

Given that a has the probability distributions $P_a \in I_\alpha$ and $Q_a \in I_{\alpha,\beta,\alpha}$, we use the Kullback-Leibler divergence Kullback and Leibler (1951) to measure the distance between P_a and Q_a. KL measures the amount of information needed to encode the differences between two probability distributions.

We assume that agents in Ag can define the *subjective* sensitivity that they attach to their PDAs. Therefore, α has a function $w_\alpha : A \rightarrow [0,1]$ such that $w_\alpha(a)$ is the *subjective* valuation that α attaches to the sensitivity of a.

Based on the Kullback-Leibler divergence and the sensitivity of the PDAs, we define the privacy loss of disclosing a PDA.

Definition 8 *Given two agents α and β, the message μ, and considering $Q_a \in I_{\alpha,\beta,\alpha}$, $Q_a^\mu \in I_{\alpha,\beta,\alpha}^\mu$ and $P_a \in I_\alpha$, the privacy loss for agent α if it sends μ to agent β is:*

$$\mathcal{L}(I_{\alpha,\beta,\alpha},\mu) = \sum_{a \in A} w_\alpha(a) \cdot (\mathsf{KL}(Q_a \parallel P_a) - \mathsf{KL}(Q_a^\mu \parallel P_a))$$

For each PDA, we measure the KL between its probability distribution in $I_{\alpha,\beta,\alpha}$ before being updated taking into account μ and its probability distribution in I_α and the KL between its probability distribution in $I_{\alpha,\beta,\alpha}$ after being updated considering μ and its probability distribution in I_α. Then, we consider the difference between these two KLs stating the amount of information that $I_{\alpha,\beta,\alpha}$ would approach to I_α if the message μ is sent. This amount of information that would be *lost* due to the sending of the message is then weighted by the subjective sensitivity of the PDA. The final result of privacy loss is the addition of the results for all of the PDAs (recall that values for PDAs that are not disclosed could be inferred from PDAs that are disclosed as explained in Section 6.2). $\mathcal{L}(I_{\alpha,\beta,\alpha},\mu) = 0$ means that sending μ to β causes no privacy loss to α. The higher the $\mathcal{L}(I_{\alpha,\beta,\alpha},\mu)$, the more privacy loss sending μ to β causes to α.

As explained later on in section 6.6, it is also useful for agents to measure the total privacy that they have lost due to the messages that they sent to other agents.

Definition 9 *Given two agents α and β, the tuple of all messages M sent from α to β and considering $Q_a \in I_{\alpha,\beta,\alpha}$, $Q_a^M \in I_{\alpha,\beta,\alpha}^M$ and $P_a \in I_\alpha$, the Total Privacy Loss from α to β is:*

$$\mathcal{L}(I_{\alpha,\beta,\alpha},M) = \sum_{a\in A} w_{\alpha}(a)\cdot(\mathsf{KL}(Q_a \parallel P_a) - \mathsf{KL}(Q_a^M \parallel P_a))$$

6.6. Self-disclosure Decision Making

In this section, we present two mechanisms for an agent α to decide which PDAs (if any) to disclose to another agent β. These mechanisms are based on general privacy attitudes and specific willingness to share a PDA. We model pragmatist and fundamentalist attitudes towards privacy. To this aim, we use the information metrics explained above.

6.6.1. Privacy Pragmatist Agents

Privacy pragmatists are concerned about privacy, but they are willing to disclose personal information when some benefit is expected (Olson et al. (2005), Ackerman et al. (1999) and Westin (1967)). In many situations, the actual benefit of disclosing personal information may not be known in advance. We present a self-disclosure decision-making mechanism modeling a pragmatic attitude towards privacy which is grounded on information measures. Specifically, we consider the estimation of intimacy gain between two agents (i.e., the amount of information two agents have about each other) and the privacy loss (the distance between what the agents believe that others believe about them and their actual UAI weighted by a subjective sensitivity).

To estimate the increase in intimacy that the sending of a message μ may cause between α and β, we consider the information gain of μ, i.e. $I(I_{\alpha,\beta,\alpha},\mu)$. We consider that $I(I_{\alpha,\beta,\alpha},\mu)$ also acts as an estimation for $I(I_{\alpha,\beta},\nu)$, considering ν as a future message received by α from β as the reciprocation to μ. Then, α estimates that after sending μ to β and receiving ν from β, $\mathcal{Y}_{\alpha,\beta} \approx I(I_{\alpha,\beta,\alpha},\mu) \oplus I(I_{\alpha,\beta,\alpha},\mu)$. This assumption is grounded on the *disclosure reciprocity* phenomenon Green et al. (2006). This phenomenon has been studied by psychologists and is based on the fact that one person's disclosure encourages the disclosure of the other person in the interaction, which in turn, encourages more disclosures from the first person.

Disclosing PDAs always comes at a privacy loss. α may estimate the privacy loss of sending μ to β using the privacy loss metric presented in Section 6.5. Then, α may choose to disclose a PDA that maximizes the estimation of the increase in intimacy while at the same time minimizing the privacy loss. We call this tradeoff the *privacy-intimacy* tradeoff. Let M be a tuple of messages that α sent to β, α will choose to disclose μ^* so that:

$$\mu^* \quad = \quad \arg\max_{\mu} I(I^{M}_{\alpha,\beta,\alpha},\mu) - \mathcal{L}(I^{M}_{\alpha,\beta,\alpha},\mu) \tag{6.3}$$

α will choose a message $\mu^* = \langle \alpha, \beta, \langle \alpha, a, P_a \rangle \rangle$ that maximizes the amount of information for the privacy-intimacy tradeoff. To model sincere agents when disclosing a PDA, μ^* must satisfy that KL$(P_a \parallel Q_a) = 0$ considering $Q_a \in I_\alpha$, i.e., $P_a = Q_a$ is the distribution that α has for a. To model agents that are not sincere when disclosing a PDA, μ^* must satisfy that KL$(P_a \parallel Q_a)$ matches the desired level of sincerity.

Balance

We assume that $I(I_{\alpha,\beta,\alpha},\mu)$ is an estimation for $I(I_{\alpha,\beta},\nu)$, considering ν as a future message received by α from β. However, this estimation may be bogus if β is not reliable when making claims about itself. This can be due to the fact that β is not sincere (β is not reliable on purpose) or β is unable to provide reliable information about itself. This could lead to $I(I_{\alpha,\beta,\alpha},\mu) >> I(I_{\alpha,\beta},\nu)$ when ν is actually received. Moreover, there could be agents that do not reciprocate disclosures because they are not willing to increase their intimacy to α for whatever reason (e.g. agents that are only interested in surveilling information about α).

α may assess to what extent β will reliably reciprocate future disclosures from α by considering the amount of information that β has sent to α and the amount of information that α has sent to β. To this aim, we use the concept of balance Sierra and Debenham (2007b).

Definition 10 *Given the UAIs $I_{\alpha,\beta}$ and $I_{\alpha,\beta,\alpha}$, a tuple of messages M from β to α and* *a tuple of messages M' from α to β, the balance between α and β from the point of view of α is:*

$$\mathcal{B}_{\alpha,\beta} = I(I_{\alpha,\beta}, M) - I(I_{\alpha,\beta,\alpha}, M')$$

α may use the balance $\mathcal{B}_{\alpha,\beta}$ as a trust model to assess to what extent β will reliably reciprocate future disclosures from α. Then, α may decide not to perform a disclosure to β if $\mathcal{B}_{\alpha,\beta} < \zeta$. In this case, ζ would act as a threshold of the minimum balance that α expects from its interaction partners. Moreover, α may specify a different ζ_β for each agent $\beta \in Ag$. In this way, α may even consider an increasing ζ_β as the intimacy $\mathcal{Y}_{\alpha,\beta}$ increases so that $\zeta'_\beta = \zeta_\beta + \lambda \cdot \mathcal{Y}_{\alpha,\beta}$, where λ is a normalizing constant. Using this dynamic ζ_β, we can model, for instance, that intimate partners can trust each other more than simple acquaintances can.

6.6.2. Privacy Fundamentalist Agents

Privacy fundamentalists are extremely concerned about privacy and very reluctant to disclose PDAs (Olson et al. (2005), Ackerman et al. (1999) and Westin (1967)). They feel like they have already lost much privacy and are not willing to lose privacy any more.

We model fundamentalist agents as pragmatist agents that establish a maximum total privacy loss ξ. In this way, a fundamentalist agent α considers the privacy-intimacy tradeoff to decide what PDA (if any) to disclose to β. α also considers the balance $\mathcal{B}_{\alpha,\beta}$ to assess to what extent β will reliably reciprocate future disclosures from α. Then, α may decide not to perform a disclosure to β if $\mathcal{B}_{\alpha,\beta} < \zeta$. The difference between pragmatists and fundamentalists is the following. If α is a fundamentalist agent, when the total privacy loss of α to β reaches ξ, α will not disclose PDAs to β any more.

Suppose that α has sent a sequence of messages $M = \{\mu_1, \dots, \mu_P\}$ to β. Then, let $\rho = min_\mu \mathcal{L}(I^M_{\alpha,\beta,\alpha}, \mu)$, i.e., ρ is the minimum privacy loss for α if she decides to disclose any PDA not yet disclosed to β. α will not disclose any other PDA to β if

$\rho + \mathcal{L}(I_{\alpha,\beta,\alpha}, M) > \xi$.

Moreover, α may specify a different ξ_β for each agent $\beta \in Ag$. In this way, α may consider an increasing ξ_β as the intimacy $\mathcal{Y}_{\alpha,\beta}$ increases so that $\xi'_\beta = \xi_\beta + \lambda \cdot \mathcal{Y}_{\alpha,\beta}$, where λ is a normalizing constant.

6.7. Implementation and Experimental Results

We implemented unconcerned, pragmatist and fundamentalist agents in Java. We implemented pragmatist and fundamentalist agents as agents that use the self-disclosure decision-making mechanisms explained in Section 6.6. We implemented unconcerned agents as agents that do not take into account privacy loss when disclosing PDAs to other agents. We considered unconcerned, pragmatist and fundamentalists to be sincere when disclosing a PDA.

We performed experiments in which unconcerned, pragmatist, and fundamentalist agents interact with other *target* agents. For each experiment, we calculated the intimacy that each agent achieved with each target agent and the total privacy that each agent lost with each target agent. The results for intimacy and privacy loss are given in bits because the unit of measure for information is the bit when base 2 logarithms are used to calculate entropies and KLs. Moreover, the results are the average of the results obtained when repeating each experiment 100 times.

The parameters used for the experiments are summarized in Table 6.1. The UAI I_α of each agent in each experiment was created as randomly generated distributions P_a for each PDA $a \in A$ over the domain V. The probability distributions in UAIs that each agent has modeling other agents and what other agents might know about it ($I_{\alpha,\beta}$ and $I_{\alpha,\beta,\alpha}$) are initialized to uniforms over V, i.e., agents are completely uncertain about the UAIs that other agents have at initialization time.

We implemented all agents as capable of making observations for the attributes of other agents. In this way, agents can call to an `observe()` method to obtain observed distributions for attributes of other agents. Then, agents estimate the reliability

Parameter	Description	Value
Nun	# Unconcerned	10
Npr	# Pragmatists	10
Nfu	# Fundamentalists	10
Nta	# Target Agents	30
A	Personal Data Attributes	$\{a_1,\ldots,a_{10}\}$
V	PDAs' Domain	$\{v_1,\ldots,v_{10}\}$
w	Subjective Sensitivity	Random $[0,1]^{10}$
ζ	Minimum Balance	-1
ξ	Fundamentalists' Threshold	1

Table 6.1: Parameters used in self-disclosure experiments.

of other agents using these observed values and the values other agents claimed for themselves (i.e. the disclosures they made) as inputs for the reliability model presented in section 6.2.

6.7.1. Sincere and Reciprocating Targets

In this section, we present the experiments that we performed comparing unconcerned, pragmatist, and fundamentalist agents when interacting with other target agents. These target agents reciprocate all of the disclosures they receive. Moreover, they perform such reciprocations in a sincere way, i.e., a target agent α reciprocates with a level of sincerity of KL$(Q'_a \parallel Q_a) = 0$ that means that Q'_a and Q_a are the same distribution, considering Q'_a as the distribution disclosed and Q_a as the distribution in I_α for the a attribute.

The experiments that we performed were composed of a number of disclosure rounds (DRs). In each DR, the agents were given the chance to choose an interaction partner and then to perform a disclosure (if any) to that particular interaction partner. We performed experiments varying the number of DRs ranging from 1 DR to 300 DRs. The maximum number of DRs is 300 because that is the number of DRs required by unconcerned, pragmatist, and fundamentalist agents to disclose all their PDAs to all of the target agents ($|A| \cdot Nta$).

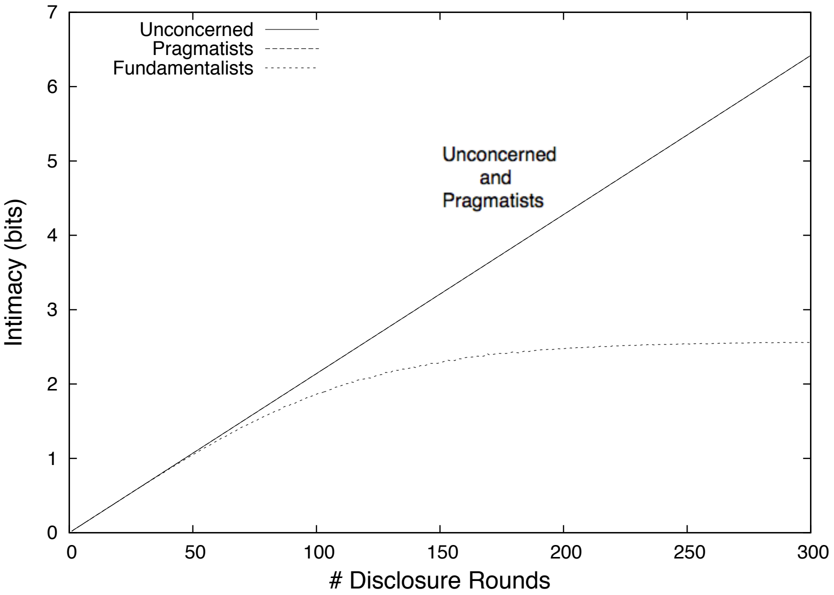

Figure 6.1: Intimacy considering sincere and reciprocating target agents.

Figure 6.1 shows the average intimacy achieved by the agents for each number of DRs considered. Both unconcerned and pragmatist agents achieve the same number of bits of intimacy for all of the experiments. Moreover, both unconcerned and pragmatist agents achieve more intimacy with target agents than fundamentalists. This is because when fundamentalists reach their maximum privacy loss ξ, they will no longer disclose PDAs so that intimacy is no longer increased.

Figure 6.2 shows the averaged privacy loss of the agents for each number of DRs. As expected, pragmatist agents lost less privacy than unconcerned agents for most of the experiments. For instance, for 16 DRs unconcerned agents lost 10 times more privacy than pragmatists; for 60 DRs unconcerned agents lost 5 times more privacy than pragmatists; for 130 DRs unconcerned lost 3 times more privacy than pragmatists; for 180 DRs unconcerned agents lost twice the privacy that pragmatists lost; and for 220 DRs unconcerned agents lost 1.5 times more privacy than pragmatists.

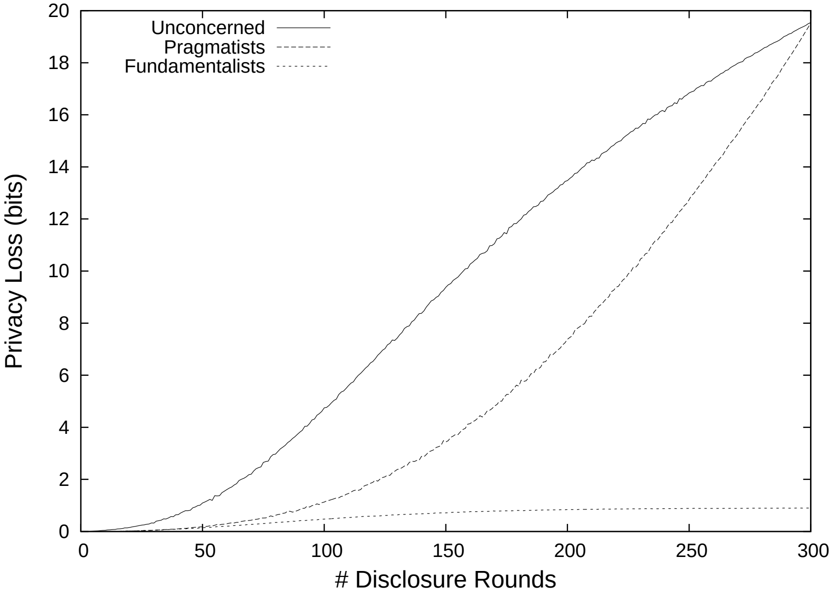

Figure 6.2: Privacy loss considering sincere and reciprocating target agents.

Therefore, for most of the experiments performed, pragmatist agents lost less privacy than unconcerned agents while achieving the same intimacy.

The privacy loss was similar for both pragmatist and unconcerned agents in the experiments with a high number of DRs (from 270 up to 300 DRs). This is because, in these experiments, the agents disclosed almost all of their PDAs to all of the agents, so that they ended up losing all their privacy regardless their privacy attitude.

As expected, pragmatist and fundamentalist agents lost less privacy than unconcerned agents. Moreover, fundamentalists lost less privacy than pragmatist agents. This is due to the fact that fundamentalists do not lose privacy beyond the threshold they define ξ.

6.7.2. Malicious Targets

In these experiments, we consider unconcerned, pragmatist, and fundamentalist agents interacting with target agents. For each experiment, we establish a number of malicious target agents (MTs) among the target agents. We consider malicious agents to be agents that are only interested in obtaining information from other agents without increasing intimacy. We model malicious agents as agents that either do not reciprocate or lie (are not sincere) about themselves. We implemented malicious agents such that when they receive a disclosure they do not reciprocate with a probability of 0.5. Moreover, when they reciprocate (the other 0.5 times) they are not sincere. We implemented malicious agents with a level of sincerity of 5 bits (recall that a level of sincerity of 0 means that an agent is completely sincere when disclosing her attributes). Thus, a malicious target agent α reciprocate with a level of sincerity of KL$(Q'_a \parallel Q_a) = 5$ considering Q'_a as the distribution disclosed and Q_a as the distribution in its AUI I_α for the a attribute. Therefore, there are 5 bits of difference between the distribution disclosed and the distribution in her AUI.

The parameters used for the experiments are the same as the ones in Table 6.1. We performed experiments varying the number of MTs from 0 up to 30. Thus, we model environments in which agents interact with a varying % of MTs among the target agents from 0% up to 100% (recall that the number of target agents Nta is set to 30). The number of DRs for all of the experiments is set to 200 DRs.

Figure 6.3 shows the average intimacy achieved by the agents for each number of MTs considered. As can be observed, all of the agents, regardless their privacy attitude, achieved less intimacy as the number of MTs increased. This is because as the number of MTs increases there are more target agents that do not reciprocate or do so with very unreliable information.

Pragmatists are able to achieve greater intimacy than unconcerned agents for 1 up to 18 MTs (from 3.3% up to 60% MTs). This is because pragmatists choose to interact with the most reliable and reciprocating agents, while unconcerned agents are not concerned about privacy and do not expect their disclosures to be reciprocated. From 19 MTs on, pragmatists achieved less intimacy than unconcerned agents. This is

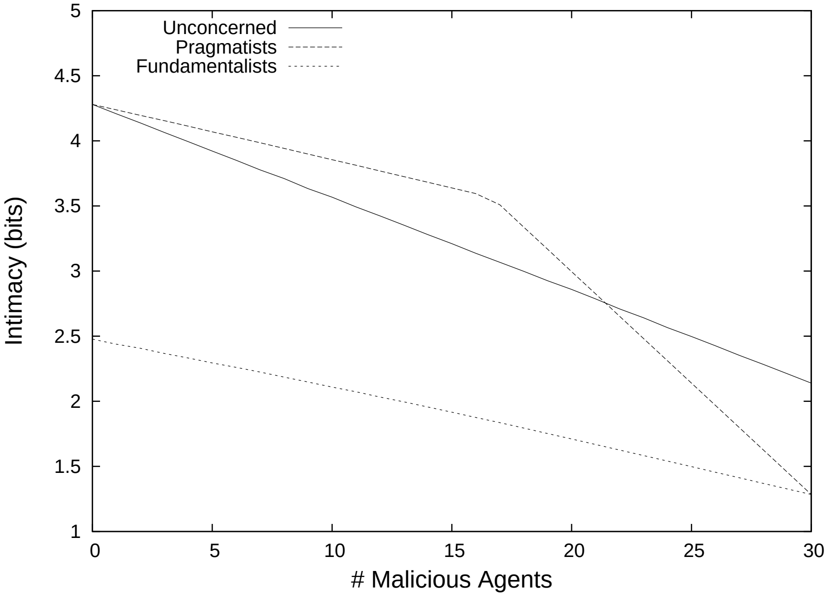

Figure 6.3: Intimacy considering malicious target agents.

because pragmatists will not disclose PDAs to MTs and there are not enough reliable and reciprocating agents in the system to achieve more intimacy.

As in the previous experiments, both unconcerned and pragmatist agents achieved more intimacy with other targets than fundamentalists. This is because when fundamentalists reach their maximum privacy loss ξ, they will no longer disclose PDAs so that intimacy is no longer increased.

Figure 6.4 shows the averaged privacy loss of the agents for each number of MTs considered. As expected, pragmatist and fundamentalist agents lost less privacy than unconcerned agents for all numbers of MTs. Unconcerned agents lost the same privacy, regardless of the number of MTs, because they always disclose one PDA to one agent for each DR without considering the privacy loss this may cause. Since the number of DRs was the same for the all of the experiments, unconcerned agents lost

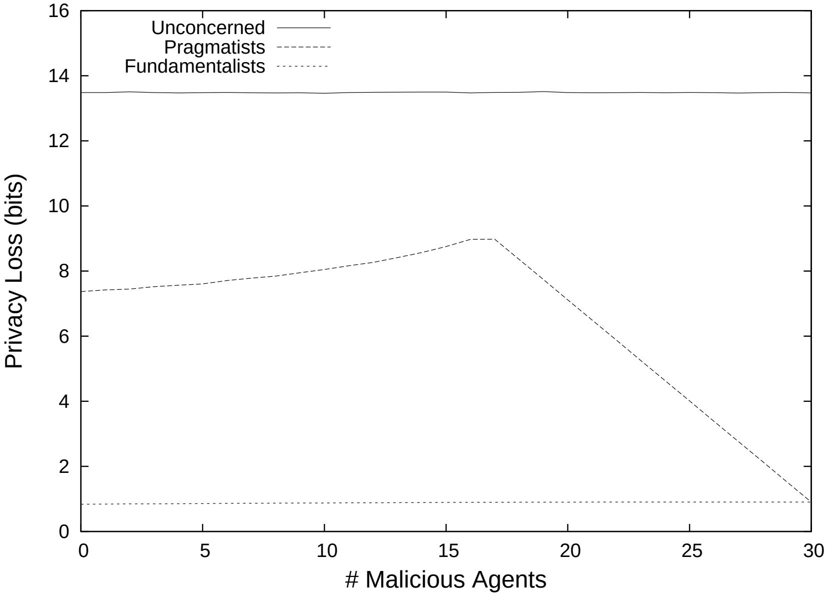

Figure 6.4: Privacy loss considering malicious target agents.

the same privacy in all of the experiments.

For 0 up to 18 MTs, pragmatists lost increasing amounts of privacy. This is due to the fact that as the number of MTs increases, pragmatists concentrate their disclosures to a decreasing number of reliable and reciprocating agents. Therefore, for a number of MTs near 18 MTs, they disclose more PDAs to reliable and reciprocating targets. Then, they end up disclosing more sensitive values, which causes a slight increase in the privacy loss.

For MTs from 18 up to 30, pragmatists lost less privacy as the number of MTs increased. This is because there are less reliable and reciprocating agents than MTs. Once pragmatists discover that an agent is malicious, they no longer disclose PDAs. As the number of MTs increases the number of total PDAs disclosed decreases so that privacy loss also decreases.

Fundamentalists lost less privacy than pragmatists and unconcerned agents. Moreover, fundamentalists lost the same privacy, regardless of the number of MTs. This is because when fundamentalists achieve their maximum privacy loss ξ, they will no longer disclose PDAs, regardless of whether or not targets are being malicious or reliable and reciprocating.

Finally, it is worth noting that for a number of MTs near 30 (100% MTs), both pragmatists and fundamentalists lost an order of magnitude less privacy than unconcerned agents.

6.8. Conclusions

In this chapter, we present self-disclosure decision-making mechanisms based on information measures. These self-disclosure decision-making mechanisms model pragmatic and fundamentalist attitudes towards privacy by considering the increase in intimacy and the loss of privacy a disclosure may cause. We adapt an existing intimacy measure and present a novel privacy loss measure. Both intimacy and privacy loss are based on uncertain agent identities, a formalism that we present to describe agents based on personal data attributes.

We experimentally show that pragmatists lose less privacy than unconcerned agents for the same intimacy. We also show that fundamentalists lose less privacy than both pragmatic and unconcerned agents but are unable to achieve the same intimacy. Moreover, in environments in which agents must interact with a percent of malicious agents less than or equal to 60%, pragmatists achieve even greater intimacy than unconcerned agents while losing less privacy. In environments in which agents must interact with a percent of malicious agents of almost 100%, both pragmatists and fundamentalists lose much less privacy than unconcerned agents.

As future work, we are exploring strategies for pragmatists and fundamentalists not to be sincere when disclosing a PDA. This could be useful once these agents detect that they are interacting with malicious agents. They could choose to keep on disclosing PDAs while being insincere instead of not disclosing any other PDA to such malicious agents. Thus, using such strategies agents would be able to lie to liars.

CHAPTER 7

Conclusions

Contents

7.1. General Conclusions

This thesis contributes to advance the state of the art in privacy and Multi-agent Systems. To our knowledge, privacy will be a matter of major concern and will receive many research efforts during this century. Multi-agent Systems can play a crucial role for preserving privacy. To this aim, Multi-agent Systems need to respect the privacy of the principals of the agents that act on behalf of them in the system. Moreover, agent-based solutions should be integrated into other information technologies to enhance the privacy. Thus, approaches based on agent technologies can further enhance privacy in other existing information technologies.

We specifically focus on avoiding undesired information collection and information processing in Multi-agent Systems. In order to avoid undesired information collection we propose a decision-making model for agents to decide whether or not to disclose personal information to other agents is acceptable or not. We propose a model based on psychological findings regarding how humans disclose personal information in the building of their relationships. This model considers intimacy on the one hand and privacy loss on the other hand. We also contribute a secure Agent Platform that allow agents to communicate with each other in a confidential fashion, i.e., external third parties cannot collect the information that two agents exchange.

In order to avoid undesired information processing, we propose an identity management model for agents in a Multi-agent System. This model avoids undesired information processing by allowing agents to hold as many identities as needed for minimizing data identifiability, i.e., the degree by which personal information can be directly attributed to a particular principal. This model also considers that agents should be able to selectively disclose parts (attributes) of their identity. Privacy is enhanced without compromising accountability and other crucial aspects for agents in a Multi-agent System, such as trust and reputation. To this aim, the model proposes a solution for the well-known identity-related vulnerabilities of trust and reputation models. Otherwise, these vulnerabilities can be exploited through whitewashing and sibyl attacks. Based on this model, we propose a software architecture that supports the development and execution of privacy-enhancing Multi-agent Systems in which trust and reputation play a crucial role. Our proposed architecture integrates an implementation of our privacy-enhancing agent identity management model into an existing agent platform.

7.2. Author's Related Scientific Publications

All these contributions are exclusively associated to the present PhD thesis and do not appear in any other PhD thesis. These contributions were published in: journals indexed in the Thomson's Science Citation Index (SCI[1]), conferences indexed by the Computing Research and Education Association of Australasia (CORE[2]), and other conferences.

7.2.1. Journals indexed in the SCI

Such, J. M., Espinosa, A., Garcia-Fornes, A. and Botti, V. (2011), *Partial identities as a foundation for Trust and Reputation*, Engineering Applications of Artificial Intelligence, Volume 24, Issue 7, pp. 1128 - 1136. DOI:10.1016/j.engappai.2011.06.008. JCR Impact Factor 1.444, Q1.

This publication describes our proposed privacy-enhancing agent identity model detailed in Chapter 4. We focus on describing the whole model and present a prototype implementation.

Such, J. M., Alberola, J. M., Espinosa, A. and Garcia-Fornes, A. (2011), *A group-oriented Secure Multiagent Platform*, Software: Practice and Experience, Volume 41, Issue 11, pp. 1289 - 1302. DOI:10.1002/spe.1042. JCR Impact Factor 0.667.

This publication describes our proposed secure agent platform detailed in Chapter 3, focusing on the privacy features it provides and the agent confinement to access only a subset of its principal's permissions.

[1] http://science.thomsonreuters.com/mjl/

[2] http://www.core.edu.au/

Such, J. M., Alberola, J. M., Barella, A. and Garcia-Fornes, A. (2011), *A Secure Group-Oriented Framework for Intelligent Virtual Environments*, Computing and Informatics, p. In Press. JCR Impact Factor 0.350.

This publication describes how our proposed secure agent platform detailed in Chapter 3 can be used to implement intelligent virtual environments.

7.2.2. Journals indexed in the SCI (Submitted)

Such, J. M., Espinosa, A., Garcia-Fornes, A. and Sierra, C. (2011), *Self-disclosure Decision Making based on Intimacy and Privacy*, Information Sciences, under review since 01/2011. JCR Impact Factor 2.833, Q1.

This publication describes our proposed decision-making model for agents to decide which personal information to disclose to other agents. This model is detailed in Chapter 6. It also includes the evaluation of the proposed model.

7.2.3. CORE Conferences

Such, J. M., Espinosa, A., Garcia-Fornes, A. and Sierra, C. (2011), Privacy-intimacy tradeoff in self-disclosure, in *10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2011)*, pp. 1189 - 1190. CORE A.

This publication describes our proposed decision-making model for agents to decide which personal information to disclose to other agents. This model is detailed in Chapter 6.

Such, J. M., (2011), Privacy and Self-disclosure in Multi-agent Systems, in *10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS2011)*, pp. 1333 - 1334. CORE A.

This publication includes how our proposed decision-making model described in Chapter 6 can be used by agents running in the infrastructures described in Chapter 3 and Chapter 5.

Bellver, J., **Such, J. M.**, Espinosa, A., Garcia-Fornes, A. (2011), Developing Secure Agent Infrastructures with Open Standards and Open-Source Technologies, in *9th International Conference on Practical Applications of Agents and MultiAgent Systems (PAAMS)*, pp. In press. CORE C.

This publication describes the secure infrastructure of Magentix2 detailed in Chapter 5.

Such, J. M., Alberola, J. M., Barella, A., Espinosa, A. and Garcia-Fornes, A. (2009), A secure group-oriented model for multiagent systems, in J. Cabestany, F. Sandoval, A. Prieto and J. Corchado, eds, *Bio-Inspired Systems: Computational and Ambient Intelligence*, Vol. 5517 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 522-529. CORE B.

This publication describes the secure group model provided by our proposed secure agent platform detailed in Chapter 3.

Such, J. M., Alberola, J. M., Garcia-Fornes, A., Espinosa, A. and Botti, V. (2008), Kerberos-based secure multiagent platform, in *Sixth International Workshop on Programming Multi-Agent Systems (ProMAS)*, pp. 173-186. CORE B.

This publication describes the use of Kerberos for authenticating agents and secure its communications. This is used as the default authentication, signing and ciphering suite in our proposed secure agent platform detailed in Chapter 3.

7.2.4. Other related Publications

Such, J. M., Espinosa, A. and Garcia-Fornes, A. (2011), An agent infrastructure for privacy-enhancing agent-based e-commerce applications, in *The Second International Workshop on Infrastructures and Tools for Multiagent Systems (ITMAS 2011)*, pp. 16-30.

This publication describes the support for privacy that we implemented in the Magentix2 AP and that is described in Chapter 5.

Such, J. M., Espinosa, A., Botti, V. and Garcia-Fornes, A. (2010), Trust and reputation through partial identities, in *The First International Workshop on Infrastructures and Tools for Multiagent Systems (ITMAS 2010)*, pp. 18-25.

This publication focuses on discussing a solution for the identity-related vulnerabilities of current trust and reputation models. This is one of the building blocks of our proposed privacy-enhancing agent identity model detailed in Chapter 4.

Such, J. M., Alberola, J. M., Garcia-Fornes, A., Espinosa, A. and Botti, V. (2009), Kerberos-based secure multiagent platform, in K. Hindriks, A. Pokahr and S. Sardina, eds, *Programming Multi-Agent Systems*, Vol. 5442 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 197-210.

This publication is the fully revised version of the “Kerberos-based secure multiagent platform” publication above. It appeared as a post-proceedings volume.

7.3. Future Work

In this section, we outline some of the most challenging possible future directions in the research field of privacy and MAS. This possible directions are open challenges identified during the realization of this thesis. We outline future directions including both privacy-enhancing studies for MAS and MAS for enhancing privacy. To our knowledge, all these open challenges will play a crucial role for agent technologies to be of broad use. On the one hand, principals can be more willing to engage with and delegate tasks to agents. On the other hand, agent technologies can be mixed/integrated with/into other information technologies to enhance the privacy that these information technologies provide.

7.3.1. Interoperability and Openness

As stated by Luck et al. (2005), interoperability is crucial for the medium-term development of MAS. Interoperability is a basic requirement for building open MAS, in which heterogeneous agents can enter and leave the MAS at any moment, can interact with each other, and can span organizational boundaries. For instance, agent-based e-marketplaces are open MAS (Fasli, 2007a), in which buyer, seller, and broker agents agents can be developed by different developers using different languages and frameworks, so heterogeneity is inherent. Thus, agents and their interaction protocols need to allow interoperation. Thus, standards that help to allow interoperation are of crucial importance.

Although there are some standards proposed for agents and their interaction protocols, as yet there is no standard focusing on privacy issues. As described in section 2.2.2, a first requirement for privacy is security. There have been some standards for security in MAS. FIPA defined a standard for security in MAS, but this standard soon became obsolete (FIPA, 1998). There have been some studies that consider this obsolete standard as a basis to analyze and propose guidelines for FIPA-based security standards (Poslad and Calisti, 2000; Poslad et al., 2003). However, there has not been another proposal for a security standard for FIPA platforms since the obsolete standard from 1998.

The OMG Mobile Agent System Interoperability Facility (MASIF) (Milojicic et al., 1998) is a standard for mobile agents. Mobile agents are a species of autonomous agents that are capable of transmitting (migrating) themselves – their program and their state – across a computer network, and recommencing execution at a remote site (Wooldridge, 2002). MASIF specifies security mechnisms for mobile agents to migrate among hosts and also secure communication mechanisms. However, the security of MASIF is dependent on Corba-IDL mechanisms. No other transport mechanisms are considered, such as HTTP, AMQP, and others.

Security standards play a crucial role in preserving confidentiality in agent interactions. However, there are many other mechanisms that are needed for preserving privacy that also need to be standardized. For instance: how can agents selectively disclose parts of their identities in a standard way (as required by the disclosure decision making mechanisms in section 2.2.1)? How can agents change the pseudonyms they use in a standard way (as required for pseudonym management technologies in section 2.3.2)? These standards do not need to be built from scratch. Instead, existing standards can be used as the basis for them. For instance, the OASIS Identity Metasystem Interoperability standard [3] is a standard for mechanisms that support pseudonymity and the selective disclosure of identity attributes.

7.3.2. Pseudonym changer Agent

According to Hansen et al. (2008), one of the main questions that is relevant for pseudonyms to be privacy-preserving is the amount of information that can be gathered by linking data disclosed under the same pseudonym. Social security numbers in the USA are a clear example that if a pseudonym is used for a long time, even spanning different contexts, different pieces of personal information disclosed in different contexts can be linked to each other, and also allow the inference of other personal information emerging from the combination of data in different contexts and applying learning and inference techniques. Moreover, linking a pseudonym to the real identity

[3] http://docs.oasis-open.org/imi/identity/v1.0/identity.html

only once is sufficient to be able to associate all of this personal information to a real identity. This link can remain over time. Thus, other personal information disclosed under this pseudonym can be linked to the real identity of the principal subject of this personal information.

Clauβ et al. (2005) points out that different pseudonyms should be used in different contexts if the principal wants to maintain the personal information disclosed under each of these pseudonyms unlinkable. The most privacy-preserving option is to use transaction pseudonyms, which treat each transaction as a different context. However, there are many cases in which the principal can be interested in reusing the same pseudonym, e.g., a social network that focus on a specific topic in which the principal is willing to establish friendships and other kinds of relationships that need the reuse of pseudonyms for recognizing entities from one time to another. Another example is that the principal itself could be willing to provide his/her profile to the a seller agent in an e-commerce scenario in exchange for a discount or a reward, as pointed out in section 2.2.1.

We mentioned several approaches for pseudonym support in Section 2.3.2. However, we could not find any approach that *suggests* pseudonym changes. In other words, the needed mechanisms for agents to be able to change their pseudonym exist, but there is no study or proposal for agents to decide when to change their pseudonym. This responsibility is given to the agent's designer or agent's principal. We envision pseudonym changer agents. These agents would be in charge of suggesting pseudonym changes by evaluating the privacy risks of reusing a pseudonym. Moreover, the models detailed in section 2.2.1 could also be applied to make the decision of whether or not a pseudonym change is appropriate. This decision would take into account: the privacy risks due to not changing the pseudonym, and the utility or intimacy that would be lost by changing the pseudonym.

7.3.3. Disclosure Decision Making based on Multiple Criteria

As stated in section 2.2.1, current disclosure decision-making mechanisms are based on policies, the privacy-utility tradeoff, or social relationships. However, there is no proposal that brings these mechanisms together. This could be very appropriate for situations in which each one of these mechanisms is not enough by itself to cope with the requirements of agents' principals.

There are many examples of environments and situations in which these disclosure decision-making mechanisms can be combined. For instance, in a controlled environment in which policies are known to be enforced, the policies themselves can be used for agents that disclose personal information based on the privacy-utility tradeoff. In this way, agents are able to valuate the privacy loss they will suffer in the event of disclosing a specific piece of data, according to the policy of the intended destination agent. Then, based on this estimated privacy loss they can determine if the expected benefit for disclosing the information is worth it.

Another example could be the combination of the privacy-utility tradeoff with other more social approaches. For instance, suppose that an agent knows the benefit that disclosing personal information to another agent may cause to itself. Also suppose that this benefit is not worth the disclosure according to the privacy-utility tradeoff. An agent can still decide to disclose this information if it has a relationship that is intimate enough with the intended destination agent. Moreover, the agent could also decide to disclose this information if it does not have a relationship that is intimate enough with the intended destination agent, but it wants to reciprocate a previous disclosure of the intended destination agent.

Now, suppose that an agent has very low intimacy with another agent. Moreover, suppose that this low level of intimacy is due to the fact that the second agent deceived the first agent by not reciprocating its disclosures. The question arises as to whether or not the first agent should disclose personal information to the second agent if the first agent knows the utilitarian benefit of doing so and this benefit is high enough. In other words, how could the agent decide which of the two mechanisms to follow in a given situation?

7.3.4. Learning the privacy sensitivity of personal information

Most of the approaches presented in this thesis assume that agents know the real privacy sensitivity of each personal attribute of their principals. However, this assumption is not always realistic. For instance, the number of personal attributes can be very large, so principals may not feel comfortable specifying the sensitivity for each of their personal attributes. Some of the approaches try to minimize this burden by clustering attributes into categories so that principals specify the sensitivity for each category (Yassine et al., 2010). However, this can be also a burden if there is a huge number of categories or if the categories must be defined per target agent and there is a huge number of possible target agents.

A possible future line of research could be to automatically learn the privacy sensitivity of personal information based on studies such as the one presented by Huberman et al. (2005). They carried out an experiment that validates that people are usually more willing to disclose certain private attributes that are typical or positively atypical compared to the target group. The experiment assesses the value (in terms of monetary compensation) that people give to disclose personal attributes like weight and age. They gathered interesting results regarding weight (for age the significance was less): people who weigh less than the average required little compensation to disclose their weight, while people who weigh more than the average required a large compensation to disclose their weight. This is due to the fact that people who weigh more are afraid to feel embarrassment or stigmatization. The authors found a linear relationship about a trait and the value one places on it. The less desirable the trait, the more reluctant a person is to disclose the information. However, small deviations in a socially positive direction are associated with a lower monetary compensation request.

7.3.5. Personal Data Attribute Inference

The decision-making models presented in Section 2.2.1 consider the privacy loss of disclosing a personal attribute before deciding whether they finally disclose it or not. This privacy loss usually considers the sensitivity of the personal attribute to be disclosed and the probability of linking this personal attribute to the real identity of the principal behind the agent. Although the agent can decide whether or not to disclose each attribute, it cannot control that other agents can infer other attributes that it does not want to disclose. This is known as the inference problem (Farkas and Jajodia, 2002). For instance, in the USA, if a principal discloses its driver license number, she/he is also disclosing that she/he is, at least, 16 years old.

Only a few of the decision-making models consider what could be inferred due to the disclosure of a personal attribute. Moreover, the decision-making models that consider these inferences provide very simple inference models. Several approaches tackle this problem in different computer science disciplines. These approaches are intended to infer the probabilities of linking personal data attributes to each other and to the principal they describe. For instance, there are approaches that deal with the inference problem when querying databases (Cuenca Grau and Horrocks, 2008), when applying data mining techniques (Zhu et al., 2009), in social networks (Zheleva and Getoor, 2009), and in general, in all activities that require the publication of data (Chen et al., 2009). All of these approaches consider complex models of personal information inference. The disclosure decision-making mechanisms for agents can either be based on these models or they can be adapted for the case of agents.

7.3.6. Information dissemination detection

As shown in section 2.4, there are few studies that focus on information dissemination. Although these studies solve some of the problems that must be dealt with for protecting information dissemination, there are still many challenges that remain open. One such open challenge involves mechanisms for agents to detect when other agents disseminate information about them.

Sierra and Debenham (2008) propose an approach for an agent to detect that another agent is disseminating information about it based on scanning all the information the first agent receives in the search for clues of possible information disseminations. However, an agent may not be able to detect by itself that other agents are disseminating information about it. Another approach for information dissemination detection is based on notifications sent by other agents warning of possible dissemination of information. These notifications can play a crucial role when an agent itself is unable to detect that other agents are disseminating information about it. Krupa and Vercouter (2010) use notifications of disseminations in the form of what they call punishment messages. These messages are sent by the agent that detects an inappropriate dissemination to the rest of the agents, so that agents can know which agents perform inappropriate information disseminations. However, this mechanism can be subject to strategic manipulation, such as agents sending messages containing fake norm violations that do not really correspond to real violations.

7.3.7. Integration of trust, reputation, and norms for protecting against information dissemination

The real connection of norm-based approaches to trust-based approaches for avoiding information dissemination needs to be specified. This open challenge is also closely related to the information dissemination detection problem. If an agent is able to detect that another agent has performed information dissemination, it could revise the trust the first agent has in the second agent. Moreover, an agent could earn a very bad reputation with other agents by performing undesired information dissemination. In this way, both trust and reputation would act as privacy-enforcer mechanisms, isolating agents that disseminate information in an inappropriate way.

Krupa and Vercouter (2010) suggest that messages informing of agents that violate the corresponding information dissemination norms can be used as inputs for trust and reputation models. Therefore, agents that do not abide by the norms would be considered as untrustworthy and would earn a bad reputation. This would finally result in the isolation of agents that do not abide by the norms. However, the authors of this work do not discuss how this integration of trust, reputation, and norms can be effectively achieved.

7.3.8. Avoiding collusion for protecting information dissemination

As shown in section 2.4, current norm-based approaches to information dissemination are vulnerable to collusion. Thus, two or more agents could easily collude by passing information to each other without other benevolent and norm-abiding agents being aware of it. This could be addressed by using a central authority that would control and monitor the information that agents exchange. However this may not be possible for various reasons. The main one, in line with this article, is privacy preservation. This is because this authority would become a *big brother*. Moreover, there may be other reasons, such as to prevent this authority from becoming a performance bottleneck and a single point of failure (SPOF).

Moreover, the problem of collusion could even be worse if we consider collusion in which one agent decides to disseminate information but without revealing the source of the information. Krupa and Vercouter (2010) identified this problem and called the agent that disseminates the information “journalist”. As they state, a journalist agent would be an agent that decides to sacrifice himself to become a relay for information that violates the information dissemination norms. Therefore, the agent that is the source of the violation will never be known, and only the journalist will be seen as a violator of these norms. A journalist agent could even be rewarded with a monetary benefit in exchange for its practices.

7.3.9. Protection against information collection and dissemination

The combination/mixing of disclosure decision-making models with information dissemination models can play a crucial role preventing both information collection and information dissemination. All of the disclosure decision-making models presented in section 2.2.1 assume separate interactions among agents for evaluating the privacy cost that a disclosure may cause. That is, these models do not consider that the agent that received the disclosure can share the received information with other agents.

An illustrative example can be: agent A decides by means of a disclosure decision-making models not to disclose its attribute location to agent B, e.g., the expected utilitarian benefit for agent A to disclose its location to agent B is not high enough compared to the privacy loss this disclose may cause to agent A. However, in a different interaction, agent A decides, by means of a disclosure decision-making model, to disclose its attribute location to agent C, e.g., the expected utilitarian benefit for agent A to disclose its location to agent C is high enough compared to the privacy loss. After this, agent A effectively disclose its location to agent C. Then, suppose that agent B and agent C are known to each other, so agent C can finally disclose the location of agent A to agent B. Therefore, B finally knows the location of agent A, even though agent A decided not to disclose it directly to agent B. Thus, we consider that if the dissemination risk is known, it should be considered when deciding whether to disclose information because if information collection is prevented, information dissemination cannot occur.

Another example can be: agent B has a bad reputation in a society because it usually disseminates the information it receives about other agents. Therefore, other agents in the society can decide to avoid disclosing information to B. However, suppose that an agent A and agent B have a very close relationship, i.e., they have a medium/high degree of intimacy. Then, suppose that agent A has to decide whether to disclose its location to agent B. In this particular case, agent A could consider that the intimacy it has with agent B is high enough to assume that if it discloses new information to agent B, agent B will not disseminate it to other agents.

Bibliography

Ackerman, M. S., Cranor, L. F., and Reagle, J. (1999). Privacy in e-commerce: examining user scenarios and privacy preferences. In *Proceedings of the 1st ACM conference on Electronic commerce (EC)*, pages 1–8, New York, NY, USA. ACM.

Acquisti, A., Gritzalis, S., Lambrinoudakis, C., and di Vimercati, S., editors (2008). *Digital Privacy: Theory, Technologies, and Practices*. Auerbach Publications.

Aïmeur, E., Brassard, G., Fernandez, J. M., and Onana, F. S. M. (2006). Privacy-preserving demographic filtering. In *Proceedings of the ACM symposium on Applied computing (SAC)*, pages 872–878, New York, NY, USA. ACM.

Aïmeur, E., Brassard, G., and Onana, F. (2006). Secure anonymous physical delivery. *IADIS International Journal on WWW/Internet*, 4(1):55–59.

Alberola, J. M., Such, J. M., Espinosa, A., Botti, V., and Garcia-Fornes, A. (2008). Scalable and efficient multiagent platform closer to the operating system. *Artificial Intelligence Research and Development*, 184:7–15.

Alberola, J. M., Such, J. M., Garcia-Fornes, A., Espinosa, A., and Botti, V. (2010). A performance evaluation of three multiagent platforms. *Artificial Intelligence Review*, 34:145–176.

Aydoğan, R. and Yolum, P. (2010). Learning opponent's preferences for effective negotiation: an approach based on concept learning. *Autonomous Agents and Multi-Agent Systems*, pages 1–37. 10.1007/s10458-010-9147-0.

Baldoni, M., Boella, G., Genovese, V., Mugnaini, A., Grenna, R., and van der Torre, L. (2010). A middleware for modeling organizations and roles in jade. In Braubach, L., Briot, J.-P., and Thangarajah, J., editors, *Programming Multi-Agent Systems*, volume 5919 of *LCNS*, pages 100–117. Springer Berlin / Heidelberg.

Balke, T. and Eymann, T. (2008). The conclusion of contracts by software agents in the eyes of the law. In *Proc. of The 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 771–778. IFAAMAS.

Barella, A., Carrascosa, C., and Botti, V. (2006). Jgomas: game-oriented multi-agent system based on jade. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology (ACE)*, page 17, New York, NY, USA. ACM.

Barth, A., Datta, A., Mitchell, J. C., and Nissenbaum, H. (2006). Privacy and contextual integrity: Framework and applications. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 184–198, Washington, DC, USA. IEEE Computer Society.

Bauer, M. (2006). Paranoid penguin: an introduction to novell apparmor. *Linux J.*, 2006(148):13.

Bhargav-Spantzel, A., Camenisch, J., Gross, T., and Sommer, D. (2007). User centricity: A taxonomy and open issues. *J. Comput. Secur.*, 15:493–527.

Bilge, L., Strufe, T., Balzarotti, D., and Kirda, E. (2009). All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web (WWW)*, pages 551–560, New York, NY, USA. ACM.

Bishop, M. (2002). *Computer Security: Art and Science*. Addison-Wesley.

Boella, G., van der Torre, L., and Verhagen, H. (2006). Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12:71–79.

Borking, J., Van Eck, B., Siepel, P., and Bedrijf, D. (1999). *Intelligent software agents: Turning a privacy threat into a privacy protector*. Registratiekamer, The Hague.

Brazier, F., Oskamp, A., Prins, C., Schellekens, M., and Wijngaards, N. (2004). Anonymity and software agents: An interdisciplinary challenge. *Artificial Intelligence and Law*, 12:137–157.

Brito, I. and Meseguer, P. (2003). Distributed forward checking. In Rossi, F., editor, *Principles and Practice of Constraint Programming*, volume 2833 of *LNCS*, pages 801–806. Springer Berlin / Heidelberg.

Bygrave, L. (2001). Electronic agents and privacy: A cyberspace odyssey 2001. *International Journal of Law and Information Technology*, 9(3):275–294.

Camp, L. J. (1996). *Privacy & Reliability in Internet Commerce*. PhD thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

Carrara, E. and Hogben, G. (2007). Reputation-based systems: a security analysis. ENISA Position Paper.

Castelfranchi, C. and Falcone, R. (1998). Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Proceedings of the 3rd International Conference on Multi Agent Systems (ICMAS)*, page 72, Washington, DC, USA. IEEE Computer Society.

Chaum, D. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24:84–90.

Chaum, D. (1985). Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28:1030–1044.

Chaum, D., Fiat, A., and Naor, M. (1990). Untraceable electronic cash. In *Proceedings on Advances in cryptology(CRYPTO)*, pages 319–327, New York, NY, USA. Springer-Verlag New York, Inc.

Chen, B.-C., Kifer, D., LeFevre, K., and Machanavajjhala, A. (2009). Privacy-preserving data publishing. *Foundations and Trends in Databases*, 2(1–2):1–167.

Cheng, A. and Friedman, E. (2005). Sybilproof reputation mechanisms. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, P2PECON '05, pages 128–132, New York, NY, USA. ACM.

Chopra, S. and White, L. (2004). Artificial agents - personhood in law and philosophy. In *Proceedings of The 13th European Conference on Artificial Intelligence (ECAI)*, pages 635–639. IOS press,.

Chopra, S. and White, L. (2007). Privacy and artificial agents, or, is google reading my email? In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI)*, pages 1245–1250, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Cissée, R. (2003). An architecture for agent-based privacy-preserving information filtering. In *Sixth International Workshop on Trust, Privacy, Deception and Fraud in Agent Systems.*, pages 1–10.

Cissée, R. and Albayrak, S. (2007). An agent-based approach for privacy-preserving recommender systems. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 182:1–182:8, New York, NY, USA. ACM.

Clauβ, S., Kesdogan, D., and Kölsch, T. (2005). Privacy enhancing identity management: protection against re-identification and profiling. In *Proceedings of the workshop on Digital identity management (DIM)*, pages 84–93, New York, NY, USA. ACM.

Crépin, L., Demazeau, Y., Boissier, O., and Jacquenet, F. (2009). Sensitive data transaction in hippocratic multi-agent systems. In Artikis, A., Picard, G., and Vercouter, L., editors, *Engineering Societies in the Agents World IX*, volume 5485 of *LCNS*, pages 85–101. Springer-Verlag, Berlin, Heidelberg.

Crépin, L., Vercouter, L., Boissier, O., Demazeau, Y., and Jacquenet, F. (2008). Hippocratic multi-agent systems. In *Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS)*, pages 301–307.

Criado, N., Argente, E., and Botti, V. (2011). Open issues for normative multi-agent systems. *AI Communications*, pages In Press. DOI:10.3233/AIC–2011–0502.

Cuenca Grau, B. and Horrocks, I. (2008). Privacy-preserving query answering in logic-based information systems. In *Proceeding of the 18th European Conference on Artificial Intelligence (ECAI)*, pages 40–44, Amsterdam, The Netherlands. IOS Press.

Diaz, C. (2006). Anonymity metrics revisited. In Dolev, S., Ostrovsky, R., and Pfitzmann, A., editors, *Anonymous Communication and its Applications*, number 05411 in Dagstuhl Seminar Proceedings. IBFI, Schloss Dagstuhl, Germany.

Dierks, T. and Allen, C. (1999). The tls protocol version 1.0. RFC 2246.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: the second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium*, pages 21–21.

Farkas, C. and Jajodia, S. (2002). The inference problem: a survey. *ACM SIGKDD Explorations Newsletter*, 4(2):11.

Fasli, M. (2007a). *Agent Technology For E-Commerce*. John Wiley & Sons.

Fasli, M. (2007b). On agent technology for e-commerce: trust, security and legal issues. *Knowledge Engineering Review*, 22(1):3–35.

FIPA (1998). *FIPA Agent Security Management*. FIPA.

FIPA (2001a). *FIPA Abstract Architecture Specification*. http://www.fipa.org/specs/fipa00001/.

FIPA (2001b). *FIPA ACL Message Structure Specification*. FIPA.

Fischer-Hübner, S. and Hedbom, H. (2008). Benefits of privacy-enhancing identity management. *Asia-Pacific Business Review*, 10(4):36–52.

Friedman, E. J. and Resnick, P. (1998). The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199.

Frier, A., Karlton, P., and Kocher, P. (1996). The secure socket layer. Technical Report MSU-CSE-00-2, Netscape Communications.

Gambetta, D., editor (1990). *Trust: Making and Breaking Cooperative Relations*. Basil Blackwell.

Gangopadhyay, A., editor (2001). *Managing Business with Electronic Commerce: Issues and Trends*. IGI Publishing, Hershey, PA, USA.

Garfinkel, S. (2001). *Database nation: the death of privacy in the 21st century*. O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Garfinkel, S. (2009). Privacy requires security, not abstinence: Protecting an inalienable right in the age of facebook. `http://www.technologyreview.com/computing/22831/`.

Gibbins, N., Harris, S., and Shadbolt, N. (2004). Agent-based semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(2):141–154.

Goldschlag, D., Reed, M., and Syverson, P. (1999). Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42:39–41.

Green, K., Derlega, V. J., and Mathews, A. (2006). *The Cambridge Handbook of Personal Relationships*, chapter Self-Disclosure in Personal Relationships, pages 409–427. Cambridge University Press.

Greenstadt, R., Grosz, B., and Smith, M. D. (2007). Ssdpop: improving the privacy of dcop with secret sharing. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 171:1–171:3, New York, NY, USA. ACM.

Greenstadt, R., Pearce, J. P., and Tambe, M. (2006). Analysis of privacy loss in distributed constraint optimization. In *Proceedings of the 21st national conference on Artificial intelligence (AAAI)*, pages 647–653. AAAI Press.

Gutknecht, O. and Ferber, J. (2000). The madkit agent platform architecture. *Lecture Notes In Computer Science*, 1887:48–55.

Hansen, M., Berlich, P., Camenisch, J., Clauß, S., Pfitzmann, A., and Waidner, M. (2004). Privacy-enhancing identity management. *Information Security Technical Report*, 9(1):35 – 44.

Hansen, M., Schwartz, A., and Cooper, A. (2008). Privacy and identity management. *IEEE Security & Privacy*, 6(2):38–45.

He, J., Chu, W., and Liu, Z. (2006). Inferring privacy information from social networks. In *Intelligence and Security Informatics*, volume 3975 of *LNCS*, chapter 14, pages 154–165. Springer-Verlag.

Head, M. and Yuan, Y. (2001). Privacy protection in electronic commerce–a theoretical framework. *Human Systems Management*, 20(2):149–160.

Hildebrandt, M. and Gutwirth, S. (2008). *Profiling the European Citizen: Cross-Disciplinary Perspectives*. Springer Publishing Company, Inc.

Hoffman, D., Novak, T., and Peralta, M. (1999). Building consumer trust online. *Communications of the ACM*, 42(4):80–85.

Hoffman, K., Zage, D., and Nita-Rotaru, C. (2009). A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42:1:1–1:31.

Horling, B. and Lesser, V. (2004). A survey of multiagent organizational paradigms. *The Knowledge Engineering Review*, 19:281–316.

Huberman, B. A., Adar, E., and Fine, L. R. (2005). Valuating privacy. *IEEE Security and Privacy*, 3(5):22–25.

Huynh, T. D., Jennings, N. R., and Shadbolt, N. R. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154.

Ismail, L. (2008). A secure mobile agents platform. *Journal of Communications*, 3(2):1–12.

JADE Board (2005). Jade security guide. `http://jade.tilab.com`.

Jaiswal, A., Kim, Y., and Gini, M. L. (2004). Design and implementation of a secure multi-agent marketplace. *Electronic Commerce Research and Applications*, 3(4):355–368.

Jøsang, A. and Golbeck, J. (2009). Challenges for Robust Trust and Reputation Systems. In *Proceedings of the 5th International Workshop on Security and Trust Management (STM)*, pages 1–12.

Jøsang, A., Ismail, R., and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decis. Support Syst.*, 43(2):618–644.

Jøsang, A., Patton, M., and Ho, A. (2001). Authentication for Humans. In *Proceedings of the 9th International Conference on Telecommunication Systems (ICTS)*, pages 1–10.

Kerr, R. and Cohen, R. (2009). Smart cheaters do prosper: defeating trust and reputation systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 993–1000. IFAAMAS.

Klir, G. J. (2006). *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley.

Koops, B.-J. and Leenes, R. (2006). Identity theft, identity fraud and/or identity-related crime. *Datenschutz und Datensicherheit - DuD*, 30:553–556.

Korba, L., Song, R., and Yee, G. (2002). Anonymous communications for mobile agents. In *Proceedings of the 4th International Workshop on Mobile Agents for Telecommunication Applications (MATA)*, pages 171–181. Springer.

Krause, A. and Horvitz, E. (2008). A utility-theoretic approach to privacy and personalization. In *Proceedings of the 23rd national conference on Artificial intelligence (AAAI)*, pages 1181–1188. AAAI Press.

Krupa, Y. and Vercouter, L. (2010). Contextual Integrity and Privacy Enforcing Norms for Virtual Communities. In *Proceedings of the 11th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems (COIN@MALLOW2010)*, pages 150–165.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86.

Lebanon, G., Scannapieco, M., Fouad, M., and Bertino, E. (2006). Beyond k-anonymity: A decision theoretic framework for assessing privacy risk. In Domingo-Ferrer, J. and Franconi, L., editors, *Privacy in Statistical Databases*, volume 4302 of *LNCS*, pages 217–232. Springer Berlin / Heidelberg.

Lee, H.-H. and Stamp, M. (2008). An agent-based privacy-enhancing model. *Inf. Manag. Comput. Security*, 16(3):305–319.

Li, N., Li, T., and Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 106 –115, Los Alamitos, CA, USA. IEEE Computer Society.

Longstaff, T., Ellis, J., Shawn, H., Lipson, H., Mcmillan, R., Pesante, H., L., and Simmel, D. (1997). Security of the internet. *The Froehlich/Kent Encyclopedia of Telecommunications*, 15:231–255.

Luck, M., McBurney, P., Shehory, O., and Willmott, S. (2005). *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink.

Maheswaran, R., Pearce, J., Bowring, E., Varakantham, P., and Tambe, M. (2006). Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *Autonomous Agents and Multi-Agent Systems*, 13:27–60.

Menczer, F., Street, W. N., Vishwakarma, N., Monge, A. E., and Jakobsson, M. (2002). Intellishopper: a proactive, personal, private shopping assistant. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 1001–1008, New York, NY, USA. ACM.

Miller, R., Perlman, D., and Brehm, S. (2007). *Intimate relationships*. McGraw-Hill Higher Education.

Milojicic, D., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B., Kosaka, K., Lange, D., Ono, K., Oshima, M., Tham, C., Virdhagriswaran, S., and White, J. (1998). Masif: The omg mobile agent system interoperability facility. *Personal and Ubiquitous Computing*, 2:117–129. 10.1007/BF01324942.

Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM*, 37(7):80–91.

Montaner, M., López, B., and De La Rosa, J. (2003). A taxonomy of recommender agents on the internet. *Artificial intelligence review*, 19(4):285–330.

Mulet, L., Such, J. M., and Alberola, J. M. (2006). Performance evaluation of open-source multiagent platforms. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1107–1109, New York, NY, USA. ACM.

Neuman, C., Yu, T., Hartman, S., and Raeburn, K. (2005). *The Kerberos Network Authentication Service (V5)*. Number 4120 in Request for Comments. IETF.

Newman, A. E. (2004). Cougaar developers' guide. `http://www.cougaar.org`.

Nissenbaum, H. (2004). Privacy as contextual integrity. *Washington Law Review*, 79(1).

Odlyzko, A. (2003). Privacy, economics, and price discrimination on the internet. In *Proceedings of the 5th international conference on Electronic commerce (ICEC)*, pages 355–366, New York, NY, USA. ACM.

Olson, J. S., Grudin, J., and Horvitz, E. (2005). A study of preferences for sharing and privacy. In *Proceedings of the SIGCHI Conference on Human factors in computing systems*, pages 1985–1988, New York, NY, USA. ACM.

Palen, L. and Dourish, P. (2003). Unpacking privacy for a networked world. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 129–136, New York, NY, USA. ACM.

Petkovic, M. and Jonker, W., editors (2007). *Security, Privacy and Trust in Modern Data Management (Data-Centric Systems and Applications)*. Springer-Verlag.

Pfitzmann, A. and Hansen, M. (2010). A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. http://dud.inf.tu-dresden.de/Anon_Terminology.shtml. v0.34.

Piolle, G., Demazeau, Y., and Caelen, J. (2007). Privacy management in user-centred multi-agent systems. In O'Hare, G., Ricci, A., O'Grady, M., and Dikenelli, O., editors, *Engineering Societies in the Agents World VII*, volume 4457 of *LNCS*, pages 354–367. Springer Berlin / Heidelberg.

Poslad, S. and Calisti, M. (2000). Towards improved trust and security in FIPA agent platforms. In *Workshop on Deception, Fraud and Trust in Agent Societies*, pages 1–4.

Poslad, S., Charlton, P., and Calisti, M. (2003). Specifying standard security mechanisms in multi-agent systems. In Falcone, R., Barber, S., Korba, L., and Singh, M., editors, *Trust, Reputation, and Security: Theories and Practice*, volume 2631 of *LNCS*, pages 227–237. Springer Berlin / Heidelberg.

Quillinan, T. B., Warnier, M., Oey, M., Timmer, R., and Brazier, F. (2008). Enforcing security in the agentscape middleware. In *Proceedings of the workshop on Middleware security (MidSec)*, pages 25–30, New York, NY, USA. ACM.

Ramchurn, S., Huynh, D., and Jennings, N. (2004). Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25.

Rannenberg, K., Royer, D., and Deuker, A., editors (2009). *The Future of Identity in the Information Society: Challenges and Opportunities*. Springer Publishing Company, Incorporated.

Recursion Software Inc. (2008). Voyager security guide. `http://www.recursionsw.com/`.

Resnick, P. and Zeckhauser, R. (2002). Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In Baye, M. R., editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Elsevier Science.

Roth, V. and Jalali-Sohi, M. (2001). Concepts and architecture of a security-centric mobile agent server. In *Proceedings of the Fifth International Symposium on Autonomous Decentralized Systems (ISADS)*, pages 435–444, Washington, DC, USA. IEEE Computer Society.

Sabater, J. and Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, 24:33–60.

Sabater-Mir, J., Paolucci, M., and Conte, R. (2006). Repage: REPutation and imAGE among limited autonomous partners. *Journal of Artificial Societies and Social Simulation*, 9(2).

Schermer, B. (2007). *Software agents, surveillance, and the right to privacy: a legislative framework for agent-enabled surveillance*. Amsterdam Univ Pr.

Senicar, V., Jerman-Blazic, B., and Klobucar, T. (2003). Privacy-enhancing technologies–approaches and development. *Computer Standards & Interfaces*, 25(2):147 – 158.

Serrano, E., Rovatsos, M., and Botia, J. (2011). Mining qualitative context models from multiagent interactions (extended abstract). In *Proceedings of the tenth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 1215–1216. IFAAMAS.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3).

Sierra, C. (2004). Agent-mediated electronic commerce. *Autonomous Agents and Multi-Agent Systems*, 9(3):285–301.

Sierra, C. and Debenham, J. (2005). An information-based model for trust. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 497–504, New York, NY, USA. ACM.

Sierra, C. and Debenham, J. (2007a). Information-based agency. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI)*, pages 1513–1518, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sierra, C. and Debenham, J. (2007b). The LOGIC negotiation model. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 1–8, New York, NY, USA. ACM.

Sierra, C. and Debenham, J. (2008). Information-based deliberation. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 689–696. IFAAMAS.

Silaghi, M. C. and Mitra, D. (2004). Distributed constraint satisfaction and optimization with privacy enforcement. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, pages 531–535, Washington, DC, USA. IEEE Computer Society.

Smith, H. J. and Milberg, S. J. (1996). Information privacy: measuring individuals' concerns about organizational practices. *MIS Quarterly*, 20:167–196.

Solanas, A. and Martínez-ballesté, A. (2009). *Advances in Artificial Intelligence for Privacy Protection and Security*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

Solove, D. (2006). A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477–560.

Spiekermann, S. (2006). Individual price discriminaton - an impossibility? In *Proceedings of the Workshop on Privacy and Personalization held with the International Conference for Human-Computer Interaction (CHI)*, pages 47–52.

Spiekermann, S. and Cranor, L. F. (2009). Engineering privacy. *IEEE Transactions on Software Engineering*, 35(1):67–82.

Stallings, W. (2010). *Network security essentials : applications and standards*. Prentice Hall.

Stamp, M. (2006). *Information Security: Principles and Practice*. Wiley-Interscience.

Such, J. M., Alberola, J. M., Espinosa, A., and Garcia-Fornes, A. (2011). A group-oriented secure multiagent platform. *Software: Practice and Experience*, page In Press.

Taylor, H. (2003). *Most People Are "Privacy Pragmatists" Who, While Concerned about Privacy, Will Sometimes Trade It Off for Other Benefits.* Harris Interactive.

Tentori, M., Favela, J., and Rodriguez, M. D. (2006). Privacy-aware autonomous agents for pervasive healthcare. *IEEE Intelligent Systems*, 21:55–62.

Udupi, Y. and Singh, M. (2010). Information sharing among autonomous agents in referral networks. In Joseph, S., Despotovic, Z., Moro, G., and Bergamaschi, S., editors, *Agents and Peer-to-Peer Computing*, volume 5319 of *LNCS*, pages 13–26. Springer Berlin / Heidelberg.

Ugurlu, S. and Erdogan, N. (2005). An overview of secmap secure mobile agent platform. In *Proceedings of Second International Workshop on Safety and Security in Multiagent Systems*.

Urbano, J., Rocha, A. P., and Oliveira, E. (2009). Computing confidence values: Does trust dynamics matter? In *EPIA '09: Proceedings of the 14th Portuguese Conference on Artificial Intelligence*, pages 520–531. Springer-Verlag.

van Blarkom, G., Borking, J., and Olk, J., editors (2003). *Handbook of Privacy and Privacy-Enhancing Technologies: The Case of Intelligent Software Agents*. College bescherming persoonsgegevens.

van Elst, L., Dignum, V., and Abecker, A., editors (2004). *Agent Mediated Knowledge Management, International Symposium AMKM 2003, Stanford, CA, USA, March 24-26, 2003, Revised and Invited Papers*, volume 2926 of *LNCS*. Springer.

van Otterloo, S. (2005). The value of privacy: optimal strategies for privacy minded agents. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS)*, pages 1015–1022, New York, NY, USA. ACM.

Vinoski, S. (2006). Advanced message queuing protocol. *IEEE Internet Computing*, 10(6):87–89.

Wallace, R. J. and Freuder, E. C. (2005). Constraint-based reasoning and privacy/efficiency tradeoffs in multi-agent problem solving. *Artificial Intelligence*, 161:209–227.

Warnier, M. and Brazier, F. (2010). Anonymity services for multi-agent systems. *Web Intelligence and Agent Systems*, 8(2):219–232.

Warren, S. and Brandeis, L. (1890). The right to privacy. *Hardvard Law Review*, 4(5).

Weitnauer, E., Thomas, N., Rabe, F., and Kopp, S. (2008). Intelligent agents living in social virtual environments – bringing max into second life. In *Intelligent Virtual Agents (IVA)*, pages 552–553. Springer Berlin / Heidelberg.

Westin, A. (1967). *Privacy and Freedom*. New York Atheneum.

Westin, A. (1984). *Philosophical dimensions of privacy: an anthology*, chapter The origins of modern claims to privacy, pages 56–74. Cambridge University Press.

Westin, A. (2003). Social and political dimensions of privacy. *Journal of Social Issues*, 59(2):431–453.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. Wiley.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152.

Xu, H. and Shatz, S. M. (2003). Adk: An agent development kit based on a formal design model for multi-agent systems. *Journal of Automated Software Engineering*, 10:337–365.

Yao, M. Z., Rice, R. E., and Wallis, K. (2007). Predicting user concerns about online privacy. *Journal of the American Society for Information Science and Technology*, 58(5):710–722.

Yassine, A., Shirehjini, A. A. N., Shirmohammadi, S., and Tran, T. T. (2010). An intelligent agent-based model for future personal information markets. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, 2:457–460.

Yassine, A. and Shirmohammadi, S. (2009). Measuring users' privacy payoff using intelligent agents. In *Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA)*, pages 169 –174.

Yokoo, M., Suzuki, K., and Hirayama, K. (2005). Secure distributed constraint satisfaction: reaching agreement without revealing private information. *Artificial Intelligence*, 161:229–245.

Yu, H., Kaminsky, M., Gibbons, P. B., and Flaxman, A. (2006). Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pages 267–278, New York, NY, USA. ACM.

Zheleva, E. and Getoor, L. (2009). To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web (WWW)*, pages 531–540, New York, NY, USA. ACM.

Zhu, Z., Wang, G., and Du, W. (2009). Deriving private information from association rule mining results. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 18–29, Washington, DC, USA. IEEE Computer Society.