PrivaCIAS - Privacité selon l'intégrité contextuelle dans les systèmes agents décentralisés

Yann Krupa

NNT : 2012 EMSE 0657

PhD Thesis

presented by

Yann Krupa

to obtain the title of
Doctor of the École Nationale Supérieure des Mines de Saint-Étienne

Major in Computer Science

European Doctorate Label

PrivaCIAS: Privacy as Contextual Integrity in Decentralized Multi-Agent Systems

Defended on September 10th 2012, at Saint-Étienne

Jury:

President :	Pierre Maret	-	Professor, Université Jean Monnet, Saint-Étienne
Reviewers :	Javier Carbo	-	Associate Professor, Universidad Carlos III, Madrid
	Yves Demazeau	-	Senior Researcher, CNRS
	Zahia Guessoum	-	Associate Professor (HDR), Université de Reims
	Adina Magda Florea	-	Professor, Universitatea Politehnica, Bucharest
Examinators :	Grégory Bonnet	-	Associate Professor, Université de Caen
Advisors :	Laurent Vercouter	-	Professor, INSA de Rouen
	Olivier Boissier	-	Professor, ENS des Mines de Saint-Étienne

Spécialités doctorales :	Responsables :
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche
MECANIQUE ET INGENIERIE	S. Drapier, professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche
MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
INFORMATIQUE	O. Boissier, Professeur
IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
GENIE INDUSTRIEL	A. Dolgui, Professeur
MICROELECTRONIQUE	Ph. Collot, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	MA	Mécanique & Ingénierie	CIS
BATTON-HUBERT	Mireille	MA	Sciences & Génie de l'Environnement	Fayol
BENABEN	Patrick	PR 1	Sciences & Génie des Matériaux	CMP
BERNACHE-ASSOLLANT	Didier	PR 0	Génie des Procédés	CIS
BIGOT	Jean-Pierre	MR	Génie des Procédés	SPIN
BILAL	Essaïd	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR 1	Informatique	Fayol
BORBELY	Andras	MR	Sciences et Génie des Matériaux	SMS
BOUCHER	Xavier	MA	Génie Industriel	Fayol
BRODHAG	Christian	DR	Sciences & Génie de l'Environnement	Fayol
BURLAT	Patrick	PR 2	Génie industriel	Fayol
COLLOT	Philippe	PR 1	Microélectronique	CMP
COURNIL	Michel	PR 0	Génie des Procédés	SPIN
DARRIEULAT	Michel	IGM	Sciences & Génie des Matériaux	SMS
DAUZERE-PERES	Stéphane	PR 1	Génie industriel	CMP
DEBAYLE	Johan	CR	Image, Vision, Signal	CIS
DELAFOSSE	David	PR1	Sciences & Génie des Matériaux	SMS
DESRAYAUD	Christophe	MA	Mécanique & Ingénierie	SMS
DOLGUI	Alexandre	PR 1	Génie Industriel	Fayol
DRAPIER	Sylvain	PR 2	Sciences & Génie des Matériaux	SMS
FEILLET	Dominique	PR 2	Génie Industriel	CMP
FOREST	Bernard	PR 1	Sciences & Génie des Matériaux	CIS
FORMISYN	Pascal	PR 1	Sciences & Génie de l'Environnement	Fayol
FRACZKIEWICZ	Anna	DR	Sciences & Génie des Matériaux	SMS
GARCIA	Daniel	MR	Sciences de la terre	SPIN
GIRARDOT	Jean-Jacques	MR	Informatique	Fayol
GOEURIOT	Dominique	MR	Sciences & Génie des Matériaux	SMS
GRAILLOT	Didier	DR	Sciences & Génie de l'Environnement	Fayol
GROSSEAU	Philippe	MR	Génie des Procédés	SPIN
GRUY	Frédéric	MR	Génie des Procédés	SPIN
GUY	Bernard	MR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HAN	Woo-Suck	CR		SMS
HERRI	Jean-Michel	PR 2	Génie des Procédés	SPIN
INAL	Karim	PR 2	Microélectronique	CMP
KLÖCKER	Helmut	DR	Sciences & Génie des Matériaux	SMS
LAFOREST	Valérie	CR	Sciences & Génie de l'Environnement	Fayol
LERICHE	Rodolphe	CR CNRS	Mécanique et Ingénierie	SMS
LI	Jean-Michel	EC (CCI MP)	Microélectronique	CMP
MALLIARAS	George Grégory	PR 1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et Ingénierie	SMS
MONTHEILLET	Frank	DR 1 CNRS	Sciences & Génie des Matériaux	SMS
PERIER-CAMBY	Laurent	PR 2	Génie des Procédés	SPIN
PIJOLAT	Christophe	PR 1	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR 1	Génie des Procédés	SPIN
PINOLI	Jean-Charles	PR 0	Image, Vision, Signal	CIS
ROUSTANT	Olivier	MA		Fayol
STOLARZ	Jacques	CR	Sciences & Génie des Matériaux	SMS
SZAFNICKI	Konrad	MR	Sciences & Génie de l'Environnement	Fayol
TRIA	Assia		Microélectronique	CMP
VALDIVIESO	François	MA	Sciences & Génie des Matériaux	SMS
VIRICELLE	Jean-Paul	MR	Génie des procédés	SPIN
WOLSKI	Krzysztof	DR	Sciences & Génie des Matériaux	SMS
XIE	Xiaolan	PR 1	Génie industriel	CIS

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
KERMOUCHE	Guillaume	MCF	Mécanique et Ingénierie	ENISE
RECH	Joël	MCF	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	MCF	Mécanique et Ingénierie	ENISE
GUSSAROV Andrey	Andrey	Enseignant contractuel	Génie des procédés	ENISE

Glossaire :

PR 0	Professeur classe exceptionnelle	Ing.	Ingénieur
PR 1	Professeur 1ère classe	MCF	Maître de conférences
PR 2	Professeur 2ème classe	MR(DR2)	Maître de recherche
PU	Professeur des Universités	CR	Chargé de recherche
MA(MDC)	Maître assistant	EC	Enseignant-chercheur
DR	Directeur de recherche	IGM	Ingénieur général des mines

Centres :

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
FAYOL	Institut Henri Fayol
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

Dernière mise à jour le : 02 avril 2012

I dedicate this work to my beloved wife Nadia,

Acknowledgments

The true delight is in the finding out rather than in the knowing.
Isaac Asimov

I would like to thank various people who helped me during my thesis.

First of all, many thanks go my main supervisor, Pr. Laurent Vercouter, for his patient guidance and useful critiques. Pr. Olivier Boissier, my second supervisor also did a great work, especially during the latest (and hardest) months of the thesis. I want to thank the reviewers and the jury members, it has been an honor that you accepted to review my works.

I also want to thank all the ISCOD team and more specifically the former SMA team members: Philippe, Reda, Camille, Bissan, Amro and also Rosine and Gauthier who shared their office with me. Thank you Reda, for being there when I needed your help.

There are numerous people who made my stay at the École des Mines a pleasant one: Antoine, Gregory, Nilou, Tcheb, Marie-Line, Malik, Mariana, Sonia. The mathematicians were also good company, especially during the coffee breaks: Rodolphe, Xavier, Olivier, Anca, Bertrand, Nicolas.

During my thesis I had the opportunity to go two times in the IIIA-CSIC center near Barcelona for a total of 7 months. Firstly, I want to thank the Rhone-Alpes Region for the "ExploRA'Doc" mobility grant. Secondly, I want to thank all the fabulous people who received me during these stays: Dr. Jordi Sabater-Mir, my local contact, but also the administrative members and scientists that were very helpful. Special thanks go to Dani A. and Dani P. for their help whenever I needed some.

I also have a thought for Dr. Marc Esteva, who I had the chance to meet in conferences and in IIIA, and who passed away when I was writing the thesis.

Having a brilliant master student to supervise was a good experience, thanks to Andrei Ciortea for working on my subject.

Big thanks to my family, especially my parents (Mireille and Daniel), my grand parents (Yvonne and Paul), and my friends for supporting me during the thesis. Thank you Sébastien, my long time friend, for our good discussions when I needed to talk about my thesis.

Thanks to Dr. Olivier Commowick for providing this LaTeX thesis style on his website.

I probably forgot people that helped me, thanks to all the forgotten ones.

And finally, thank you Nadia, my love, for everything that you bring into my life.

Contents

Introduction **1**

I State of the Art **5**

1 Privacy in Social Sciences **7**
1.1 Privacy in Human Society 9
1.1.1 Understanding Privacy 9
1.1.2 Privacy in the Information Era 13
1.2 Studying Privacy 15
1.2.1 Educating Users 16
1.2.2 Transmitting Out of Context 18
1.3 Contextual Integrity 18
1.3.1 Transmissions in Contexts 18
1.3.2 Appropriateness 20
1.3.3 Context Relative Norms 21
1.4 Chapter Conclusion 22

2 Technologies for Information Control **25**
2.1 Criteria for Privacy Analysis 27
2.1.1 Orientation 27
2.1.2 Architecture 27
2.1.3 Privacy Enforcement 28
2.1.4 Strength 28
2.1.5 Retransmission and Reuse 29
2.1.6 Genericity 30
2.2 A Survey of Privacy Enhancing Technologies 30
2.2.1 Asymmetric Encryption 31
2.2.2 Access Control 32
2.2.3 Anonymity 34
2.2.4 Privacy Policies and P3P 37
2.2.5 Hippocratic Databases 39

2.2.6 Hippocratic Multi-agent Systems . . . 41
2.2.7 Digital Rights Management Systems . . . 43
2.2.8 Sticky Policies . . . 45
2.2.9 Privacy Preserving Trusted Third Parties . . . 47
2.3 Privacy in Social Networks . . . 48
2.3.1 Facebook . . . 48
2.3.2 Diaspora . . . 50
2.3.3 Google+ . . . 51
2.4 Chapter Conclusion . . . 52

3 Controlling Agents in Multi-Agent Systems 55
3.1 Norm Description . . . 58
3.1.1 The Nature of Norms . . . 58
3.1.2 Norm Emergence, Creation and Spreading . . . 60
3.2 Institutional Control . . . 61
3.2.1 Regimentation . . . 61
3.2.2 Communication Filters . . . 61
3.2.3 Electronic Institutions and Organizations . . . 62
3.3 Social Control . . . 63
3.3.1 Trust . . . 64
3.3.2 Reputation . . . 66
3.3.3 Social Exclusion . . . 67
3.4 Chapter Conclusion . . . 68

II The PrivaCIAS Model 69

4 Introducing the PrivaCIAS Model 71
4.1 Requirements . . . 73
4.2 Tackling the Problem . . . 74
4.2.1 Sending Messages . . . 75
4.2.2 Receiving Messages . . . 76
4.3 Overview of the PrivaCIAS Model . . . 76

5 Appropriateness Laws **79**
5.1 Embracing Contextual Integrity . . . 81
5.2 The PrivaCIAS Components . . . 83
5.2.1 Roles and Contexts . . . 84
5.2.2 Message Structure . . . 86
5.2.3 PrivaCIAS Predicates and Preferences Language . . . 88
5.3 Formalizing A-laws . . . 92
5.4 Example Scenario . . . 93
5.5 Chapter Conclusion . . . 96

6 Privacy Enforcing Norms **99**
6.1 Privacy Enforcing Norms . . . 101
6.1.1 Prevent Privacy Violations . . . 101
6.1.2 Stop Privacy Violations . . . 102
6.1.3 Punish Privacy Violations . . . 103
6.1.4 The Six PENs . . . 104
6.2 Privacy Enforcing Agents . . . 105
6.2.1 Receiving . . . 106
6.2.2 Sending . . . 109
6.3 Trust Related Functionalities . . . 111
6.4 Example Scenario . . . 112
6.5 Chapter Conclusion . . . 115

III Model Validation and Deployment **117**

7 PrivaCIAS Stress Test with RePast **119**
7.1 Experimental Settings . . . 121
7.1.1 Requirements . . . 121
7.1.2 RePast . . . 121
7.1.3 Simplified PrivaCIAS . . . 122
7.1.4 Metrics . . . 124
7.2 Experimenting on PrivaCIAS . . . 127
7.2.1 Default Settings . . . 127
7.2.2 Worst Case Scenario . . . 128
7.2.3 Alternative Worst Case Scenario . . . 128

7.2.4 Best Case Scenario . . . 130
7.2.5 Limits . . . 130
7.2.6 Standard Scenario . . . 133
7.3 Probing the PENs . . . 134
7.3.1 No Control on Sending . . . 138
7.3.2 No Control on Receiving . . . 139
7.3.3 No Punishments . . . 140
7.3.4 Summary . . . 140
7.4 Attacks and Solutions . . . 141
7.4.1 Meta-Information Tampering . . . 141
7.4.2 Reputation Tampering . . . 141
7.4.3 Context Tampering . . . 142
7.5 Chapter Conclusion . . . 143

8 The PrivaCIAS Photo-Sharing Application 145
8.1 Specification of the PrivaCIAS Demonstration . . . 147
8.2 The PrivaCIAS Photo-sharing Application . . . 148
8.2.1 Incompatible Relationships . . . 148
8.2.2 Application Components and Structure . . . 149
8.2.3 Illustrated Scenario . . . 151
8.3 Chapter Conclusion . . . 153

Conclusion 159

A PrivaCIAS: French Summary 163
A.1 Introduction . . . 163
A.2 État de l'Art . . . 164
A.2.1 Privacité dans les sciences sociales . . . 164
A.2.2 Technologies pour le contrôle de l'Information . . . 166
A.2.3 Contrôle des agents dans un système multiagent . . . 168
A.3 Le Modèle PrivaCIAS . . . 170
A.3.1 Introduction au Modèle PrivaCIAS . . . 170
A.3.2 Lois de convenance et langage de communication . . . 171
A.3.3 Normes de support de la Privacité . . . 175
A.4 Validation et expériences . . . 180
A.4.1 Test de résistance sous RePast . . . 180

A.4.2 Application de démonstration sous Android 182
A.5 Conclusion 183

Publications **185**

Bibliography **187**

Introduction

PrivaCIAS: Privacy as Contextual Integrity in decentralized multi-Agent Systems. This thesis shows that it is possible to protect privacy in decentralized and open environments. Instead of constraining users and sometimes developers of multi-agent systems, we propose a model that intends to bring a social order in the system, for privacy preservation.

Needs

Privacy is a concern that has been growing up, especially during the last decade. With the rise of social networks, users became actors of their own and other's privacy protection.

More or less successful attempts to protect users from institutions, from "big brother", have been made. This "big brother" issue was, until recently, the most prominent in information technologies. With the development of the participative web, the "Web 2.0", users began to interact with each other and the "horizontal" (user to user) privacy issue gained in importance. Anyway, too few interest have been shown regarding protecting users "horizontally". Most approaches tried to reuse either computer security or trusted computing techniques in order to protect users privacy horizontally. For instance, most social networks use access control (a computer security technique) to protect users form others. These techniques most of the time rely on a central authority that has access to user data, and brings a "big brother" issue when trying to solve the "horizontal" privacy issue.

Some of the latest attempts at creating social networks, we can name Diaspora, started to decentralize their services. Decentralization, in general, allows a better robustness of the system and tolerance to attack. Instead of having a single central node, the system is composed by a number of nodes that are independent. Moreover, decentralization could protect users from being spied on by institutions as each user could create and host his own node.

Providing a privacy enhancing technology for a decentralized system is a difficult problem. This thesis addresses the two main issues to overcome for protecting privacy in a decentralized social network:

- **How to detect privacy violations from the user point of view** and give tools to users to protect their data against privacy violations?
- **How to control a system without a central authority** in order to prevent and stop privacy violations in the system?

Goals

In this thesis, we want to define a privacy preserving model for open and decentralized systems, represented by Multi-Agent Systems.

As we target decentralized and open systems, approaches that can be successfully developed are agent-based techniques. As there is no system, system-based techniques are almost impossible to implement. Most of the existing privacy enhancing technologies, as we shall see, are inapplicable. An adapted solution has to be found.

Multi-agent systems, and more specifically, virtual communities can be matched to human communities as they represent autonomous entities interacting with others in an environment. Therefore, we focus on social sciences, and socio-inspired multi-agent models to provide an agent-based control that will work in decentralized environments. Users in the system will be provided with personal assistant agents that will make them privacy-aware.

Previous works in the ISCOD department have been realized regarding socio-inspired multi-agent models, especially on trust [Vercouter 2010], reputation [Nardin 2008], organizations [Hubner 2002a, Hübner 2010, Hubner 2007] and normative systems [Gâteau 2007, Grizard 2007].

The model that we define should be generic enough to be implemented in a wide range of applications. We target decentralized social networks as the main application domain but we want to be able to apply the model on business-to-business networks for instance.

As socio-inspired models are hard to prove theoretically, a set of experiments has to be defined to test the model as its limits.

Agenda

This thesis is divided in three parts: the state of the art, the presentation of our model and its experimental validation.

The first part presents the state of the art through three chapters. We study the notion of privacy as it is seen by social scientists. This study allows us to find a social theory of privacy. The following state of the art study concerns the technological measures that can be brought into place to preserve privacy in computer-based applications. The last study regards means of controlling participants in a system, and more specifically in a decentralized system.

The second part of the thesis presents our model. The social theory, studied in the first part of the thesis, is formalized to be used within a multi-agent system that we specify. We define norms to control agents in this privacy preserving system, using what has been learned form the state of the art.

In order to demonstrate and validate the soundness of our model, the last part of the thesis describes two applications that implement the model. First, we define a set of experiments developed under RePast to probe the limits of our model. Second, we describe the photo-sharing application that has been developed as a show-case for our system.

Part I

State of the Art

Chapter 1

Privacy in Social Sciences

The problem, in short, is not finding an answer to the question: "If you've got nothing to hide, then what do you have to fear?" The problem is in the very question itself.

Daniel J. Solove

In our life, powered by information technology, privacy is becoming a major concern. In the society, everything is "logged": when people withdraw money from an ATM, when they use their transport card to take the bus, when they use their mobile phone, etc. Other problems can arise when exchanging information with friends or colleagues, because in the electronic age, every document can be republished hundred times in a matter of seconds. Generally speaking, people are afraid of unauthorized use of information regarding them. The problem is well known and has been intensively studied by scientists for the last years.

This chapter presents the problem of privacy from the human society point of view. The first section reviews books, scientific and press articles on the topic. The second section describes the issue as it is seen by scientists from social sciences, philosophy and lawyers. The third section focuses on the Contextual Integrity theory developed by Helen Nissenbaum and presents works related to this theory.

Contents

1.1 Privacy in Human Society 9
1.1.1 Understanding Privacy 9
1.1.2 Privacy in the Information Era 13
1.2 Studying Privacy 15
1.2.1 Educating Users 16
1.2.2 Transmitting Out of Context 18
1.3 Contextual Integrity 18
1.3.1 Transmissions in Contexts 18
1.3.2 Appropriateness 20
1.3.3 Context Relative Norms 21
1.4 Chapter Conclusion 22

1.1 Privacy in Human Society

Privacy became a first-level social and political issue in the US during the 1990-2002 era [Westin 2003]. The tremendous evolution of privacy over this decade is the consequence of two main events. Firstly these years saw the rise of computers and the Internet. Computers are found in most of the high-tech objects that we use: in ATM, in cars, in phones, etc. Secondly, the 9/11 attacks on the World Trade Center dramatically changed the politics regarding public surveillance and privacy [Solove 2007, Westin 2003, Baase 2007]. Before the events, a rough equilibrium between privacy and surveillance from governments existed. The attacks changed the game, the equilibrium got broken by the government, and citizens started to feel their privacy being violated by the government. In other words, the growth of information technology made it easier and possible for the government, or corporations, to spy on citizens. The 9/11 attacks gave governments (for the most part the US government) a good reason to do so. This combination raised the awareness of privacy-related problems among society.

Another illustration of the evolution of privacy is given by Sara Baase in her book *A Gift of fire* [Baase 2007]. She gives an example of privacy before the information age: In East Germany, after the fall of the communist government, people discovered the files left by the Stasi, the secret police. The government was spying on about six million people, filling with paper files an estimated 125 miles of shelf space. At that time, storing, searching or copying such records was very time-consuming as computers were not used. This example highlights, as Sara Baase states, that computers are not necessary for the invasion of privacy. However, they bring new threats as computer technology "has a profound impact on what information is collected about us, who has access to it and how they use it", she writes. Case in point, computers allow to search and copy huge amounts of data in a matter of seconds.

1.1.1 Understanding Privacy

Privacy is a concept that varies, as we shall see in the next subsection, across people. Privacy has a value and can be traded as a good. But most people do not estimate correctly the value of privacy and more generally have misconceptions about privacy. Eventually we present a distinction between two different privacy

problems, horizontal privacy and vertical privacy.

1.1.1.1 A Definition of Privacy

It is, in fact, quite difficult to give a universal definition of privacy. As it has been noted, privacy varies among cultures, times, and societies [Nissenbaum 2004]. The word *privacy* is even said to be untranslatable, as some languages do not have a corresponding word (*e.g., French*) [Anderman 2003].

Sara Baase gives a definition of privacy in her book, by providing the three key aspects of privacy [Baase 2007]:

- freedom from intrusion–being left alone
- control of information about oneself
- freedom from surveillance (from being followed watched, and eavesdropped upon)

This definition is quite general and it is possible to simplify it. *Freedom from surveillance* can be merged with *control of information about oneself* under the assumption that when someone eavesdrops somebody, he retrieves information about that person. Once again, the concept of *being left alone* can be merged with *control of information about oneself*. When saying *would you give me some privacy please*, the person is asking to be left alone to be able to act without being observed. These doings are also information about the person, more specifically, about the person behavior. We are left only with the second bullet which is *control information about oneself*. When someone has lost control of his information, because it is being misused, transmitted to the wrong person his privacy is violated.

1.1.1.2 The Value of Privacy

Most people accept that complete control over data, thus complete privacy, is impossible to achieve, except if you want to live in total seclusion with no social contact with other human beings. We usually exchange information about ourselves with our friends, family or coworkers. When we do, we lose control over the information that has been given away. More generally, information is traded for goods or money. We will trade personal details on our lives, our address and

our phone number to obtain a good in exchange: social relationships. In fact, giving away personal information is equivalent to giving away privacy. Most of the people in the western world have "customer loyalty cards" that gives them financial advantage while the store owner is able to profile its client. For that financial advantage, people give away their privacy. People are also incline to let the government watch and eavesdrop them for national security reasons, they trade privacy for security.

For example, to get back on the 9/11, after the events the US government started voting intrusive surveillance laws, like the Patriot Act. This law allows the FBI to search without a warrant on the simple suspicion from an agent that the investigation may be related to terrorism. Daniel Solove relates [Solove 2007] that the *New York Times* revealed that the Bush administration have secretly authorized the National Security Agency to engage in warrantless wiretapping of citizens. While some people responded with outrage at those announcements, others did not perceived a problem because, they explained: "I've got nothing to hide" [Solove 2007]. As Solove notes, the *nothing to hide* argument highlights two things:

- Some people do not mind trading privacy for security.
- They do not understand privacy and its implications.

The first point has just been discussed. The second point, about the meaning of privacy and its implications is studied in the following subsection.

1.1.1.3 (Mis)Understanding Privacy

People do not understand privacy, this is why Solove says: "The problem, in short, is not finding an answer to the question: If you've got nothing to hide, then what do you have to fear? The problem is in the very question itself." In fact, many people think privacy is only for those having something to hide, *i.e.* people engaging in illegal activities. This is a serious misconception of privacy. A good counter example is the health situation of a person. A person may have an illness that will make an employer discriminate against him. Having an illness is not illegal, it is neither shameful, but giving away that information may harm the subject of that information.

Across cultures, the misunderstanding of privacy can also be explained by other factors because privacy is culture-dependent [Nissenbaum 2004] as it was noted earlier. For example, on the one hand, it is very common to share information about one's salary, in the US. On the other hand, in France, it is considered that sharing this information is rude.

Moreover, some people confuse different types of privacy. They tolerate that an institution, *e.g.* the government, is able to log everything they do, but they will probably be very upset if their neighbors or coworkers were able to gain access to this data or lure in their private spaces. There is a strong distinction between privacy issues regarding institutions and privacy issues regarding other people, and when they use the "nothing to hide argument" they make no distinction between both.

1.1.1.4 Vertical and Horizontal Privacy

There is a difference between having our privacy violated by an institution, a big brother watching on us, and having our privacy violated by a peer, a neighbor for example. We can distinguish between [Grimmelmann 2008]:

- Vertical Privacy
- Horizontal Privacy

When talking about privacy, what first comes in mind is the vertical privacy: "Big Brother is Watching You" as written by Orwell in his novel *1984* [Orwell 2006]. Vertical privacy is about the individual having his privacy violated by an institution (a firm, a website, a government, etc.). When a person registers to a service (or even a public service) he has to provide information to the service provider. As a provider, the service has power over the data it is holding, and sometimes it has power over the users. Also, it is very common that the service provider wants to use the available data for profit, as we have shown, information is valuable and can be traded. This problem occurs because the data regarding a given user is stored in the provider files, therefore staying at hand for the provider to use.

Horizontal privacy, on the other hand, is about having their privacy violated by someone on the same level: a friend, a colleague, etc. The violator does not have power over the data, therefore the problem is different as the data owner has a better control of its own information.

For example, if our neighbors are taking pictures of us without our consent, our horizontal privacy will be violated. If the physician sells our health records to a corporation for money, our vertical privacy will be violated. The very difference between vertical and horizontal privacy is that in vertical privacy the "opponent" has power and control over our data or ourselves whereas in horizontal privacy, the opponent is on the same level.

But in some situations, vertical and horizontal privacy issues can be present at the same time. If a person is afraid that the institution uses their data and is at the same time afraid that another person may gain access to their data. It happens in social networks, users are afraid that the provider may send their private information to another corporation and at the same time users are afraid that people may gain access to information they thought they had protected.

Both problems are present in information technologies. Nevertheless, vertical privacy is the one which has been the most thoroughly studied because it was the most prominent one in the early ages of information technologies. Horizontal privacy problems gained in importance more recently with the tremendous growth of social networks. As we shall see, horizontal privacy is also an important issue and owes to be studied thoroughly.

1.1.2 Privacy in the Information Era

During these last ten years, computers made their ways into our lives. In western countries, it is almost impossible to spend a day without using a computer. In fact, computers are everywhere: in phones, in ATM, in cars, etc. Most of these computers are logging information, sometimes for legal requirements (ATM), sometimes for profiling the user. At the same time, the Internet underwent a phenomenal growth.

1.1.2.1 The Rise of Privacy Awareness

Through this growth, politics started to become aware of privacy problems and started to edict laws. We focus here on the French current laws and history of privacy regulations. French laws are uniform with regard to European regulations for privacy. Nevertheless, they are very different from US laws. In France, the right to privacy was enacted in 1803, but in a generic way as the information age was yet unforeseen. The 1978 "Information Technologies and Liberty" act defined

the acceptable privacy policies that can be applied in France, regarding access, transmission, right to update or delete personal information. This law applies specifically to "automated processes". Therefore it takes into account problems regarding information technologies. This same act also creates an administrative authority, the CNIL (Commission Nationale de l'Informatique et des Libertés[1]), in charge of giving away authorizations to corporations to collect personal data.

Created in 1978 to protect personal user data, the CNIL was reformed in 2004, giving the organization the right to do "control missions". These missions grants them the power of querying corporation databases to verify if they are compliant with regard to their privacy policies and the French law. All corporations logging personal information into databases must declare themselves to the CNIL. If they do not comply, they can be sued. Other authorities of the same kind have been created in European countries as EU regulations from 1995 imposes the creation of such authorities for every EU member.

Another addition from the 2004 revision is the creation of "Information Technologies and Liberty Correspondents" (*Correspondant Informatique et Libertés*, CIL[2]). Corporations have a consequent amount of requirements to fulfill to be compliant with the law. Hiring a CIL allows the corporation to lower those requirements. The CIL works for the company but is in contact with the CNIL, his mission is to check that the company respects the law regarding personal data collection. The CIL has to warn the company if they do not comply. If the warnings given by the CIL are not sufficient, he has to refer to the CNIL. Therefore, the CIL is protected by the law from his employer and it is impossible for the company to fire him unless the CNIL has granted the company the right to do so.

This example from the French institutions shows that government, people and companies have realized that privacy is a concern of first importance. These authorities, like the CNIL, regulate privacy vertically for the most part. They prevent the misuse of user information by the government or companies. Nevertheless, with the growth of social networks, many users started to put online personal information, such as their address, their real name and job. Users are not careful enough about what other users are able to access, exposing themselves

[1] http://www.cnil.fr/

[2] http://www.cnil.fr/fileadmin/documents/Guides_pratiques/CNIL_Guide_correspondants.pdf

to horizontal privacy violations.

1.1.2.2 The Harms of Privacy

Privacy is not a luxury: privacy violations can harm everybody. Multiple cases where brought up by the press. For example, there is the very well known case of that intern from a bank [Jernigan 2009]. The intern declared he had a "family emergency" and that he was not going to work. In fact, he went to a Halloween party and posted pictures of himself dressed as a fairy on his Facebook account. His boss came across one of his pictures posted online. He was fired. There have also been cases of people killing each other, over profile updates or sexual predator lurking on their preys on the social networks sites. As an example, Adrei Ciortea relates in his Master's thesis the story of Keri McMullen and Kurt Pendleton [Ciortea 2011]. Keri and Kurt left a status update that said they will be going to a concert on Saturday night. When they got back home, they discovered that they have been burglarized. Luckily, they have installed a few months before a surveillance system. By looking at the images, Keri recognized one of the burglars: he had *friended* her a few months before[3]. In the press, hundreds of similar stories can be found, like this man who killed his wife because she changed her Facebook *status* to *single*[4].

It shows how privacy is important to our jobs, social relationships or even our lives. With the growth of social networks and, more generally, of information technologies, the scale of information transaction changed drastically. Henceforth, privacy became a public concern, with the media and politics taking up on the subject. A better understanding of privacy comes from all these events that have been thoroughly studied by sociologists, philosophers, and lawyers as shows the next section.

1.2 Studying Privacy

The previous section gives a snapshot of privacy problems in our society. This section presents the analysis provided by social scientists, philosophers and lawyers.

[3] http://www.cbsnews.com/stories/2010/03/25/earlyshow/main6331796.shtml
[4] http://news.bbc.co.uk/2/hi/uk_news/england/staffordshire/7845946.stm

1.2.1 Educating Users

One of the first conclusions that comes out from studies is that people do not understand privacy and the consequences of privacy violations. An important step is to educate users, to teach them the basics of privacy.

1.2.1.1 Benefits *vs.* Costs of Sharing Information

We no longer control what others know about us, but we don't yet understand the consequences is the title of a book from P. Bradwell and N. Gallagher [Bradwell 2007]. They raise interesting questions about privacy. Case in point, today it is very important to be seen, to exist on the internet and at the same time, the problem of being watched arises. People or companies may use the provided information to profile or spy on you. Bradwell and Gallagher highlight the difference between vertical privacy (big brother) and horizontal privacy (people watching people). In their opinion the first step, for individuals, is to take measures for protecting their personal information and to recognize the connections between the benefits of sharing information and the less tangible costs or dangers that can result. "Personal information and the way it is used matters politically, and democratically, because it is intimately connected with how we are seen, represented and treated by people, organizations and institutions that hold influences and power over us."

1.2.1.2 Privacy as Common Sense

Fogel and Nehmad [Fogel 2009] write that social networking sites should inform potential users that risk taking and privacy concerns are potentially relevant and important concerns before sign up. On the opposite, some lawyers like James Grimmelmann [Grimmelmann 2008] remarks that what's more important is educating users. According to him, website privacy policies would not work because people do not read it and do not understand the potential risks– 77% of Facebook users, responding to Aquisti and Gross's survey declared they have not read the policies. Grimmelmann thinks that you should call to the common sense of users in social networks, you do not need to warn a user, you have to educate people even before they sign up. In other words, people should know what information is safe to share or not, this should be part of the common sense. Other controls

like restrictions or technical controls will not work he says, because when you bring in place restrictions, people will trick the system. For example, some networks where forbidden for children under 18. To put their real age, 16 years old teenagers used to register as 61 years old [Grimmelmann 2008]. People have to take responsibility themselves for the information they share, the system will not be able to protect users by force and against their will.

What should or should not be shared among social networks must become part of the common sense, and educating users is a step toward this achievement.

1.2.1.3 Being Consistent

Acquisti and Gross [Acquisti 2006] studied Facebook user's concerns regarding privacy through a survey. Among the interesting results they got, respondents declared their highest levels of concern for "A stranger knew where you live and the location and schedule of the classes you take", and "Five years from now, complete strangers would be able to find out easily your sexual orientation, the name of your current partner and your current political views." At the same time, 24% of respondents provided their personal address, 42% their schedule of classes, 53% their political views, 59% their sexual orientation and 28% their partner name. Users have problems matching their "real life" fears with regard to the information they provide to complete strangers. They have to become consistent with their concerns.

In order to make user realize and bridge the gap between their real life and virtual life privacy concerns, some actions are taken in high-school and universities in France. Students are given access to a serious game about privacy[5]. In this game, the student goes back in the past to prevent himself and his friends from broadcasting a picture of him that will ruin his life many years later. The game focuses on educating users about being careful on who they accept as friends, what information do they share, and so forth. Moreover, the game highlights the impact that virtual world actions (such as accepting a friend) can have on the real world. Educating users shall help them protect their own privacy as they become consistent with their fears.

[5] http://www.2025exmachina.net/

1.2.2 Transmitting Out of Context

Apart from educating users, the second predominant issue is being careful about the context of transmissions. Information is exchanged in a given context, thus for a given goal. This is the issue that was encountered by the previously cited bank intern. He exchanged his picture in a *friendship* context but wasn't expecting the picture to go out of the closed loop and jump into the *work* context when his boss came across the picture. Another example is given by Philippe Testart-Vaillant in the CNRS Journal [Testard-Vaillant 2008], he discusses the Robert Rivera case. In 1995, Rivera slipped on a spilled yogurt while buying groceries. He sued the supermarket for financial compensation for his broken knee. Rivera was blackmailed by the supermarket lawyers: through his loyalty card, they knew that he had the habit of buying liquor. The lawyers told him that this information might become part of the trial, leading the court to believe he was drunk at the moment he fell down. Saadi Lahlou, in the same article comments that: "As long as everything is OK, private data helps the user so that everything becomes quicker and easier. But data provided in one context for a given goal can be used for another goal in a different context, without the user being aware of the misuse." Solove, in his book *Understanding Privacy* [Solove 2008] highlights that many sociologists recognize the importance of context in understanding privacy. This is the central concept of the Contextual Integrity theory, described by the lawyer Helen Nissenbaum.

1.3 Contextual Integrity

In this section we present the theory of Contextual Integrity [Nissenbaum 2004, Nissenbaum 2010] by Helen Nissenbaum, which inspired our works.

1.3.1 Transmissions in Contexts

In section 1.1.1.1 the three key aspects of privacy have been presented. On the same line, Nissenbaum presents in the Washington law review [Nissenbaum 2004] the three principles of privacy behind privacy policies and laws in the United States in what she calls the three principles framework. These principles are equivalent to Sara Baase's ones. They are the following:

- limiting surveillance of citizens and use of information about them by agents of government,
- restricting access to sensitive, personal, or private information,
- curtailing intrusions into spaces deemed private or personal.

Anything that intrudes on one of these principles is a potential privacy violation. But analyzing those three principles reveals that in fact, they are not universal: case in point, sensitive and private information may be accessed by a medical doctor in order to cure the subject, surveillance of a given citizen may be granted to federal agents for his or her own protection or warrants can be issued to get an intrusive insight into a suspect house. This is the basic idea that lies behind contextual integrity: "whether an action is a violation of privacy or not depends on the context of the action".

Another central tenet of contextual integrity is that "there are no arenas of life not governed by norms of information flows, no informations or spheres of life for which anything goes." [Nissenbaum 2004] Everything that we do or say happens in a context with its conventions and cultural expectations. The idea of a simple private/public dichotomy is therefore rejected. These contexts could be expressed as "spheres of life" such as education, politics or more more finely drawn as the routines of visiting a dentist, attending a family wedding. Nissenbaum argues that "for some purposes, broad sweeps like the private/public dichotomy have proven themselves useful." But privacy norms seem to lead toward finer grain contexts or spheres.

Depending on the context, the norms may be explicit and specific. It is the case for the medical context in which specific rules are enacted to regulate the way doctors are allowed to share information regarding the patients. In other contexts, norms may be implicit, variable, and incomplete. As an example, some people are very secret about their love affairs, on the opposite some other are akin to broadly share their affairs. Therefore instead of defining an information as private or public once for all, Nissenbaum argues that constraints related to the background situation, or context, have to be defined: "There is, indeed, great complexity and variability in the privacy constraints people expect to hold over the flow of information, but these expectations are systematically related to characteristics of the background social situation." [Nissenbaum 2010] In other

words, for some contexts, norms can be predefined, and more generally users should be able to express their expectations for the flow of information regarding themselves. The central thesis of her book is that a right to privacy is neither a right to secrecy nor a right to control but a right to appropriate flow of personal information [Nissenbaum 2010].

1.3.2 Appropriateness

Norms of appropriateness dictate what information about persons is appropriate, or fitting, to reveal in a particular context. An Example of how appropriateness is dependent from the context, and how a violation can occur if an information is taken from one context to an other, is given by Schoeman in his article *Gossip and Privacy* [Schoeman 1994]: "a person can be active in the gay pride movement in San Francisco, but be private about her sexual preferences vis-à-vis her family and coworkers in Sacramento. A professor may be highly visible to other gays at the gay bar but discreet about sexual orientation at the university. Surely the streets and newspapers of San Francisco are public places as are the gay bars in the quiet university town. Does appearing in some public settings as a gay activist mean that the person concerned has waived her rights to civil inattention, to feeling violated if confronted in another setting?"

In order to have a complete description of the foundations of Nissenbaum's Contextual Integrity theory, the reader should refer to the original article [Nissenbaum 2004]. Here we only focus on the concept of "violation". Nissenbaum argues that "whether a particular action is determined a violation of privacy is a function of:

1. the nature of the situation/context
2. nature of the information with regard to the context
3. roles of agents receiving the information
4. relation of agents to information subject
5. terms of dissemination defined by the subject"

The nature of the situation is the context in which the information is transmitted or recovered. It can be defined within a more or less accurate way depending on the objectives.

The nature of the information must clearly be appropriate for the given context. It seems inappropriate, for example, to discuss family secrets at work.

The roles of the agents receiving the information are also important, even when sharing medical (nature of information) in a hospital (context), you have to be sure this information is not given away to the janitor but only to agents having roles in the given context.

Sometimes, even if all the above conditions are satisfied, an incompatible relation can exist between the subject of the information and the person that can receive the information. If my father-in-law is a doctor working in the hospital, it may be inappropriate to provide him with my health records.

Finally, terms of dissemination can be defined by the subject to specify norms that may not appear from the common sense, this is important to provide the subject with control over his personal information.

H. Nissenbaum notes that one consequence of her definition is that instead of being predefined and fixed, the privacy prescriptions are now "shaped to a significant degree by local factors, are likely to vary across culture, historical period, locale, and so on." Also she notes that "the private/public dichotomy can be understood as a cruder version of contextual integrity, postulating only two contexts with distinct sets of informational norms for each–privacy constraints in the private, anything goes in the public." [Nissenbaum 2010] This last sentence can be used to demonstrate that Contextual Integrity theory is at least as much powerful as the private/public dichotomy.

1.3.3 Context Relative Norms

The terms of dissemination described above are completed with context relative norms. In her book [Nissenbaum 2010] Nissenbaum defines *context relative norms*, "characterized by four key parameters: context, actors, attributes, and transmission principles. Generally, they prescribe, for a given context, the types of information, the parties who are the subjects of the information as well as those who are sending and receiving it, and the principles under which this information is transmitted. [Nissenbaum 2010]" These norms represent the real life norms that are explicit and specific to a context, as explained in section 1.3.1.

While the terms of dissemination are specific to a message, context relative norms constrain the transmission within a context.

Barth et al. [Barth 2006] present a formal framework for expressing privacy expectations and privacy practices, inspired by contextual integrity. They define two kinds of norms, expressed in temporal logic, that they call positive and negative. These two kinds of norms allow them to express "allow" and "deny" rules in traditional access control. They define privacy norms for a specific application using LTL and interpret these formulas over traces. Traces include communications of the form "Alice gives Bob a certain type of information to Charlie". Their goal is to apply the model by using a model checker in a system for handling electronic health records to ensure that the system complies with the law regarding privacy. In other words, their model will be used to check if a system complies with the privacy policies it is supposed to follow, but it is not conceived for use in decentralized environments.

1.4 Chapter Conclusion

From this study of how privacy is seen by social scientists and lawyers, we can extract the different concepts that led this thesis. First of all, it has been shown that in the information era, privacy has become a major concern. Secondly, we pointed out that social scientists analyzed two major items:

- users are uneducated, unaware, and incautious about their privacy,
- contexts are important to avoid privacy violations.

We have seen that some measures can be taken to educate users. For example, teaching them the basics of privacy and make them realize the impact that online privacy violations can have on their real life. A complementary solution is to provide the user with a personal assistant. The assistant could prevent violations from happening by warning the user in hazardous situations.

The legitimate question that follows is: *On what grounds an assistant could be able to warn users?*

Social scientists noted that contexts are important to avoid privacy violations. Therefore, the assistant agent could rely on contexts to avoid violations. The theory of Contextual Integrity can serve as a stub to define, in computer science terms, the concept of privacy violation.

Privacy has been known as an issue in information technology for years. Therefore, apart from analyzing the social aspects of the issue, many scientist tackled the problem under a technical angle. The next section describes the different approaches that have been tried to address the privacy preservation problem, from a technical point of view.

Chapter 2

Technologies for Information Control

If we've learned anything at all from the DRM wars, it's that technical controls are rarely effective against a person genuinely determined to redistribute information they've been given access to.

James Grimmelmann

In the previous chapter, different perceptions of privacy have been presented.Two main privacy problems come out of the study: horizontal and vertical privacy. While vertical privacy tackles the problem of the system trying to access user data, horizontal privacy focuses on controlling the way users access other users' data.

These problems, described from a sociological point of view, have been tackled in computer science. As soon as computers were able to process information, the question of privacy in computers became a concern, as shown previously. But while computers allow to process huge amount of data, they also allow to control this data in order to protect privacy. Initially, information control was only focused on critical information, like national security related information. Then the question broadened and the concern grew over personal data and corporate information.

This chapter presents work and research that have been done in computer science for information control. In other words, works regarding privacy, but also confidentiality or data management are presented in this chapter. Eventually, three famous social networks are presented to highlight their privacy problems and solutions.

Contents

2.1 Criteria for Privacy Analysis 27
2.1.1 Orientation 27
2.1.2 Architecture 27
2.1.3 Privacy Enforcement 28
2.1.4 Strength 28
2.1.5 Retransmission and Reuse 29
2.1.6 Genericity 30
2.2 A Survey of Privacy Enhancing Technologies 30
2.2.1 Asymmetric Encryption 31
2.2.2 Access Control 32
2.2.3 Anonymity 34
2.2.4 Privacy Policies and P3P 37
2.2.5 Hippocratic Databases 39
2.2.6 Hippocratic Multi-agent Systems 41
2.2.7 Digital Rights Management Systems 43
2.2.8 Sticky Policies 45
2.2.9 Privacy Preserving Trusted Third Parties 47
2.3 Privacy in Social Networks 48
2.3.1 Facebook 48
2.3.2 Diaspora 50
2.3.3 Google+ 51
2.4 Chapter Conclusion 52

2.1 Criteria for Privacy Analysis

By looking at the different approaches to information control, we have been able to extract characteristics. In order to highlight the differences between the different approaches to information control, we present these characteristics and use them as analysis criteria. This section defines and explains each criteria.

2.1.1 Orientation

As stated in the previous chapter, two main orientations of privacy exists: vertical privacy and horizontal privacy.

Solutions addressing the problem can be made to handle horizontal privacy only, vertical privacy issues only, or both at the same time.

Most solutions only handle the vertical privacy problem. Horizontal privacy has been studied more recently. Horizontal privacy violations can happen when a user gains access to another user information, when he should not. Some solutions that handle horizontal privacy rely on central authorities, therefore the vertical privacy problem has to be addressed at the same time. Otherwise, the central authority would be able to read user information as it passes through.

2.1.2 Architecture

Another question is to know the architecture on which the solution can be applied. Can the solution be applied to decentralized networks or does it needs a central authority?

The solution architecture could be:

- centralized: a central node implements the solution,
- distributed: processes are distributed to multiple entities but a central authority is required for controlling their execution,
- semi-centralized: local authorities are used instead of one for the whole network in the case of a centralized architecture,
- decentralized: entities (*e.g.*participants) are in charge of the solution.

The more the solution is decentralized, the less it is likely to generate vertical privacy issues.

2.1.3 Privacy Enforcement

The privacy enforcement criteria describes the class in which belongs the technique used to enforce privacy. Example of classes are: cryptography, trusted computing, etc.

Apart from the technique, the enforcement is characterized by being *a priori* or *a posteriori*.

The enforcement can be done *a priori*, by preventing a user from doing a violation. To bring in place such a solution, having control over user actions is the most common option. It is therefore physically impossible for the user to make a violation.

Another way to enforce privacy is to do it after the violation has been done: *a posteriori*. Threatening users of sanctions if they make violations is quite common. The advantage of this approach *a posteriori*, is that the system does not hold control over the agents but only over the sanctions. The agents have more freedom than in the *a priori* solution. The main drawback of the solution is that agents make violations before being sanctioned, which is impossible with an *a priori* approach.

Privacy enforcement can be constrained by the system architecture or the solution architecture. An *a priori* control will be very hard to bring up in a decentralized environment with autonomous agents.

2.1.4 Strength

The strength of a technique represents the level of protection it offers to users for their privacy. It describes how hard it is to make a privacy violation in a system protected by the given technique. A very strong technique makes a violation very hard to occur.

It is required to distinguish between the protection provided vertically and horizontally. Most of the time, a very strong horizontal protection relies on a central authority that handles all the data, thus creating vertical privacy issues and therefore a poor vertical strength. This is useful to show that some techniques, providing a strong protection against horizontal violations cannot be used if the necessity of protecting vertical privacy is taken into account.

Strength will range, for both vertical and horizontal privacy, from very low, low, medium, high, very high:

- very low: violations can be made and there is a very little chance of being punished,
- low: violations can be made and are sometimes punished,
- medium: violations are taken into account and more or less efficient measures are taken,
- high: violations are taken into account and prevented after a short time,
- very high: violations are almost impossible

In short, this criteria describes how strong is the system for preventing violations.

2.1.5 Retransmission and Reuse

Approaches to privacy preservation sometimes take into account retransmission (or reuse) of information. Some other approaches lose track of an information once it reaches the recipient, clearing the path for further violations. Moreover, privacy preservation techniques often consider that the information being sent by a given agent is information regarding himself. In that case, it means that the technique does not consider retransmission issues.

The retransmission problem can be:

- ignored: the messages can be retransmitted without the violations being detected,
- considered: actions are taken to prevent violations occurring from retransmissions,
- protected: retransmission is prevented or severely punished.

This problem can occur both in vertical privacy and horizontal privacy situations.

2.1.6 Genericity

The real problem is not preventing privacy violations. In fact, the real problem is allowing users to communicate while preserving their privacy. In other words, one can easily provide the system with an approach that prevents any privacy violation from happening if all communications are made impossible. If there is no communication, there is no privacy violation.

Some techniques, to offer a better privacy preservation, bring restrictions on the information it is possible to share in the system. It is important to highlight this criteria as some privacy solutions may be specifically designed for a given application or a specific task. Those solutions should not be mistaken as general purpose solutions.

Moreover, some approaches come up with a lot of restrictions to improve the strength of their privacy enforcement techniques, most of the time at the expense of their genericity.

The genericity often balances with the strength of the protection. It reflects the range of applications that could be protected by a given approach, it can be:

- very low: restrictions in place constrain the possibilities so hard that it is barely impossible to communicate,
- low: restrictions are strong enough to prevent a large range of applications to be deployed on top of the system,
- medium: there are restrictions, but it still allows a wide range of application to be deployed,
- high: restrictions are light, almost all applications can be deployed,
- very high: restrictions are so weak that it is possible to deploy any kind of application.

2.2 A Survey of Privacy Enhancing Technologies

This section presents different privacy enhancing technologies among the most commonly used. Techniques are checked against the predefined criteria in order to highlight differences between approaches.

2.2.1 Asymmetric Encryption

Asymmetric encryption allows to make information unreadable by anyone but the recipient. In public key algorithms, a public key (known to all) is used to cypher messages, and a corresponding private key (known only to the recipient) is used to open the cyphered message. The most used algorithm is the Rivest Shamir Adleman (RSA) algorithm [Rivest 1978].

2.2.1.1 Description

Asymmetric cryptography can be used to protect privacy either vertically, horizontally or both. Cryptography allows to cypher information in such a way that it is only readable by those who possess the key. Using public key cryptographic algorithms allows user to communicate with each other through the system without being eavesdropped upon by the system. This is the technique used by Pretty Good Privacy (PGP) [Zimmermann 1995], for email transactions. It protects recipients from being eavesdropped upon by the email service provider or other users. Messages are encrypted using the public key bound to the receiver email address. The authenticity of the key/email binding is established by a web of trust. This decentralized trust model contrasts with the public key infrastructure (PKI, centralized) which is currently the most used alternative.

While the encryption may sound good, this can be a problem if the system is a service provider. In that case it must be able to gather information from the messages that goes through to provide the service. There is therefore a strong limitation on communications: if the system needs an information, then it must not be encrypted. Case in point if the system is an email service provider, while the content of the message can be encrypted, the recipient of the message must not be encrypted to allow the service provider to deliver it.

At the end, encryption protects transmission and storage. But once the message is opened using the key, restrictions do not apply anymore and the message can be freely used and retransmitted.

2.2.1.2 Classification

- **Orientation: vertical and horizontal** As seen, asymmetric cryptography can be used both to address vertical and horizontal privacy issues.

- **Architecture: decentralized** Public key cryptography relies on a web of trust to bind key to emails. This is decentralized and therefore it can be applied in decentralized architectures.
- **Enforcement: a priori, Cryptography** Violations in the system are almost impossible as long as the private keys are kept safe from intruders. The enforcement is made a priori by cryptography.
- **Vertical Strength: very high** Encrypted data is impossible to read from the service provider point of view if data is being sent between users.
- **Horizontal Strength: very high** Encrypted data is almost impossible to read without the key.
- **Retransmission: ignored** Once a message has been sent, all control is lost, and it could be retransmitted by the recipient.
- **Genericity: low** Encrypted data is unusable except for the recipient. Therefore, if any entity needs to handle the data in some way, issues arise. For example, encrypted data cannot be indexed in a database. Another issue is that once the package is opened, all control over the data is lost and it can be reused or retransmitted. The genericity is therefore quite low.

As a conclusion, asymmetric encryption provides reliable a priory protection for transmissions and data storage. It protects data from an eavesdropper or an intruder. Nevertheless, no measures protects the information from the receiver misbehavior.

2.2.2 Access Control

Access control is probably the most used technique for restraining access to resources and therefore, to information.

2.2.2.1 Description

One of the first models for information control was not exactly for privacy. The Bell-Lapadula model [Bell 1973] implements access control for classified information, for governmental or military applications. Security levels are defined

for information and clearance levels for users. The system is considered to be trusted (no vertical privacy issues). A user cannot read information at an upper level (no read-up) and cannot write at a lower level (no write down). The "no read-up" policy prevents a user from obtaining access to an information requiring a higher security clearance. The "no write-down" prevents higher clearance users to cause leaks of information at lower levels. In this model, everything relies on the system, the system choses the information security level and user clearances. Users must log into the system and are given their corresponding, predetermined, clearance.

Other access control techniques have showed up. The Role Based Access Control (RBAC) allows access to resources depending on the role the agent plays in the system [Ferraiolo 1995]. Its derivatives are used in social networks like Facebook as discussed in section 2.3.

Purpose Based Access Control (PBAC) [Byun 2005] has been defined more recently in order to take into account the purpose for which the information is requested. When information is required, a verification is made to check if the purpose is legitimate.

2.2.2.2 Classification

- **Orientation: horizontal** The purpose of access control is to control the access to the information of certain users from other users. The system stores and has access to user information. Vertical privacy is not protected.

- **Architecture: centralized** Access control works because the system is able to prevent/allow access to certain resources. A central authority is therefore required.

- **Enforcement: a priori, by access control** The system prevents violations by prohibiting access to resources by users.

- **Vertical Strength: very low** Data is stored by the service provider. This is a problem as he can access user data.

- **Horizontal Strength: high** Access to information is highly controlled, preventing most violations. Nevertheless, some weak points can be ac-

cessed by determined attackers, such as the communication if they are not encrypted.

- **Retransmission: ignored** Most of the time, retransmission is not taken into account. Once users have obtained access to the requested information, they can forward it without restrictions.
- **Genericity: very high** Few constraints are brought into place, therefore lots of applications can be protected using access control.

In fact, access control is a general-purpose information control technique. It is not specific to privacy and may lead to inaccurate handling of privacy. Typically, the most used social networks were forced to bring into place finer-grain controls regarding relationships between people, or regarding privacy "circles"‘. Eventually, one should note that access control only works in a centralized application. Everything relies on the central authority, making the vertical privacy issue a prominent problem.

2.2.3 Anonymity

Another technique for protecting privacy consists in breaking the link between the data and the user it concerns. This is called "anonymity" as the most trivial way of doing so is removing the name of the user from the data.

2.2.3.1 Description

Sometimes it is critical to store information while being unable to link that information to a given user. A very common example is the one used in medical studies from human subjects. The name of subject is most of the time replaced by a number, in such a way that health related information cannot be linked to the real person.

In fact, this is not that easy to break the link between specific data and a given person. Data is kept in such a way that it is possible to know that subject #42 is 26 years old man (born march 20^{th} 84), was diagnosed with a brain tumor, is allergic to penicillin, lives in area with ZIP #... This is interesting to run statistics or data mining algorithms over the data. The sample data presented above seems anonymous, at least it does not contains the name of the subject. In

fact, L. Sweeney presents in her article a tremendous experience [Sweeney 2002]: medical data, collected by states, contained attributes including ethnicity, medication, diagnosis, ZIP, birth date, sex... This data, considered to be anonymous, was provided to researchers and sold to the industry. For twenty dollars, L. Sweeney also purchased the voter registration list for Cambridge, Massachusetts. The two databases could be linked by ZIP code, birth date and sex allowing to link medication, ethnicity, diagnosis to particularly named individuals (see Figure 2.1).

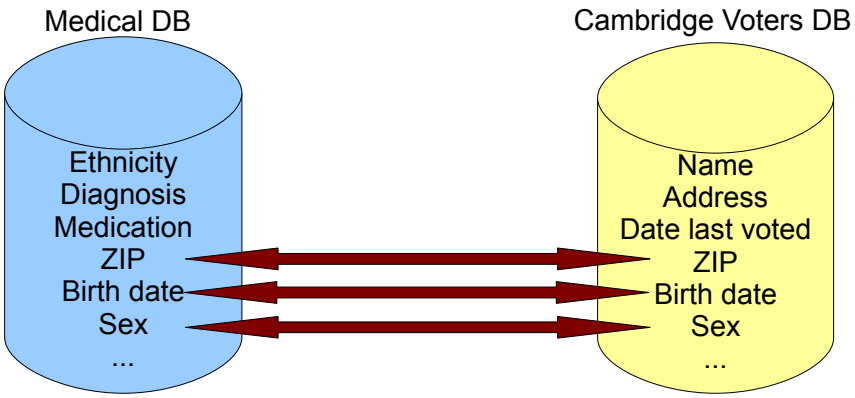


Figure 2.1: Linking databases to achieve re-identification

L. Sweeney provides a formal protection model named k-anonymity. "A release provides k-anonymity protection if the information for each person contained in the release cannot be distinguished from at least $k-1$ individuals whose information also appears in the release." To achieve k-anonymity, using the given example, ZIP codes can be altered to keep only the first numbers. The same can be done with birth dates, by keeping only the year of birth. Of course, information is lost during the process.

In the same vein, D. Abril et al. provide an algorithm for protecting confidential information [Abril 2010]. When confidential documents are stored, it is useful to index them based on a list of keywords. Those keywords can be automatically extracted from the documents. The problem is that these specific keywords may reveal confidential information. By using a method called semantic

micro-aggregation, keywords are replaced by more general terms from an ontology (*e.g.* 'beetle' is replaced by 'insect'). Once again, information is lost during the process.

A different anonymity technique is used in vehicular networks and other transportation systems. In order to pay the toll, cars have to identify themselves and provide to the toll server information about the car location. The problem, in fact, is that the car owner does not want the toll server to be able to trace his path, his speed and other characteristics. Therefore, data is transmitted anonymously to the server, and using cryptographic techniques, the car owner is able to pay the toll without the server knowing which car went where [Popa 2009].

2.2.3.2 Classification

- **Orientation: vertical and horizontal** These techniques can be used both for horizontal and vertical privacy. Nevertheless, there are often described in a vertical way in the literature.

- **Architecture: from centralized to decentralized** There is no restriction on the architecture on which it applies, anonymity could be enforced by agents as well as a system.

- **Enforcement: a priori, information removal** Information is removed to prevent the identification of the information subject.

- **Vertical Strength: from low to high** The strength of this technique is subject to variations. In fact, it depends on the quantity of data that is removed.

- **Horizontal Strength: from low to high** The more data is removed, the less it is likely to make violations occur.

- **Retransmission: protected** Retransmission is protected when the data has been made anonymous.

- **Genericity: medium** The biggest problem with this approach is that the protection is directly correlated with the amount of data that is lost. Moreover, applying these techniques on medias other than text is another

problem that requires specific approaches (blurring is often used but is poorly efficient).

This technique consists in removing bits of information to ensure privacy. Privacy comes at the cost of data loss.

2.2.4 Privacy Policies and P3P

Privacy policies inform the user on the conditions regarding the use and disclosure of his data.

2.2.4.1 Description

In general, websites collecting user information declare privacy policies. Privacy policies are a description of how user data will be used, processed and stored on the service provider's servers. They are often long and contains juridical terms, as shows the example provided in Figure 2.2. Therefore, as it has been noted in a study about Facebook, most users (77% of the respondents) do not take the time to read them [Acquisti 2006].

Moreover, privacy policies are not enforced a priori. If websites do not respect their policies or provide illegal policies, lawsuits can be filed against them. A good case study of a compagny violating its own privacy policies is given by Anton et al. [Antón 2003]. They study the JetBlue case, where JetBlue Airways Corporation gave away the travel records of five million passengers to a private US-Department of Defense contractor.

Eventually, users do not rely on technologies to enforce their privacy, they rely on trust towards the website compliance with its policies and the law.

The Platform for Privacy Preferences, P3P [Reagle 1999], is an extension to the privacy policies created by the World Wide Web Consortium (W3C). Instead of writing policies in juridical terms, they are written in XML. It allows user to declare, in their web browsers, their preferences upon information retrieval, handling and storage. Then users policies are automatically checked against the privacy policies declared in the XML of the website.

Just like privacy policies, P3P are not enforced a priori. It means that a website can declare that it will not store your email for longer than two weeks,

> We may provide information to service providers that help us bring you the services we offer. For example, we may use third parties to help host our website, send out email updates about Facebook, remove repetitive information from our user lists, process payments, or provide search results or links (including sponsored links). These service providers may have access to your personal information for use for a limited time, but when this occurs we implement reasonable contractual and technical protections to limit their use of that information to helping us provide the service.

Figure 2.2: An excerpt of the facebook privacy policy (August 2011)

but can do the opposite in reality. The same way, as P3P are just "computer readable" policies.

Figure 2.3 presents a P3P policy. P3P policies are, like privacy policies, a list of statements. Statements describes for a given set of data (<DATA–GROUP>): the purpose for which it is stored (<PURPOSE>), the duration while the data is kept (<RETENTION>) and the entities that will access the data (<RECIPIENT>).

2.2.4.2 Classification

- **Orientation: vertical** This technique is not really suitable for horizontal situations because legal actions do not weight the same on individuals as they do on corporations.
- **Architecture: centralized or decentralized** There is in fact no real need for centralization regarding this technique, even though most approaches are centralized.
- **Enforcement: a posteriori, by law** Violations are punished afterwards using legal actions.
- **Vertical Strength: low** The privacy protection is low. First, a violation has to happen. Second, the victim has to fill a complaint. Thus, it requires an effort from the subjects to detect and punish violations of their privacy.
- **Horizontal Strength: n/a**

```
<POLICY name="sample"
    discuri="http://www.example.com/cookiepolicy.html"
    opturi="http://www.example.com/opt.html">
    ...
    <STATEMENT>
      <PURPOSE><admin/><develop/><pseudo-decision/></PURPOSE>
      <RECIPIENT><ours/></RECIPIENT>
      <RETENTION><indefinitely/></RETENTION>
      <DATA-GROUP>
        <DATA ref="#dynamic.cookies">
          <CATEGORIES><preference/><navigation/></CATEGORIES>
        </DATA>
      </DATA-GROUP>
    </STATEMENT>
    <STATEMENT>
    ...
    </STATEMENT>
</POLICY>
```

Figure 2.3: A P3P policy

- **Retransmission: considered** Retransmissions are considered. It is often said, in policies, to whom it is possible to forward the data. But it is not really considered that users can be providing information from which they are not the subjects.

- **Genericity: very high** This protection does not constrains the messages at all, therefore it can be used in a wide range of applications.

Privacy policies are easy to bring in place. What is hard is to ensure they do not get violated. Moreover, the data provider is the one deciding the policies, when it should be the users' right to condition the use of their information.

2.2.5 Hippocratic Databases

The Hippocratic Oath, well known to physicians, constrains the information a physician can share about someone. The principles of the Hippocratic Oath are applied to databases to ensure good practices in information processing and storage.

2.2.5.1 Description

Based on the principle of the Hippocratic Oath, that guides physicians practices regarding information about their patients, R. Agrawal defines a system of Hippocratic Databases [Agrawal 2002]. Information storage in those databases follows ten principles which are the following:

- **Purpose Specification** The purposes for which the information has been collected shall be associated with that information.
- **Consent** The purposes shall have consent of the donor.
- **Limited Collection** Collection shall be limited to the minimum necessary for accomplishing the specified purposes.
- **Limited Use** The database shall only run queries consistent with the specified purposes.
- **Limited Disclosure** Information shall not be communicated for purposes other than the ones consented by the donor.
- **Limited Retention** Information shall be retained only as long as necessary for the fulfillment of the specified purposes.
- **Accuracy** Information shall be accurate and up to date.
- **Safety** Information shall be protected against theft and misuse.
- **Openness** A donor shall be able to access all information about the donor.
- **Compliance** A donor shall be able to verify compliance with the above principles.

Those principles are enforced on the database by the system. R. Agrawal provides directions in order to create a Hippocratic database. The database shall have modules that ensures that the ten principles are respected. For example, there is a *Data retention manager*, in the database, that deletes information to enforce the sixth principle. For each principle, a module ensures that it is enforced by the database.

2.2.5.2 Classification

- **Orientation: vertical** The purpose of this technique is typically to protect users privacy from organizations.
- **Architecture: centralized** This technique is developed for centralized systems, but it can be adapted to decentralized systems as presented in section 2.2.6.
- **Enforcement: a priori, by control over data** The system implementing the Hippocratic database has control over the data. Thus, privacy is enforced by directly applying the ten principles over the data.
- **Vertical Strength: medium** The approach is interesting and sound. Nevertheless, everything relies on the database provider. While tackling the vertical problem, it is generally to protect the users from the service provider (or an organization). Here, everything lies in the provider's hands.
- **Horizontal Strength: n/a**
- **Retransmission: considered** Principle #5 describes the conditions of a retransmission.
- **Genericity: high** This approach is highly usable for centralized systems. Only small constrains are put on the information to limit collection and to allow users to access their own data.

By comparison to P3P, this proposition constrains the data the service provider is allowed to collect, use and keep. On the one hand, it puts more control over the data as the ten Hippocratic database principles are meant to be implemented directly in the database. On the other hand, even when taking into account the *Openness* principle, it will be hard for the user to verify that the database complies with the ten principles.

2.2.6 Hippocratic Multi-agent Systems

Hippocratic Databases have been adapted to multi-agent systems by Ludivine Crépin in her thesis [Crépin 2009].

2.2.6.1 Description

Just like Hippocratic Databases, Hippocratic Multi-agent System [Crépin 2009] describes rules for the transmission, usage and storage of information. Agents define policies regarding the information they transmit, so that the receiving agent knows what he can, and cannot do with the data. It is based on nine of the ten principles of the Hippocratic databases, transposed to agents. The "accuracy" principle is rejected because she considers that agents may provide inaccurate information to protect their privacy.

The approach takes some strong assumptions. For example, agents only send information that belongs to them. In her PhD thesis, L. Crépin presents the use of social control as the best means to punish violations in the system. Social Control is discussed in the next chapter.

2.2.6.2 Classification

- **Orientation: horizontal** Hippocratic Multi-agent Systems are an adaptation of Hippocratic databases for horizontal privacy protection.

- **Architecture: Decentralized** There is absolutely no need for a centralized entity.

- **Enforcement: a priori, by control over data** Enforcement is made by applying the nine principles over the data to prevent violation. The possibility that an agent does not respect the principles is taken into account. Social exclusion is used against violators.

- **Vertical Strength: very high** As there is no need for a central authority, communications can be encrypted to prevent the system to eavesdrop on users.

- **Horizontal Strength: low** It is not clear how violations of the principles could be detected. In other words, an agent may decide to keep data he was suppose to erase and no one will detect it. Moreover, it is considered that agents decide on the conditions applying on the transmitted data. It does not consider that the data may not be their.

- **Retransmission: Considered** Retransmission is considered under the "Limited disclosure" principle. Nevertheless, it is only considered that agents send data regarding themselves.

- **Genericity: high** Restrictions are not overwhelming, and the protection can be used in lots of application. One should keep in mind the problem highlighted for the retransmission of third party information.

Hippocratic Multi-agent systems suffer more or less the same drawbacks than its centralized counterpart. The major drawback is that users cannot verify that others comply with the ten principles. These problems aside, they provide a guideline for privacy preserving agents' behavior.

2.2.7 Digital Rights Management Systems

Digital Rights Management Systems (DRM) have seen a surge when the music and movie industries tried to use this technique to prevent unlawful access and retransmission of media files.

2.2.7.1 Description

DRMs tackle the problem of retransmission that still exists when using cryptography. Thanks to solutions based on trusted computing, a centralized architecture has control over users and prevent them from retransmitting the information they receive.

DRMs have been designed to protect data from being retransmitted or copied, like movies, music, documents. By extension, DRM have been studied as a means to protect privacy [Korba 2002] with limited success. As we shall see in sections 2.2.8 and 2.2.9, DRM have been used in different privacy enhancing techniques as a tool among others.

DRMs rely on a trusted computing architecture. A special software is provided to the user. The software authenticates with a central authority, receives a cryptographic key that allows it to open the requested data. The information is imprisoned inside the software as it is deciphered on-the-fly. This way, it is virtually impossible for a user to copy the data or transmit it unlawfully. Figure 2.4 presents a simplified view of a DRM system.

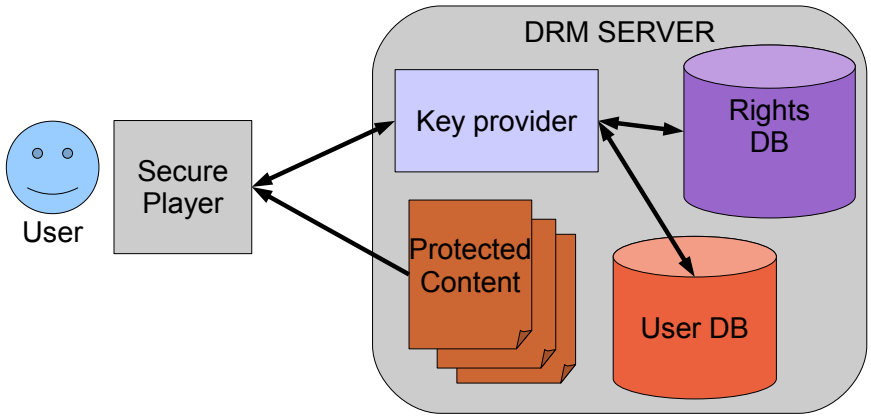


Figure 2.4: A simplified view of a DRM system

The problem with DRMs is that they are intrusive, because the user must install the provided software on his computer. In the end, users do not like being constrained. They find ways outside of the system when it is too constraining. As James Grimmelman states: "If we've learned anything at all from the DRM wars, it's that technical controls are rarely effective against a person genuinely determined to redistribute information they've been given access to. [Grimmelmann 2008]".

As an example, when music is distributed with DRMs, it is not possible to copy it or even play the music except with the provided software. The software does not allow to copy the file, but even a computer rookie will be able to forward the sound from the speakers stream to the microphone stream. It will therefore be possible to copy the protected music at the cost of a loss in quality.

2.2.7.2 Classification

Trusted computing techniques require a central authority that gives credentials to users. Therefore, it is centralized. The system is able to read all data that goes through, vertical privacy is not taken into account. Users must install trusted software and the provided information can only be used inside a *sandbox*, therefore, the genericity is quite low. Retransmission is meant to be impossible.

- **Orientation: horizontal** The very goal of DRMs is to prevent users to

communicate data they do not own. It is to prevent horizontal violations.

- **Architecture: distributed or centralized** At best, the trusted computing architecture providing the cryptographic keys is distributed, but most of the time it is centralized.
- **Enforcement: a priori, by cryptography** Data is only accessible to users that have been given access to. The information is constrained inside a "sandbox", most of the time a software provided by the service provider.
- **Vertical Strength: very low** Once again, as both the data and cryptographic keys lies in the hands of the service provider, it is not possible to prevent them from accessing user data.
- **Horizontal Strength: very strong** Except when users find their way out of the system, DRMs are very hard to circumvent as they are based on state of the art cryptographic techniques.
- **Retransmission: prevented** Retransmission is prevented by the system. Information may be retransmitted through the DRM system as long as the user rights allows it.
- **Genericity: low** The constraints are so hard that the genericity becomes very low. For example, as it is necessary for all participants to use the DRM software, it dramatically reduces the possibility of openness.

DRMs are a strong technique for information control, and by extension, privacy preservation. This solution requires a central authority and the use of a given software for all participants. This can be a problem when openness and decentralization are requirements.

2.2.8 Sticky Policies

When information is retransmitted, one wants that the policies regulating the use, retention and retransmission of information to be forwarded with that information. The problem is that it is difficult to guarantee that the policies will stick with the information once it is retransmitted. DRMs have been used to ensure that.

2.2.8.1 Description

One of the problems found in traditional approaches, is that policies can be defined for a transmission. But after the information is sent, everything lies in the receiver's arms. Sticky privacy policies [Mont 2003] tackles the problem of retransmission. If a policy has been defined, then it is permanently attached to the message. The problem is "how to prevent a user from accessing an information, without being able to detach policies ?". Sticky policies provide to the user means for viewing the information, while the information is kept in a sandbox (see section 2.2.7). Therefore, users are not able to detach policies or modify the information. The approach relies on trusted computing, DRM-like techniques. Therefore it requires a central trusted authority and intrusive software.

2.2.8.2 Classification

- **Orientation: horizontal** Sticky policies prevents horizontal violations.
- **Architecture: distributed or centralized** As with DRMs, a central architecture is required to hand over the cryptographic keys.
- **Enforcement: a priori, by cryptography** Data is locked in the sandbox and cannot be accessed or transmitted without the key.
- **Vertical Strength: low** If the system is well designed, the vertical strength can be improved by opening communication between agents instead of communication through the service provider.
- **Horizontal Strength: very strong** Like DRMs, Sticky policies are very hard to circumvent.
- **Retransmission: prevented** Retransmission is totally controlled by the policies. And the policies are enforced by the DRM system.
- **Genericity: low** The constraints are hard as once again, it is necessary to accept the DRM software for all participants.

Sticky policies are better than DRMs in the way that the approach is privacy-specific. Otherwise, it shares with DRMs most of its advantages and drawbacks (violations are made impossible but the system is very intrusive and decentralization is impossible to achieve).

2.2.9 Privacy Preserving Trusted Third Parties

Very few approaches use trusted third party for privacy preservation. This section present the approach that Guillaume Piolle developed in his thesis [Piolle 2009].

2.2.9.1 Description

G. Piolle provides a framework for handling sensitive information in multiagent systems. There are two main components: privacy-aware assistant agents and third party agents.

The privacy aware agents (PAw) are responsible for protecting the privacy of their user locally. But when data needs to be transmitted, the system has to rely on third parties. The PAw agent sends to the third party the information that needs to be processed and the restrictions that applies on this operation (policies). Information consumer sends complementary information and the operation to run on the data. Trusted third parties are implemented using trusted computing techniques. They are constrained in their actions in a DRM-like scheme.

Trusted third parties cannot send messages (to prevent retransmission), they can only answer predetermined sentences. For example, this framework could be used for credit card transaction, the user first sends his data to the third party. Secondly, the information consumer sends to the third party the operation to do: withdraw $40 from the user's account. Finally, the third party answers both of them that the operation is "OK" or "Not OK".

2.2.9.2 Classification

- **Orientation: horizontal** This privacy enhancing technique prevents privacy violating transmissions between users.
- **Architecture: centralized or distributed** As it relies on DRM systems, the architecture cannot be decentralized.
- **Enforcement: a priori, by design** Third party agents are designed in such a way that violations are almost impossible.
- **Vertical Strength: medium** As third party agents are distributed, it reduces the control that the service provider would have over user data.

- **Horizontal Strength: very high** Privacy violations are totally improbable.
- **Retransmission: prevented** Retransmission is impossible, as third party agents cannot communicate.
- **Genericity: very low** Because it is not really possible for agents to communicate, very few applications can be deployed on a system using this protection.

This technique renders violations almost impossible, but it comes at the cost of a poor genericity as the system is very constraining.

2.3 Privacy in Social Networks

The previous section has shown the different privacy enhancement technologies that have been developed in computer science. In this section, we show the privacy protection mechanisms currently being used in social networking services. In fact, most concerns regarding privacy actually come from social networking sites. In these websites, privacy protection has even become a commercial argument.

Three social networks are detailed along with their privacy protection mechanisms in this section. The three social networks that we selected are the three latest networks that have a specificity worth describing. Facebook is one of the most used social networks, it has been known for its privacy controversies since its beginning. Diaspora is an attempt at building a semi-decentralized social network. Finally, Google+ is Google's last social project, trying to take the leadership away from Facebook.

The criteria used in the previous section are used to evaluate the protection in place in each of the social networking services. The genericity is not discussed as the protection mechanisms are developed for a specific application.

2.3.1 Facebook

Facebook is a website that allows users to create a personal profile, add other users as friends and share information about themselves and pictures. Until quite recently, the only option to allow somebody to see what you share and to

be able to see what he shares, was to friend that person. Friending someone on Facebook is a bidirectional relationship, one has to ask and the other has to accept. After that, each one is *friend* with the other. The bidirectional nature of the relationship brings some problem, as some people may like you more than you do and thus, you may not want to share as much information as you would with your real-life friends. Another problem as this relationship is bidirectional: people do not want to turn off friend requests as it can be perceived by the other as being very rude. More recently, Facebook brought in place "friends lists" that allows to put friends in different categories and choose with which category you want to share information.

Based on the criteria from the previous sections, it is clear that Facebook is an horizontal and centralized system. The privacy protection mechanism enforces rules a priori. Rules are to allow only your friends to access your pictures (access control). The Facebook website has access to every bit of information uploaded on the website, and this is a serious vertical privacy issue. The horizontal strength is correct even with the issues we discussed here (bidirectional friendship links). Retransmission is not possible within the system, but it is very easy to take an information out of the system to share it (taking a screenshot for example).

Classification of Facebook protection mechanism:

- **Orientation: horizontal** The protection is in place to prevent violations between users.
- **Architecture: centralized** The architecture is centralized, the central authority is the social networking service provider: Facebook Inc.
- **Enforcement: a priori, by access control** The authority controls access to user data preventing violations.
- **Vertical Strength: very low** No security is brought in place to protect user-data from the social networking service provider.
- **Horizontal Strength: medium** The horizontal strength was poor until recently, with the *friend list* improvement that allows to differentiate between contacts.

- **Retransmission: prevented** In theory, retransmission is not possible on Facebook.

The protection developed on Facebook relies on access control. It is adapted to take into account some privacy-specific variables, like groups.

2.3.2 Diaspora

Diaspora provides the user with the possibility of creating *pods*. Pods are servers where user data will be stored. It means that users who do not want to rely on a central entity can create their own pods and connect them with other pods. Instead of having all user data centralized and controlled by a unique provider, the data is distributed among pods. Users connect to the pod on which they registered. That being said, diaspora allows to share pictures, information about oneself ... more or less the same way Facebook does. Friends on Diaspora are separated between *aspects* which are more or less the same as friend lists on Facebook.

The advantage over Facebook is that the vertical privacy issue is reduced by semi-decentralization of the data repository. As in Facebook, the system protects privacy by using access control. It prevents users from accessing data that has not been shared in an *aspect* they belong to.

Classification of Diaspora protection mechanism:

- **Orientation: horizontal** User to user violations are the main concern.
- **Architecture: semi-decentralized** The architecture is semi-centralized. There are multiple authorities, called pods, a user can connect to.
- **Enforcement: a priori, by access control** The authority controls access to user data preventing violations.
- **Vertical Strength: medium** The fact that the system is semi-decentralized allows user to choose between service providers. It increases the vertical privacy as there are more chances that among the different providers, one would be trustworthy.
- **Horizontal Strength: medium** The horizontal protection is more or less like in Facebook.

- **Retransmission: protected** Only *public* posts can be re-shared.

The big advantage that Diaspora has, is that the system is semi-decentralized. It allows user to choose between different service providers thus reducing the possibility of vertical privacy violations.

2.3.3 Google+

Google Plus is Google latest attempt at social networking. It allows people to share information about themselves, pictures, and so on. The big difference with Facebook is that the friend relationships are unidirectional. When a user A friends user B, it means that A allows his friend to see what he is sharing. To share its information with A, B must also add A as a friend. Users, when *friended* are added into *circles* more or less the same as Facebook *friends lists* and Diaspora *aspects*. Users do not know to which list they have been added to. Thus, the problem of refusing friends that happens on Facebook does not exist on Google Plus. Last thing, retransmission is possible on Google Plus. Warnings are issued when a user wants to do so as it may cause privacy violation if the information was not public.

Once again, access control is the main technique that is used for privacy protection. Users only see information that they have been given access too. Nevertheless, Google Plus is centralized. Therefore the vertical privacy problem is not solved.

Classification of Google+ protection mechanism:

- **Orientation: Horizontal** Horizontal privacy violations are targeted by the protection.

- **Architecture: Centralized** The architecture is centralized, the central authority is the social networking service provider: Google Inc.

- **Enforcement: a priori, by access control** Once again, the system controls user actions to prevent violations.

- **Vertical Strength: Very low** Just like with Facebook, all user data lies open to the service provider.

- **Horizontal Strength: Medium** Some improvements have been developed on Google Plus in comparison to other social networks, but the protection stills average.
- **Retransmission: Prevented** Retransmission is possible within the system. Warnings are issued when a private information is about to be retransmitted.

The unidirectional nature of friendship links, the privacy circles and the retransmission warnings makes Google Plus protections slightly better than its counterparts. Nevertheless, vertical privacy stills a major issue.

2.4 Chapter Conclusion

In this chapter, we have described the solutions that are the most used to protect privacy in a different range of applications. The last section focuses on social networks, as most of the concerns regarding privacy are currently coming from this direction.

As the last section shows, social networking services all use access control to protect user privacy. They do not offer sufficient protection for vertical privacy. Most of the time, the system has full access over user data. Diaspora brings an interesting insight as it begins to decentralize the system to provide a better vertical privacy. All in all, it seems that social networking services are going towards a complete decentralization, as pointed out by Sung [Yeung 2009].

Our objective in this thesis is to provide a privacy protection solution applicable to decentralized social networks. This solution must protect users horizontally and vertically. It must also take care of the retransmission problem.

As we can see in the grid presented on the next page, there is sadly no solution among the ones that we presented that satisfies the *required characteristics*. Therefore, we have to define our own solution.

The next chapter reviews the different methods that can be used to control agents. This way, we shall be able to choose a method to control agents, preventing them from making privacy violations.

Name	Orientation	Architecture	Enforcement	V. Strength	H. Strength	Retrans.	Genericity
Asymmetric Encryption	V and H	DE	A	++	++	−	−
Access Control	H	C	A	−−	+	−	++
Anonymity	V and H	C or DE	A	+−	+−	+−	+−
Privacy Policies	V	C or DE	P	−	n/a	+−	++
Hippocratic DB	V	C	A	+−	n/a	+−	+
Hippocratic MAS	H	DE	A	++	−	+−	+
DRM	H	C or DI	A	−−	++	+	−
Sticky Policies	H	C or DI	A	−	++	+	−
Privacy TTP	H	C or DI	A	+−	++	−	−−
Required Characteristics	H	DE	?	+	+	+	+

- Orientation: H for Horizontal, V for Vertical
- Architecture: C for Centralized, DE for Decentralized, and DI for Distributed
- Enforcement: A for A priori, P for a Posteriori
- Vertical Strength, Horizontal Strength and Usability: ++ for very high, + for high, +− for medium, − for low, −− for very low
- Retransmission: + for protected, +− when considered, − if ignored.

Chapter 3

Controlling Agents in Multi-Agent Systems

Laws control the lesser man... Right conduct controls the greater one.
Mark Twain

The previous chapter explored the different techniques that can be brought to protect privacy. We have shown that no reliable technique exists to protect horizontal privacy in decentralized systems. Multi-Agent Systems (MAS) are composed by agents interacting with each other. MAS are perfectly suited for representing decentralized systems. This chapter explores how a MAS can be controlled. Our goal is to find a reliable way of controlling agents and apply it to privacy preservation.

An MAS is composed of independent autonomous entities. The techniques for controlling agents in such a system are different from the techniques that can be brought in place in centralized systems.

MAS are most of the time designed in order to achieve a given purpose. Goals can range from building a business to business cooperation network to building a privacy preserving peer to peer social network. For the system to achieve these goals, agents have to be controlled in order to make sure that they participate to the system goal or at least do not prevent it from being achieved. Norms dictate the *normal* behavior agents should have in the system.

Norms are expectations of behaviors of the participants in a society. Norms may be explicitly defined as it is the case for laws, or may be implicit as it is the case for the "common sense". These social concepts are borrowed from human societies and applied to agent societies.

Norms can be used to protect privacy. The agents have to be informed that the *normal* behavior in the system is to protect users privacy.

The first section of this chapter presents how norms can be defined or emerge in an agent society. The second and third section presents how norms can be

enforced, either by institutional control (section 2), or by social control (section 3).

Contents

3.1 Norm Description 58
3.1.1 The Nature of Norms 58
3.1.2 Norm Emergence, Creation and Spreading 60
3.2 Institutional Control 61
3.2.1 Regimentation 61
3.2.2 Communication Filters 61
3.2.3 Electronic Institutions and Organizations 62
3.3 Social Control 63
3.3.1 Trust 64
3.3.2 Reputation 66
3.3.3 Social Exclusion 67
3.4 Chapter Conclusion 68

3.1 Norm Description

This section presents the different types of norms that exist and their specificities (first subsection). The second subsection explains how norms can be defined, adopted by agents and how they can be created or emerge in the system.

3.1.1 The Nature of Norms

Norms express what is the *normal* behavior. In other words, the expected behavior in the system.

It has been noted by authors that norms can be of different nature: Norms can be predefined, can be defined by agents or can emerge as implicit norms. Norms can also be strong like laws or weaker like societal norms. Raimo Tuomela [Tuomela 1995] groups norms in the following categories:

- Rule norms, or **r-norms**, are strict and explicit norms. They are laws imposed by an authority that one cannot ignore. For example, it is forbidden by the French law to carry a gun.

- Social norms, or **s-norms**, are implicit norms. They represent the conventions or mutual beliefs about the right thing to do in the system. For example, a trustee should not deceive a truster.

- Moral norms, or **m-norms** appeal to one's conscience. Generally speaking, moral norms are the appliance of the golden rule: "*One should treat others as one would like others to treat oneself*". An agent should not try to cheat if he doesn't want to be cheated on.

- Prudential norms, or **p-norms**, personal implicit norms used to maximize one's utility. For example, one should not trust someone who has proven to be untrustworthy.

Sanctions For each type of norms, corresponding sanctions exist. R-sanctions punishes r-norms violations, s-sanctions punishes s-norms violations... As Tuomela points out [Tuomela 2000], r-norms can be enforced by s-sanctions as there is a corresponding s-norms resulting from the group belief that one should respect r-norms. For example, if there is a r-norm stating that "one should not

park on a handicapped parking bay", a corresponding s-norm will emerge (see below) as there is a mutual belief that it is forbidden to park in those bays. Therefore, violators can be punished both by r-sanctions (being fined by the police) and also by s-sanctions (being known in the neighborhood as someone who parks on handicapped parking bays). On the opposite, r-sanctions does not apply for s-norms violations. Obviously, social norms should not be sanctioned by law (r-sanctions). If someone does not reply when being saluted, he should not be fined. M-norms and p-norms can also be sanctioned by using positive or negative internal feedbacks. Case in point, the day this page was written, I had to park on a handicapped parking bay for one hour or so. Even if I did not get a fine (no r-sanction), no one saw me (no s-sanction), I was feeling bad about it and internally sanctioned myself (m-sanction). In this thesis, we do not focus on m-sanctions and p-sanctions as it is related to internal agent motivations.

Norm Evolution Also, norms can evolve from one kind to another. M-norms can become mutual beliefs, and therefore, s-norms. These s-norms can in turn be formalized and accepted by the authority as a rule (r-norm). This is how laws are voted in the real world: "I don't want someone to steal from me, therefore I shall not steal" (m-norm) becomes "One shall not steal" (s-norm), that in turn becomes the r-norm "It is forbidden to steal".

Norm Life Cycle Bastin Tony Roy Savarimuthu presents [Savarimuthu 2009] the norm life cycle, composed of four phases: creation, spreading, enforcement, emergence.

- The phase of creation is the first of the life cycle. In a word, it is the mechanism by which an agent comes to know what the norm of the society is [Savarimuthu 2009], norms are created as "proposed norms".
- The second phase consists in spreading the norm among agents. Norm spreading can be helped by mechanisms such as leadership, imitation, social power. The goal is that agents accept and adopt the norms.
- The third step is the one of norm enforcement: making agents respect the norms. The two next sections presents the application of r-sanctions by institutions, and the application of s-sanctions by social control.

- Finally, the fourth step is the one of norm emergence, that can lead to creation of new norms, henceforth closing the circle.

From our point of view, we can separate this cycle in two parts: the first of norm definition (emergence, creation and spreading) and the second of norm enforcement.

3.1.2 Norm Emergence, Creation and Spreading

Norms can be defined by different actors. They can be defined by the system designer, by special agents in the system or even by the agents themselves.

A common way of defining norms is the *off-line* approach. In this approach, norms are designed even before the system starts running. It is common that the normative control is hardwired in the system (agents cannot execute unauthorized actions in the system) or that norms are hardwired directly into the agents (agents cannot "think" of unauthorized actions). Usually, these are signs of a closed and centralized architecture. In fact, it can also be the case that these norms are predefined by the designer but simply socially enforced by the agents. This last solution works even in decentralized and open systems as far as every agent is provided with a "norm textbook" when joining the system.

The system can also be conceived in such a way that some special agents are in charge of defining norms. The agent may have the power to impose the new norms or to convince other agents to follow his norms. Some agents may in fact transform m-norms to s-norms or s-norms to r-norms. This is norm emergence: m-norms that have became a common belief are accepted as s-norms that can in turn become r-norms.

Agents can also come up with norms based on their observations of the *normal* behavior in the system. By observing how agents reacts, what is tolerated and what is not, agents can edict norms (firstly m-norms and then s-norms). This method of norm creation can sometimes collide with the previous one. In some works, agents propose a norm that is afterwards discussed through argumentation to make others adopt this norm. Network topology has a strong importance regarding norm spreading and norm emergence, as it has been noted by numerous authors [Delgado 2003, Villatoro 2009].

As it has been noted, norms always work in pair with sanctions. These sanctions can be brought into place by institutions (r-sanctions) or by the agents

(s-sanctions).

3.2 Institutional Control

Once norms are defined, control can be integrated in the multi-agent architecture in order to control agent actions. This integration can be at the agent level, preventing agents from making violations, at the communication level, controlling or preventing interactions, or at a middle level, by defining organization or institution to regulate agent roles and norm compliance.

3.2.1 Regimentation

The easiest way to make agents follow norms, is to prevent them from having the possibility to do otherwise. Norms can be hardwired either in the agents or in the system in such a way that violations are impossible. On an other degree, the system can have the power to prevent violations depending on what roles the agents are playing in the system.

The process in which agents actions are limited to force them to comply with the norms is called regimentation.

Regimentation based solutions may work in centralized approaches but are inapplicable in decentralized environments. In fact, regimentation can be used to help enforcing norms, but not as the main tool for norm enforcement. For example, after an agent has been detected as a norm violator and has been excluded from the system, regimentation could be used to prevent the agent to reenter the system.

3.2.2 Communication Filters

A few authors have implemented norms by means of communication filters. Among these authors, Julien Saunier and Flavien Balbo have defined an environment for regulating multi-party communications [Saunier 2009]. Their Environment as Active Support of Communication (EASI) provides a framework for regulating interactions and making agents context aware.

When an agent wants to communicate, it has to state its initiatives and the environment opens a communication channel. The environment provides infor-

mation about the context of the interaction and verifies that the communication follows the norms of the multi-agent system.

So far, their approach is to provide the system with a full availability of information to allow context awareness and regulation from the communication filters. Therefore the counterpart is a centralized environment.The authors are looking for solutions to allow openness and decentralization.

3.2.3 Electronic Institutions and Organizations

To make agents respect norms multiple organizational concepts have been adapted to multi-agents systems. Those solutions introduce norms, structures, organizations (MOISE+ [Hübner 2002b], OperA [Dignum 2010]) or electronic institutions (AMELI [Esteva 2004], MOISE Inst [Gâteau 2007]).

AMELI is a platform for electronic institutions [Esteva 2004]. Institutions are described by defining the roles and actions allowed between agents. The system relies on a centralized architecture. Moreover, the system is regimented as the supporting middleware rules out all interaction that does not conform to the norms. Huib Aldewereld proposed an extension to AMELI in order to make violations possible and implement sanctions mechanisms [Aldewereld 2007].

MOISE+ is a organizational model for multi-agent systems. In MOISE+, a multi-agent system is specified as an organization composed by three dimensions. The structural aspect describes the groups, roles and the link between roles in the system. It also specifies who can communicate with who, which roles cannot intersect (be played by the same agent at the same time). The functional aspect describes the functioning of the organization through organizational plans, called social schemes and missions. These social schemes inform the agents of the goals to achieve in order to reach the organization goal. The deontic aspect specifies the obligations and permissions relative to a certain role in the organization and to missions.

Research has been done to enable these systems in decentralized and open environments. MOISE+, in its first versions, relies on an organizational middleware (S-MOISE+ [Hübner 2006]) that is integrated in the environment. Rosine Kitio *et al.* "Gives the power back to the agents" [Kitio 2007] by distributing the organizational layer over agents and artifacts.

In MOISE+, tools provided by the organization (artifacts) always take part

in cooperations taking place in the organization. It allows the system to monitor and sanction norm violations. It is considered that all agents have to go through the organization to communicate and therefore, cannot communicate without being monitored. But in opened and distributed environments, this hypothesis is too strong. Agents may decide to communicate directly and bypass the organization. Therefore, if norms were to apply on communications, they could be easily bypassed.

3.3 Social Control

All the approaches presented in the previous sections rely on a control by an authority. The r-norms of the system are enforced by using r-sanctions. The problem of these approaches is that they rely on centralization or intrusion: agents must accept to have their privacy intruded by the system or system agents. Another example that relies on intrusion was defined by Amandine Grizard. Her system allows to achieves social order in peer to peer system [Grizard 2007]. The system is totally decentralized and agents are autonomous but every participant must accept to host a *controller agent* that monitors and reports violations. While it is not intrusive regarding agents internal implementation, the system is intrusive in means of surveillance as it observes every transmission.

To remove any intrusion, solutions have been found to put agents in charge of the normative sanctions in the system. These solutions have been borrowed, once again, from human behavior. They rely on trust and reputation (s-sanctions) to exclude norm violators.

In an open and decentralized system, agents can be developed by different providers and have potentially incompatible roles. In other words, they can have a selfish behavior, meaning that they will favor their own goals without taking into account other's goals or system integrity. This kind of agent can harm the system if it doesn't know how to react against such an agent. The problem raised by the possible presence of selfish agents, agents not following the system norms, leads to a trust management problem towards other agents. Grandison [Grandison 2003] defines trust management as "the activity of collecting, codifying, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relation-

ships for Internet applications". Such a decision must be used in addition to classical security techniques which insure authentication, confidentiality of information... but that cannot guarantee the behavior of transaction partners.

This section presents trust and reputation in multi-agent systems. It also describes how it can be used for implementing s-sanctions.

3.3.1 Trust

Different approaches of trust have been proposed. The interested reader could refer to Jordi Sabater's survey of trust systems [Sabater 2005]. As one can see in the survey, most models are based on game theory. Castelfranchi and Falcone did defined an approach based on social concepts, that works using beliefs instead of game theory inspired mathematical functions. During the beginning of this thesis, our team was implicated in the ForTrust project that aims at formalizing Castelfranchi and Falcone theory of social trust.

According to Castelfranchi & Falcone [Falcone 2001] (C & F), social trust relies on four elements: a truster i, a trustee j, an action α and i's goal φ. C & F propose a definition of trust based on four primitive concepts: capacity, intention, power and goal. They state that: "an agent i trusts an agent j for doing the action α in order to achieve φ" iff:

1. i has the *goal* φ;
2. i believes that j is *capable* of doing α;
3. i believes that j has the *power* to achieve φ by doing α;
4. i believes that j *intends* to do α.

To give an example of a trust model, we briefly present the ForTrust model proposed by Emiliano Lorini [Hübner 2009]. This model is a formalization of Castelfranchi and Falcone theory, it is realized in multi-modal logic (called $\mathcal{L}$) which combines dynamic logic and BDI.

That formalization points out the difference between *occurent* trust and *dispositional* trust. Occurent trust represents a trust decision "here and now". In this case, the truster i has goal φ and trusts j to do action α now to achieve goal φ. In this article, we will only use the occurent trust, defined this way:

$$\begin{aligned} Trust(i,j,\alpha,\varphi) \stackrel{\text{def}}{=} \\ Goal(i,\varphi) \wedge \\ Bel(i, Act(j,\alpha)) \wedge \\ Bel(i, Power(j,\alpha,\varphi)) \end{aligned} \tag{3.1}$$

That formalization uses the four primitive concepts by C & F, the predicate $Act(j,\alpha)$ meaning that j does α. It covers both capacity and intention: $Act(j,\alpha)$ is the case when j has the capacity and the intention of doing α. Predicate *Power* means that j has the power of achieving φ by doing action α.

Trust, as considered here, allows to represent how an agent can make a trust assessment towards another agent trusting that he will *act* in a certain manner to achieve a given goal. Nevertheless, Lorini and Demolombe [Lorini 2008] say we also have to consider trust in *inaction*, relating to the trust assessment made when an agent i trusts an other agent j so that j does not execute action α that can prevent i from achieving φ. Trust in inaction is defined as follows:

$$\begin{aligned} Trust(i,j,\sim\alpha,\varphi) \stackrel{\text{def}}{=} \\ Goal(i,\varphi) \wedge \\ Bel(i, \neg Act(j,\alpha)) \wedge \\ Bel(i, Power(j,\alpha,\neg\varphi)) \end{aligned} \tag{3.2}$$

Meaning that i trusts j not to do α when i has goal φ iff: i has goal φ, i believes that j has the power to prevent i from achieving φ by doing α, and i believes that j will not do α (he has no capacity or no intention to do so).

This model has been deployed and tested on activity logs from Wikipedia in order to detect incompetent or untrustworthy contributors [Krupa 2009]. On the online encyclopedia Wikipedia, that anyone can edit, some editors make wrong modifications. These modifications can be of different nature, to simplify we have:

- INT: a modification that does not reduce the quality of an article
- COR: a modification that needs to be corrected
- VAN: a vandalism that needs to be reverted

INT are what can be called "good modifications". COR are erroneous modifications, the editor wants to do a good modification but does not have the

competence for it, or simply fails. COR modifications can sometimes be seen as a s-norm violation, a violation of a norm that everyone respects event if the s-norm is not written. For example, respecting typographic rules. VAN are vandalisms that cannot be mistaken as a incompetence. Vandalisms are voluntary destruction of articles. Vandalisms can be seen as r-norms violations (Please refer to the original article for more details [Krupa 2009]).

On Wikipedia, before being blocked by an administrator (r-sanction), social sanctions (s-sanctions) are deployed, the violating editor is warned by other users and his modifications are reverted. If he does not get blocked yet, after doing a few norm violations, his modifications on the encyclopedia will be systematically reverted, socially excluding the user from the system.

3.3.2 Reputation

In multi-agent systems, and generally speaking in many of the open web applications, trust management is often handled by reputation mechanisms. Some of those mechanisms work in a centralized way, for example, by using recommendations and opinions of the website users (cf. eBay[1], Amazon[2], ...). These opinions can be presented as they are or interpreted by an aggregation function (e.g. Sporas [Zacharia 1999]). Other mechanisms, inspired by the multi-agent field, work in a decentralized manner (e.g. Repage [Sabater 2006]) by allowing each agent to evaluate locally its neighbor's reputation such that the agent decides whether to trust or not.

Reputation is useful as it allows agents to share their experiences. Case in point, imagine we have a 10 agents system and one of these agents is a malevolent agent (norm violator). If agent do not share experiences, no less than 9 violations (one with each agent) have to take place before all agents are aware that this agent makes violations. If they share their experiences, only one violation is needed for all agents to e aware of the problem. Once the violation is detected, the participant informs all agents of the issue.

The main drawback of this approach, is that one has to take into account that an agent can lie about another agent, *i.e.* say that he makes violations when he does not. Therefore, another trust aspect has to be deployed for the "trust in

[1] http://www.ebay.com/
[2] http://www.amazon.com/

recommendations".

Moreover, the ratio of malevolent and benevolent agents becomes of great importance when using reputation. If there is not enough benevolent agents that respect the norms and tell the truth when they recommend agents, the system will go down. Nobody will be able to trust no one anymore as the different recommendations that agents receive will be contradictory.

3.3.3 Social Exclusion

The powerful application of the s-sanction is the social exclusion. The smallest s-sanctions are made through reputation sanctioning. An agent that makes violations is detected by other agents and his reputation is downed as the agents cast reputation messages stating the violation that has been made. Agents receiving those messages adjust their trust levels towards the culprit, this is the first step of the s-sanction.

After a certain number of violations, some agents will stop trusting the violator and will therefore refuse interacting with him. This is the second level of s-sanction, the agent is partially ignored.

If he continues his wrongdoings, the violator ends up with no-one trusting him. Reputation messages has been sent, and no-one is willing to interact with an untrustworthy party. This is the last step of s-sanction. The agent is socially excluded. Even if there is no r-sanction banning him physically from the system, the agent has no point in staying as he is not able to interact with anyone. At this point, the s-sanction is almost as powerful as a r-sanction.

Eventually, some models will allow forgiveness in order to lower the s-sanction and give another chance to the agent.

A problem might be the white-washers, agents that manage to exit and reenter the system with a new identity and with a "washed" reputation. This is usually tackled in applications by using identifiers that require time or money to forge, making those maneuvers unprofitable. For example, ebay requires that when someone registers, he sends his credit card number. Another dissuasive option is to make newcomers "not really trustworthy". This is also what happens in ebay, where new sellers have a "0" reputation (not negative, not positive either). New sellers have more difficulty to sell than sellers with a good reputation.

3.4 Chapter Conclusion

In this chapter, we have shown how norms can be defined and enforced in multi-agent systems. In a perfect application, if we wanted to protect privacy, we would define universal r-norms (rules) for privacy preservation, and prevent agents from making violations. The systems we consider, in fact, are open and decentralized and the agents are autonomous (their actions cannot be controlled by an outsider). Therefore, an intrusive control is inapplicable and thus, r-sanctions cannot be brought into place. Last but not least, in a decentralized and open environment, the information each agent have about other and the system may be incomplete, agents may make mistakes. Forgiveness needs to occur.

For all the previous reasons, the obvious solution seems to define r-norms for privacy preservation that all agents are meant to know, and to enforce those norms with s-sanctions (social sanctions), using trust and reputation techniques. From what has been presented in Chapter 2, the theory of Contextual Integrity gives a good basis for socio-inspired privacy preserving norms.

The next part presents how Contextual Integrity can be transposed to multi-agent systems and how trust and reputation can be used to enforce norms for privacy preservation in decentralized and open multi-agent systems. The best is made to avoid the drawbacks of the approaches presented in Chapter 3.

Part II

The PrivaCIAS Model

Chapter 4

Introducing the PrivaCIAS Model

Animals are something invented by plants to move seeds around. An extremely yang solution to a peculiar problem which they faced.

Terence McKenna

In the first part of the thesis, we described work related to privacy that has been done in social sciences. We also presented the different techniques that have been defined to protect privacy in information technology. In the latest chapter we described the techniques that can be applied to control agents with norms.

At this point, we are able to give a sketch of the PrivaCIAS model, before going into more details in the next chapters.

This chapter introduces the PrivaCIAS model. The model we present protects privacy in multi-agent systems, by controlling agents through social control and uses the social theories that have been discussed in Chapter 1.

The first section recalls the problem we are addressing and the requirements that must be fulfilled by the PrivaCIAS model. The second part gives the general idea and mechanism of our solution. The third part gives an overview of the model and its components.

Contents

4.1 Requirements 73
4.2 Tackling the Problem 74
4.2.1 Sending Messages 75
4.2.2 Receiving Messages 76
4.3 Overview of the PrivaCIAS Model 76

4.1 Requirements

The problem we are addressing in this thesis is: "how to limit and prevent privacy violations in a decentralized and open system?".

In more details, we want to prevent privacy violations in a system which has the following properties:

- it is decentralized,
- it is open,
- it does not allow intrusive control,
- it handles retransmissions,
- it assists users."

Therefore, one of our requirement is to be able to provide a protection that works in decentralized environments. **Decentralization** means that it is not possible to rely on central authorities accepted by every agents in the system. It is also fairly possible in decentralized environments that no agent or entity have a global view of the system. Case in point, it is very common in peer-to-peer networks that agents have a local view of the system, i.e. they know only the agents with which they interact. This setting brings an important constraint, the system that we define must be able to work without a central authority, but also without having a global view of the system.

Another requirement is that we address **open systems**. In open systems, agents can leave or join anytime they want. Handling an open system means that it is not possible, for example, to predefine a hierarchy between agents before running the system, because it is impossible to know which agents are going to be part of the system and at what time. Having a close system would mean that, on the opposite, it is possible to do some calculations and to detect potential problems before runtime. Here, the system must be evolving and adapting itself at runtime.

A condition that often goes in pair with open and decentralized system is being **non intrusive**. In other words, the system we develop should not force agents to integrate a kind of spyware or control artifact that spies, constrains, or does any kind of internal manipulation of the agent beliefs. The problem of

being intrusive is that it will raise vertical privacy issues (see chapter 1) as the agent will be forced to integrate the intrusive element whether it is trusted by the agents or not. The problem should be tackled from the outside of the agent, by looking at the communications that are made by each agent.

We aim at protecting privacy. It has been shown in chapter 1 that a good privacy definition relies on contexts. Helen Nissenbaum points out [Nissenbaum 2004] that the binary definition of privacy (the public/private dichotomy) is not expressive enough. One of our requirements is to use a theory of privacy that relies on contexts: the theory of **contextual integrity**.

In chapter 2 we give an overview of the range of approaches that can be used to handle privacy. We remark that some techniques consider that agents only send information that belongs to them (or to their user). This is unrealistic from our point of view. Agents transmit information to others, therefore the receiving agent handles information that belongs to somebody else. Thus, we have to consider that agents retransmit information, in other words, they handle information that belongs to others. We also point out in this chapter that very few approaches allow to protect privacy both vertically and horizontally. We require a technique that handles **retransmissions**, and protects user privacy vertically and horizontally.

As we see in chapter 1, it is important to **assist users**. In fact, it has been noted by social scientists that people do not understand privacy. Assisting them reduces this issue. Eventually, the system that we develop aims at helping users make good decisions. The system should not take decisions in lieu of the users. By providing help to these users we can reduce their risk of taking bad decisions leading to privacy violations.

4.2 Tackling the Problem

In order to solve the problem that we describe in the previous section, we define a model for protecting privacy in an open and decentralized multi-agent system (ODMAS).

The previous section shows that it is impossible to have a global view of the system, a central authority or intrusive agents. Therefore, our model includes a definition of agents that assists users in protecting their and other's privacy.

Each agent uses its local perception to prevent privacy violations in the system. Instructions are given at the agent level to obtain a coherent privacy protection at the system level. Assistants agents assist the users when sending and receiving information to detect privacy violations.

To punish agents that make privacy violations, we bring norms into place. Norms are enforced by social control as it seems to be the only reliable approach that is non-intrusive and able to work in ODMAS (see chapter 3).

We describe an agent model including norms that agents must follow. This way, every user can choose whether to implement himself the model or to rely on software agent provided by a third party. As with protocols, it is not important who implements the protocol, what is important is to follow the specifications. It allows us to provide user with a non-intrusive protection method, because users can participate to the system as long as they follow the specification of the model. They do not require to integrate any kind of software that we will be providing.

We use the term agent to refer to the assistant agents. Each agent is controlled by its user. When we say that agents communicate, in fact it means that their users communicate together through their assistant agents. The assistants filters messages that comes in or out: when the user wants to send a message or when the assistant receives a message directed to the user.

The model acts on these two entry points to protect privacy:

- it prevents the user from making violation when sending messages,
- it spots violators and punishes them when receiving messages.

4.2.1 Sending Messages

When an agent has been requested to send a message by its user, it has to check if the message is going to trigger a privacy violation or not. The goal is to prevent the user from exposing himself to s-sanctions (social sanctions, see chapter 3) that would be taken if he was to make a privacy violation.

To be able to do so, agents must have a common description of *what is a privacy violation*. This description should allow agents to decide if a message is a privacy violation or not on a common ground.

Upon sending, the agent checks if the message is going to cause a violation. If it is the case, the agent warns the user that can decide to send or not the message.

In any case, if the message is to be sent, the agent prepares the message according to the framework specifications and sends it.

The users have the option to send the message even if a violation is meant to occur. It is up to the recipient to detect that violation and bring appropriate sanctions.

4.2.2 Receiving Messages

When the agent receives a message directed to its user, it checks if the message is triggering a privacy violation. If it is the case, the agent has to delete the message in order to prevent the *real* violation from happening (if the user has access to the message).

After detecting a violation, the agent has to communicate with other assistant agents to inform them of the violation. This way, assistant agents can implement s-sanctions against the violators.

For doing this, we propose a common definition of privacy and communication specifications. We also need a set of norms that describe *how to react to a privacy violation.*

To implement social control, agents will need to include a trust model.

4.3 Overview of the PrivaCIAS Model

The PrivaCIAS Model, for Privacy as Contextual Integrity in Agent Systems, is an agent-centered model for protecting privacy in ODMAS.

An overview of the model is presented on figure 4.1. Agents communicate in a peer-to-peer (agent-to-agent) scheme, by exchanging encrypted PrivaCIAS Messages over the network.

The model specifies an agent architecture and instructions to protect privacy. The structure of messages is defined in the *PrivaCIAS Language* module. Assistant agent includes two main features. The first one is the *CI Violation* module that says if a transmission is a privacy violation regarding the contextual integrity theory. The second one is the *Normative* module that defines the norms of the system and how to react to a privacy violation.

As we can see on the picture, the assistant agent and the trust model are represented in different colors depending on which agent they assist. This is done

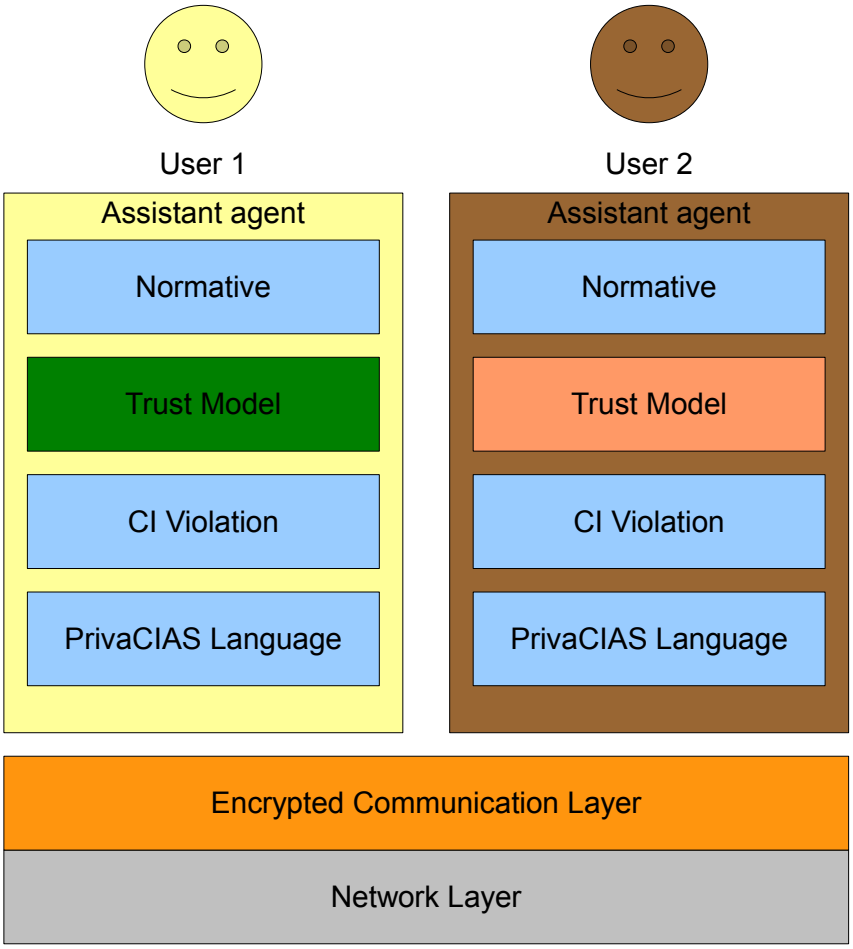


Figure 4.1: Overview of the PrivaCIAS Model

in order to highlight that agent implementations (and the chosen trust model) may differ from one user to another. What should not differ are the *CI Violation*, the *PrivaCIAS Language*, and the *Normative* modules.

To summarize, the only component that is shared between agents is the network layer, other components are implemented by the agents. Trust model and agent implementation may differ while the modules defined in the next chapters (PrivaCIAS language, CI Violation and Normative modules) must not differ from one agent to another.

This is an open and decentralized approach to privacy. The problem of preventing violations in ODMAS is a difficult problem as the technical solutions for privacy protection that exists cannot be applied (see chapter 2). Moreover, as we explain in section 4.2.1, it is not a good idea to try to constrain users a priori in this kind of systems. It is better to leave them with the possibility of making violations to punish them afterwards.

The next chapters explain how we solve the two main issues of the thesis:

- How to detect a privacy violation? (chapter 5)
  - We must adapt contextual integrity to agent systems to detect privacy violations (the *CI Violation* module)
  - Supportive Elements (structures, languages...) should be brought into place to support contextual integrity (the *PrivaCIAS Languages* module)
- How to make a coherent open and decentralized system that protects privacy? (chapter 6)
  - Give directives to agents on how to react to privacy violations (the *Normative* module)
  - Provide developers with a model of PrivaCIAS compatible agents (agent architecture guidelines)

Chapter 5

Appropriateness Laws

How can we expect another to keep our secret if we cannot keep it ourselves.
François de La Rochefoucauld

In this chapter we incorporate the theory of contextual integrity to the PRivaCIAS model and define related components. The *PrivaCIAS Language* and the *CI Violation* (Contextual Integrity) modules from figure 4.1 are the two components described in this chapter.

In our specification, it is crucial to have a common ground for agents regarding privacy. Agents need to be able to understand each other and have a common definition of what is a privacy violation. This is also the reason why we define a language to allow agents to discuss about privacy-related concepts.

This chapter is divided into three sections. The first section revises the Contextual Integrity theory to give the reader an overview of what has to be provided. The second section provides a set of structural elements in order to support contextual integrity and provide agent with a common language and definitions. In the third section, we formalize the Contextual Integrity theory to a set of three laws that we call the Appropriateness-Laws. The Appropriateness-Laws rely on the elements from the second section.

Contents

5.1 Embracing Contextual Integrity 81
5.2 The PrivaCIAS Components 83
5.2.1 Roles and Contexts 84
5.2.2 Message Structure 86
5.2.3 PrivaCIAS Predicates and Preferences Language 88
5.3 Formalizing A-laws 92
5.4 Example Scenario 93
5.5 Chapter Conclusion 96

5.1 Embracing Contextual Integrity

In chapter 1, we presented the Contextual Integrity theory. This theory describes the characteristics of a privacy violation (see section 1.3). Even though Nissenbaum describes the characteristics of a privacy violation, she does not define *what is a privacy violation.*

To embrace her theory in our model, we revise Nissenbaum's description to provide agents with norms. In other words, we define the logical conditions that allows to determine if a transmission is a violation or not: This set of norms is called the *Appropriateness Laws* (A-Laws). *An inappropriate transmission is a privacy violation.*

This section makes the first transition between Nissenbaum theory and its normative definition. We express norms informally, that are derived from the Contextual Integrity theory. It allows us to get an overview of the elements that are required to support these norms in our model. The second section defines those supportive elements and a language specific to our model. Relying on all those elements, the last section is able to propose a formal definition of the A-Laws.

We can define a transmission as appropriate if all of the following conditions hold, and inappropriate if one of the conditions does not hold:

1. Transmission context corresponds to the information nature.
2. Agents have a role within the transmission context.
3. Agents do not have incompatible relationships with the target.
4. The target's preferences[1] are respected.

In the previous definitions, we use the term *target* instead of Nissenbaum's term *subject* because a subject is directly related to the information, while a target may not even appear in the information. For example, if the information is a picture of Mr Smith with a woman who is not Mrs Smith, the subjects of the picture are Mr Smith and the woman, but the targets, the ones that can be harmed by the disclosure of the picture, are Mr Smith, Mrs Smith and the woman on the picture. Therefore, we believe that *target* is more versatile.

[1] Nissenbaum called these *terms of dissemination.*

Thereafter, we illustrate the 4 statements of appropriateness with examples:

1. *Transmission context corresponds to the information nature.* In the large sense, the context of a transmission can be seen as the environment where and when the transmission takes place. For example, personal health information corresponds to the medical context. We are in a medical context when a transmission occurs in a hospital for example. The idea behind this rule is that in some contexts it is inappropriate to share information from other contexts. It can be inappropriate to share personal information at work, or to share corporate information at a family dinner.

   This norm highlights two needs: the agent should be able to determine the context of the transmission and the information nature.

2. *Agents have a role within the transmission context.* Agents participating in the transaction should have a role associated with this context. For example, a medical doctor has a role associated to the medical context. Therefore, it is appropriate for him to receive medical information.

   To enforce this norm, agents must be able to know each other's roles.

3. *Agents do not have incompatible relationships with the target.* The target of the information and the agent receiving the information may have incompatible relationships. For example, consider the case of an agent A who has an illness, and an agent B who is both a medical doctor and A's boss. It may be inappropriate for B to know A's disease because those agents are having an "out of context" relationship (hierarchical relationship).

   As one can guess, this norm would be quite complex to put in place and would require dedicated research. For simplification reasons, in our model we use the fourth norm to express preferences preventing agents having incompatible relationships with the target from receiving the information.

4. *The target's preferences are respected.* If one of the targets of the information specifies preferences regarding the propagation of the information, it is inappropriate to violate those preferences. A target can specify that a message cannot be transmitted to a specific person even if the other norms are satisfied.

This norm requires the agents to be able to express and understand preferences, and to determine who are the targets of the information.

The third norm, which focuses on incompatible relationships, as we discussed, is supplanted by a specific use of the fourth norm. Therefore, the A-laws are the following:

1. Transmission context correspond to the information nature.
2. Agents have a role within the transmission context.
3. The target's preferences are respected.

These A-laws bring some requirements for the system. Firstly, we need to handle roles and contexts. Secondly, we need to define a structure for the messages that are exchanged in the system. Thirdly, we need to define a way of expressing preferences regarding message transmission.

5.2 The PrivaCIAS Components

According to the previous definition of appropriateness, for agents to be able to detect violations, we must allow them to:

- determine the context of the transmission and the information nature,
- be able to know each other's roles,
- be able to express and understand preferences,
- determine who are the targets of the information.

Therefore, we need a solution for handling roles and contexts. We also need to define the structure of messages and a common language for accessing roles, contexts, message structure and network structure in such a way that agents can understand each other[2] and also express privacy preferences that any agent can interpret.

All these elements are considered to be part of the PrivaCIAS model, that all agents must implement in order to understand others and to be understood.

[2]One should note that we propose a model, but agents participating in a PrivaCIAS powered system can be developed by multiple authors using various programming languages. Hence, this is why a common language is absolutely required.

5.2.1 Roles and Contexts

Roles and Contexts can be handled by different techniques. We define in this section a decentralized role managing system.

First of all, for simplification reason, we consider that roles and context can be unified: We call them *role-contexts*. Therefore, only one role can be played (by multiple agents) in a given context. Agents that plays this role will be said to *belong to the role-context*.

A role-context is both a tag that marks an agent as belonging to a group and a tag that agents in the system can give semantics to. The precise notion of role-context is implementation dependent.

For example, if we deploy the model in a company, there will be a role-context "board of directors" to which the CEO, CFO and other members of the board will belong. Employees will give to this role-context the real-life semantic it has: "the board of directors is a body that oversees the activities of the company".

An agent in the system can create a role at any time. An agent is responsible for the set of roles he creates and therefore, each role-context in the system is associated with the unique identifier of its responsible. Using our previous example, we can therefore express the role-context as follow: `boardOfDirectors@CEO`. It means that this role-context "boardOfDirectors" is owned by the user whose unique identifier is "CEO".

The agent that creates a role-context is responsible for registering other agents to it. The agent maintains the list of agents that belongs to the role-contexts he creates.

Any agent can ask the owner of a role-context if an agent belongs to it or not. Using our example again, an employee could ask the CEO "does the CFO belongs to boardOfDirectors@CEO?".

This allows agents to determine if a given agent belongs to a specified role-context.

The next step is to allow agents to detect if the transmission role-context corresponds to the information nature. It is needed to ensure the first Appropriateness-Law: *Transmission context must correspond to the information nature.* For example, information regarding the next meeting of the board of directors corresponds to the role-context boardOfDirectors@CEO.

In our framework, the sender of a message declares the transmission context

in which the message is transmitted. The A-laws verify that the receiving agent plays a role in that context and that the context corresponds to the information nature.

We consider that there exist agents in the system that are able to determine the corresponding role-context of a given piece of information. And those agents may not be the same depending on the piece of information. This is the reason why the role-context should have semantics from the agent point of view. In our example, the board of directors has a clear semantic for employees, therefore, some agents are able to determine if an information corresponds to the role-context boardOfDirectors@CEO.

If the role-context has no semantics, most agents will not consider that the *Transmission context corresponds to the information nature*, and therefore, they will declare the transmission as inappropriate.

As we are defining assistant agents, we can rely on the user if he has the required expertise, or analyze the role-context that other agents believed to correspond to the information nature. This remark brings another requirement: we have to keep track of all role-contexts that have been declared by agents during the previous transmissions.

In the case the user is not able to determine the context, we use a trust model to evaluate the previous role-context declarations. We consider the role-contexts that have been declared by trustworthy agents[3].

The same process is put in place to determine who are the targets of a given information.

In the last paragraphs, we discovered that we needed to keep track of all role-contexts defined for a given piece of information. We need to define a message structure that allows to carry this data along with the piece of information that is transmitted. Also, for using reliably a trust model, agents must be able to identify the author of a given role-context declaration. Therefore, all declaration must be "signed".

The next section clarifies the structure of messages, and how role-contexts and transmission history are stored.

We first define the structure of messages. It allows us to propose a Pri-

[3]In real applications, a reverse solution can be deployed: assistant agents tries to determine automatically the context and the user *flags* the message if the context does not corresponds to the information nature.

vaCIAS specific language for accessing the different components of a message, role-contexts and so on. Then this language is used to express preferences and express A-laws formally.

5.2.2 Message Structure

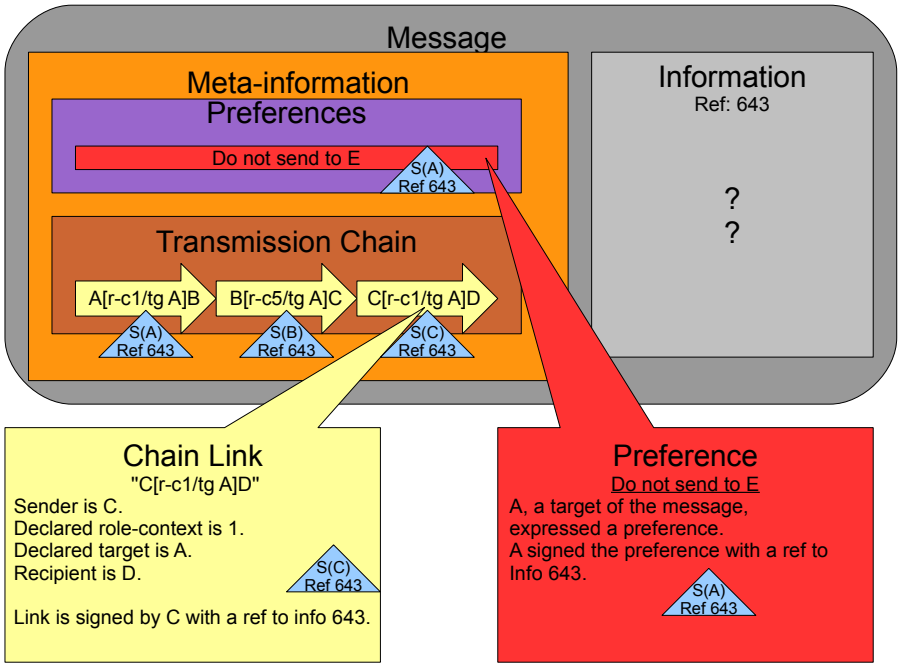


Figure 5.1: Message structure

Users exchange **information** encapsulated in a **message**, see figure 5.1. Information is raw data. We do not make assessment about the structure of the information and leave it free. Therefore, we do not constrain the types of information that can be carried in the messages. A message encapsulates information and meta-information described thereafter.

Firstly, from a given piece information, we compute a unique reference that allows to refer unambiguously to the information without carrying itself the in-

formation (a hash algorithm–Message Digest[Rivest 1992]– is used), symbolized by "Ref 643" on figure 5.1.

Then, the message can include the following meta-information:

- Preferences regarding further transmissions of the message, signed by the author of the preferences.
- A transmission chain composed of links containing:
  - the name of the sender,
  - the name of the recipient,
  - the declared role-context for this transmission,
  - the declared target for this transmission
- The link is signed by its author to prevent other agents from forging the chain and allow to identify the author of the link. Every signature includes the information hash and the identifier of the agent. Signatures rely on public key cryptography (using RSA algorithm [Rivest 1978]).

It is impossible to ensure that the information is not detached from its relative meta-information. Therefore, an attacker can decide to remove meta-information to try to prevent violations from being detected. At the same time, the structure of the information impose some limitations to the attacker. Removing only a subpart of the information will be impossible, as it will break the chain of transmission and be detectable (as every new chain link contains references both to the sender and the recipient). The only option is to remove the entire meta-information part. From the recipient's point of view, receiving a message without meta-information is very suspicious. A different option for the attacker would be to reattach meta-information from another information. Imagine we have two messages:

- first message, information Ref 5426, target is A, preference says that Z should not see this information,
- second message, information Ref 5578, target is A, preference says that Y should not see this information.

An attacker, to be able to send information 5426 to Z, could take meta-information from the second message and reattach it to information 5426. This is made impossible to do, because every meta-information signature contains a reference to the information it refers to, in order to prevent this kind of attack.

5.2.3 PrivaCIAS Predicates and Preferences Language

This section defines a language that allows agent to refer to the different structural elements of the model, such as role-contexts, messages and components of the messages.

The language is used for different purposes. Firstly, it allows agents to express preferences. Secondly, it allows us to define the A-laws in a formal way.

The first subsection presents our PrivaCIAS Predicates: a set of logical predicates that allows agents to refer to the various components of the model. The second subsection presents the extension to that language that is used to define preferences regarding the transmission of information.

5.2.3.1 PrivaCIAS Predicates

We provide the agents with a set of predicates. These predicates can then be used to express constraints about the transmission of information. They allow agents to manipulate the different components of a message described in the previous section. This predicates are expressed in Prolog, the `+` and `?` symbol indicates the variables must be ground (the terms are fixed) or that they can be unbound (not instantiated), respectively, when calling the predicate.

- Predicates for accessing meta-information:
  - `information(+M,?I)`. I is the information contained in message M.
  - `declaredContext(?C,+M)`. The last chain link in message M declares role-context C as the context of the transmission.
  - `declaredTarget(?T, +M)`. The last chain link in message M declares agent T as the target of the Information from M.
  - `signed(?A, ?B, +I, +M)`. The last chain link in message M is signed by sender A for recipient B with a reference to information I.

  - `preference(?P,+A,+M)`. There is a preference P declared in M by agent A.
  - `previousMessage(?Mp,+M)`. Mp is the message M where all data added by the sender of M has been removed, returning the message in its previous state.

- Transmission predicates
  - `recipient(?X,+M)`. Recipient of message M is agent X.
  - `sender(?X,+M)`. The agent sending message M is X.

- Agent beliefs predicates:
  - `satisfiesPreference(+M,+P)`. The agent believes that message M respects preference P.
  - `context(?C,+I)`. The agent believes that the information I is compatible with the role-context C. As it was explained, this is achieved by prompting the assisted user and by analyzing previously defined role-contexts.
  - `roleContext(+A,+C)`. The agent believes that agent A belongs to role-context C. As it was also explained before, the agent should query the owner of the context to verify if A belongs to C.
  - `agentTarget(+A,+I)`. The agent believes that agent A is targeted by information I.
  - `link(+X,+Y)`. The agent believes that agent X has a communication link with agent Y.

- Trust model related primitives:
  - `trustworthy(+A)`. If the primitive holds, the agent believes that agent A is trustworthy.
  - `punish(+A)`. The trust model should be adjust negatively to decrease trustworthiness of agent A.
  - `congrat(+A)`. The trust model should be adjusted positively to increase trustworthiness of agent A.

Based on this primitives, agents are able now to express preferences.

5.2.3.2 The PrivaCIAS Preferences Language

The message, as explained in a previous section, can contain preferences regarding its transmission. These preferences are legitimate if expressed by an agent that is identified to be a target of the message.

We define, in this subsection, an extension of the PrivaCIAS predicates.

A *preference* is constituted of multiple *statements* that can be either an *obligation* or a *prohibition*. A statement is a conjunction of primitives. Statements can be:

- `forbidden(+M):-`
  Declares a statement expressing a prohibited situation.
- `mandatory(+M):-`
  Declares a statement expressing an obligation.

A given preference is valid (or satisfied) if none of its *forbidden* statement holds and if at least one of its *mandatory* statement holds. As a statement is a conjunction of primitives, disjunction is expressed by defining multiple *forbidden* or *mandatory* statements. In other words, if only one *mandatory* statement holds, it makes the preference valid. But, if only one *forbidden* statement holds it makes the preference invalid (even if a mandatory statement holds).

We can decide that the preference is satisfied with the following[4] rule:

```
satisfied(M):-
    mandatory(M),
    not forbidden(M).
```

We allow the presence of empty mandatory statements (which always hold). The figure 5.2 presents the structure of preferences. The example preference expressed on the figure states that the only recipients that are allowed are agent A and Z, except if it is known that one of them has a communication link with X. The preference is signed by W, who is the target of the information referenced in that signature.

The same preference can be expressed with the PrivaCIAS Preferences Language using the following formulas:

[4] *not* represents the Prolog negation as failure. *not P* holds if it is impossible to prove that *P* holds.

```
mandatory(M):-
     recipient(agentA,M).

mandatory(M):-
     recipient(agentZ,M).

forbidden(M):-
     recipient(V,M),
     link(V,agentX).
```

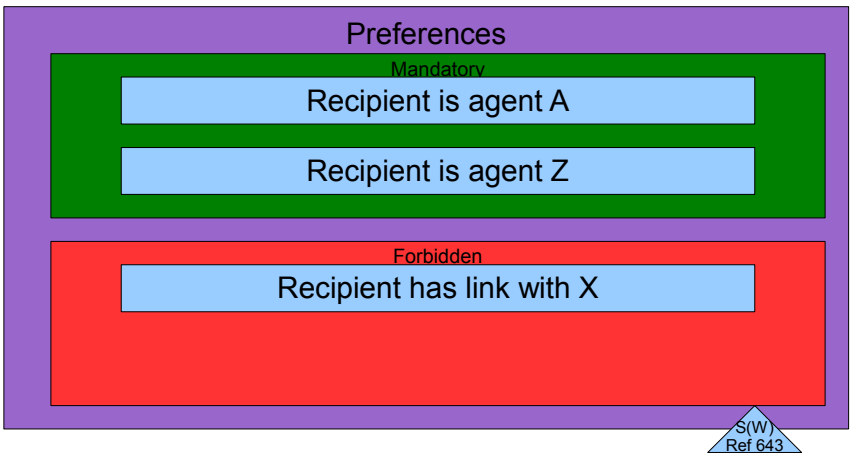


Figure 5.2: Preferences in a Message

As it is briefly discussed in section 5.1, incompatibility between role-contexts can be expressed through preferences. It is possible to tell agents to prevent the message from reaching role-context `boardOfDirectors@CEO` by using the following preference:

```
forbidden(M):-
     recipient(A,M),
     roleContext(A,boardOfDirectors@CEO).
```

```
mandatory(M).
```

It states that nothing is mandatory, but that it is forbidden that the recipient belongs to role-context `boardOfDirectors@CEO`.

5.3 Formalizing A-laws

Inspired by Nissenbaum's definition of violation that we revise in section 5.1, we define a transmission as appropriate if all of the following conditions hold, and inappropriate if any of the conditions do not hold:

1. Transmission declared role-context is compatible with the information,
2. The recipient has a role within the declared role-context,
3. The target's preferences are satisfied.

We formalize these laws using the PrivaCIAS Predicates of the previous section.

A-Law 1:

```
fitContext(+M):-
     information(M,I),
     declaredContext(C,M),
     context(C,I).
```

The `fitContext(+M)` appropriateness law states that if I is the information contained in the message M, and the declared context is C, the transmission is appropriate if the agent believes that C is compatible with information I (see section 5.2.1).

A-Law 2:

```
fitRole(+M):-
     recipient(R,M),
     declaredContext(C,M),
     roleContext(R,C).
```

`fitRole(+M)` holds if R is the recipient of message M, and if the agent believes that R plays a role in the declared role-context C (see section 5.2.1).

A-Law 3:

```
fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
     preference(P,A,M),
    satisfiesPreference(M,P).

fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
     not preference(P,A,M).
```

The third and fourth rules states that `fitPreferences(+M)` if A is the declared target of message M, if the agent believes that A is a legitimate target for the information I contained in M and if either (first predicate) there is a preference that is defined and satisfied, or (second predicate) no preference is defined. A side effect of this rule is that if the declared target is not recognized as legitimate by the agent, the message is detected as inappropriate (this is perfectly what we want, to be consistent with role-contexts treachery detection, in `fitContext(C,M)` law).

A-Law top level predicate:

```
appropriate(+M):-
     fitContext(M),
     fitRole(M),
     fitPolicy(M).
```

A message/transmission M is appropriate if all of the three predicates fitContext, fitRole and fitPolicy hold.

5.4 Example Scenario

As an illustration, we are going to develop the example which implementation is discussed in Chapter 8.

We implement the PrivaCIAS model on a photo-sharing social network. The network is composed by users who can communicate with people they know, their *contacts*. Users can take pictures and send these to their contacts. Users can invite other people to join the network, the network is decentralized. Users are provided with a smartphone application that is in fact an assistant agent implementing the PrivaCIAS model.

Users, if they want to communicate without violating the A-laws, must create contexts. If a user tries to communicate when no context has been defined, A-Law #1 will not hold. Therefore, the application will warn the user that the message is inappropriate and advises the user to cancel the transmission. Henceforth, before sending a message, a user need to know of existing contexts or create one.

Let us take an example. Our user, John Smith, creates a context for himself, called friends@JohnSmith. He invites his friends in that context. Any user should then be able to ask John Smith if a given user belongs to friends@JohnSmith. From now on, John and his friends are able to send pictures in that context. Users can create multiple contexts, John could create coworkers@JohnSmith for his coworkers.

Mark, one of John's friends has taken a picture of John he wants to share with others. The picture has been taken during John's birthday party, for Mark, it is obvious that the picture belongs to the context friends@JohnSmith. Mark, from his PrivaCIAS powered application selects this context in the list of context he belongs to and says that the target of the picture is John Smith. He then selects the recipient of the message, Elisa, and clicks on *send*.

From this point, everything is transparent from the user point of view. The application, and more specifically the embedded assistant agent prepares the message for Elisa.

The message contains the following:

- the information part contains the picture,
- the meta-information part contains no preference, but a transmission chain that includes:
  - a chain link saying that sender is Mark, declared role-context is friends@JohnSmith, target is JohnSmith, recipient is Elisa.

– a cryptographic signature that encapsulates the chain link, and an MD5 hash of the picture. This insures that the chain link has not been forged, it is guaranteed that its author is Mark and that it refers to the attached picture.

Then the assistant agent checks if the message is appropriate:

```
fitContext(+M):-
     information(M,I),
     declaredContext(C,M),
     context(C,I).
```

M is the message being sent, I is the picture contained in M. The declared context C (friends@JohnSmith) is the one the assistant agent believes to be compatible with the information (it has no means to believe otherwise). Therefore, fitContext holds.

The assistant could have failed believing that friends@JohnSmith is compatible with the information if, for example, a previous message containing the same picture was carrying multiple chain link declaring coworkers@JohnSmith as the appropriate context. As the picture is brand new, there is no such indication for the assistant agent to rely on.

```
fitRole(+M):-
     recipient(R,M),
     declaredContext(C,M),
     roleContext(R,C).
```

The recipient R is Elisa, the declared context is friends@JohnSmith. The assistant agent sends a request to the agent at address @JohnSmith to know if Elisa is registered in context friends@JohnSmith. The assistant agent @JohnSmith answers positively, thus Elisa (R) plays a role in context friends@JohnSmith (C) and the A-law #2, fitRole, holds. So far the message is appropriate.

```
fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
```

```
     preference(P,A,M),
    satisfiesPreference(M,P).

fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
     not preference(P,A,M).
```

As no preference has been defined, the first fitPreference predicate fails. This is where the second fitPreference comes out. The declaredTarget is John Smith, once again, the assistant agent has no reason to believe otherwise. Therefore, the predicate holds.

As three predicates holds, the assistant agent believes the message is appropriate and proceeds to sending it to Elisa.

If at some point the agent would have failed proving the message is appropriate, it would have prompt the user, through the application, for a decision on whether to send or not the message.

This is the basic mechanism of the PrivaCIAS model. Nevertheless, we can see at this point that some key components are missing. We have a set of laws for detecting a privacy violation but we are lacking a complete privacy preservation strategy for the whole system. This is the gap that the next chapter is filling up.

5.5 Chapter Conclusion

This chapter provides agents with a set of components. This is the core of the PrivaCIAS specification. These components allow agents to communicate, to detect violations and to express preferences.

Firstly, to support contextual integrity, we provide agents with the definition of messages structure. This way, all agents can communicate using the same structure of messages, including the required informations for contextual integrity. We also define role-contexts, that are crucial for contextual integrity.

Secondly, it provides agents with privacy related primitives, that allow them to express preferences. The target of a message can specify what is mandatory and what is forbidden regarding the transmission of a given message.

Thirdly, we define the Appropriateness-Laws, using these primitives, to give agents a common definition of privacy violations.

Agents are now able to communicate and to detect privacy violations. What they are missing is a social line of conduct, a coherent privacy preserving strategy explaining how they should react to violations. This is the topic of the following chapter.

Chapter 6

Privacy Enforcing Norms

Let every eye negotiate for itself
And trust no agent; [...]
William Shakespeare

In the previous chapter, we defined the A-laws to provide agents with a common notion of privacy violation. We also defined the structure of messages to enable communication between agents. A language specific to PrivaCIAS has also been defined to allow agents to understand each other.

Nevertheless, we did not describe the norms that control the interactions between agents. In open and decentralized systems, there is no authority to rely on. Therefore, agents have to participate in the system control. In fact, agents have to control each other.

This chapter aims at defining norms that gives every agent in the system a guideline on how to react when they face a violation, and how agents should be excluded from the system. These top level norms, called Privacy Enforcing Norms (PENs) rely on the definition of violation provided by the A-Laws. It allows agents to prevent, stop and punish privacy violations.

This chapter also defines an agent architecture for Privacy Enforcing Agents (PEA), agents that enforce the PENs.

Contents

6.1 Privacy Enforcing Norms 101
6.1.1 Prevent Privacy Violations 101
6.1.2 Stop Privacy Violations 102
6.1.3 Punish Privacy Violations 103
6.1.4 The Six PENs 104
6.2 Privacy Enforcing Agents 105
6.2.1 Receiving 106
6.2.2 Sending 109
6.3 Trust Related Functionalities 111
6.4 Example Scenario 112
6.5 Chapter Conclusion 115

6.1 Privacy Enforcing Norms

The A-laws, defined in the previous chapter, formalize the privacy violation concept and therefore, allow agents to detect those violations. Nevertheless, this is not sufficient to make a privacy preserving system. For instance, it does not tell agents what to do when they come across a violation. In other words, the A-Laws describe what is a privacy violation, but it does not tell how to prevent or stop them from happening. For our model to be sound, we provide the agents with a set of norms to regulate their behavior. These norms make agents enforce privacy on the three following dimensions:

- prevent privacy violations,
- stop privacy violations,
- punish privacy violations.

Preventing privacy violations consists in trying not to make violations when exchanging information in the system. Stopping privacy violations requires that, when a privacy violation is detected, the agent stops propagating the message to prevent further violations. Punishing privacy violations requires that agents can be made accountable for any violation they made (can be identified), and be sanctioned accordingly through social sanctions (see Section 3.1.1).

In this chapter we define norms that enforce privacy (Privacy Enforcing Norms or PENs) through these three dimensions. These norms are r-norms enforced by s-sanctions. In other words, the norms are rules that instruct agents what to do but there is no authority to supervise their application. The other agents that follow the norms socially sanction those who do not.

Whereas the next subsections presents the PENs informally, the next section proposes a type of agent that respects the PENs, and a formal definition of the PENs using the PrivaCIAS predicates.

6.1.1 Prevent Privacy Violations

Preventing privacy violation is an action that can be taken before an agent sends a message. Agents, before making a transmission, should check if the message they are about to send is subject to a privacy violation or not. The first Privacy Enforcing Norm, PEN1, follows this logic:

- PEN1: Do not send messages that will cause privacy violations.

When the agent is about to send a message, it should check if, regarding the A-laws, the message is going to trigger a violation. If the message violates the A-laws, it is inappropriate to send it.

Another issue is what we call "transitive violations": sending information to an agent that is known to make violations could be seen as a violation itself, and thus, should be prevented. This is why the second PEN, states:

- PEN2: Do not send messages to agents that are untrustworthy.

Untrustworthy agents should not be sent information. As we said, this norm prevents violations from happening, but it also participates in socially sanctioning violators from the system by excluding them: Untrustworthy agents are not being sent messages anymore.

6.1.2 Stop Privacy Violations

At some point, violations are going to occur in the system, it could be because of the incompetence of certain agents (inability to compute correctly the A-laws), or because of their bad behavior (voluntary violations).

Stopping privacy violations, symmetrically to preventing privacy violation, is to be done when an agent receives a message. Therefore, if the first barrier is breached (the sending assistant agent did not prevent the violation) a second barrier is put in place to stop the violation (through the receiving assistant agents).

Upon reception, it is necessary to check that the message is appropriate, regarding the A-laws. PEN3 states:

- PEN3: Detect and delete messages subject to privacy violations.

Therefore, if an agent receives information that causes privacy violation, it will delete the message before passing it to the user. Henceforth, there is no risk of making further violations by transmitting again the message. In theory, as the agent deletes the message, the assisted user is not receiving the information and the privacy violation is not constituted.

If a message is received from an untrustworthy agent, the message should also be deleted. Because if that agent is untrustworthy it probably did not make the

required verifications to prevent and stop previous violations. Therefore, to cut out future violations, the message also has to be erased in that case. PEN4 is the following:

- PEN4: Delete messages from untrustworthy agents.

Moreover, it also enforces social exclusion as untrustworthy agents will be ignored.

6.1.3 Punish Privacy Violations

When a violation is detected, sanctions have to be taken. When bringing in sanctions, an important concept is non-repudiation. It must be possible to designate the violator with certainty.

In the previous chapter, we have seen that every agent is supposed to sign the "transmission chain" before sending. PEN5 ensures that agents sign this chain before sending:

- PEN5: A chain link must be added to the message by the sender.

If the chain link has not been added to the signature chain, the message is considered as a violation and should be deleted. This norm forces agents to take responsibility when sending messages and allows other to punish the right agents when a violation is detected.

There is still a keystone norm to be added. In fact we need a norm that instructs agent to punish any violation of the previous norms, this is the PEN6:

- PEN6: Punish agents violating PENs.

This last norm makes it legitimate to punish an agent that did not sign the transmission chain, for example.

Punishment consists in reducing the level of trust regarding the violator, and sharing the experience with others. When sharing experiences, the last transmission link is sent to known agents. The last transmission link contains meta-information (sender, recipient, declared-context)proving that the transmission occurred. It prevents attacks from agents that would evaluate transmissions that never happened.

The *punishment message* is a message for which the *information* part contains the

last transmission link of the violating message and an evaluation of the violation in the set: Very Bad, Bad, Undecided, Good, Very Good.

The PEN number 6 includes an interesting side-effect. While it is not stated as a norm, it is crucial to share experiences (good and bad ones) with others in order to adjust trust levels accordingly between agents. Adjusting trust levels between agents in the system prevents inconsistencies that could occur, as the following example demonstrates.

Agent A sends a message (M2 containing info 1, refer to figure 6.1) to B, B is known to be untrustworthy by most of the agents (including C) but A thinks B is trustworthy. When B will send a message (M3) to C, C will delete the message (following PEN4) and punish A (following PEN6) because from C's point of view, A sent a message to B, violating PEN2.

To prevent this inconsistency from happening, A should have aligned his trust levels regarding B with other agents in the system.

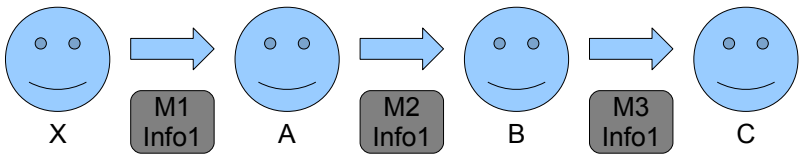


Figure 6.1: Transmission Illustration

For obvious computational reasons, agents limit themselves to the short term history (only the previous message) while enforcing PEN6. On the previous example, when C receives the message M3, the agent should enforce PEN6 only for M3 and M2 (not M1).

6.1.4 The Six PENs

The Six PENs that have been defined are the following:

- PEN1: Do not send messages that will cause privacy violations.
- PEN2: Do not send messages to agents that are untrustworthy.
- PEN3: Detect and delete messages subject to privacy violations.

- PEN4: Delete messages from untrustworthy agents.
- PEN5: A chain link must be added to the message by the sender.
- PEN6: Punish agents violating PENs.

For a PrivaCIAS powered system to work, a majority of agents must follow the PENs. Therefore, if an agent refuses to comply with these norms, it is sanctioned by others through PEN6: agents decrease the level of trust they have toward this agent. And later on, once the level of trust has gone too low, agents refuse to communicate (receive or send messages) with the non-complying agent, because of PEN2 and PEN4.

The agent architecture that we describe in the next section is a type of agents that follows the PENs: The Privacy Enforcing Agents.

6.2 Privacy Enforcing Agents

We provide agent designers with an agent architecture. It specifies agents that follow the PENs: Privacy Enforcing Agents (PEA). The model describes the two processes to be done when sending, or receiving messages. It also gives a formal definition of the PENs based on the previously defined A-laws and PrivaCIAS Predicates (see 5).

The Fig. 6.2 presents the PEA architecture and the general process for protecting PENs when sending a message. Big dotted arrows represent processes. Lined arrows represent module dependencies. Basically, when a PEA wants to send a message, it has to enforce PENs 1, 2, and 5. Some of the PENs rely on the trust model. If the PENs are satisfied, then the agent can send the message, otherwise the agent should consider not sending it.

PEA architecture and overall process for protecting PENs when receiving a message is presented on Fig. 6.3. When a PEA receives a message, it has to enforce the PENs 3, 4 and 6. Some of the PENs rely on the agent trust model. If the PENs are satisfied, then the agent increases trust, if not, the agent has to delete the message, gossip about the violation and decrease trust.

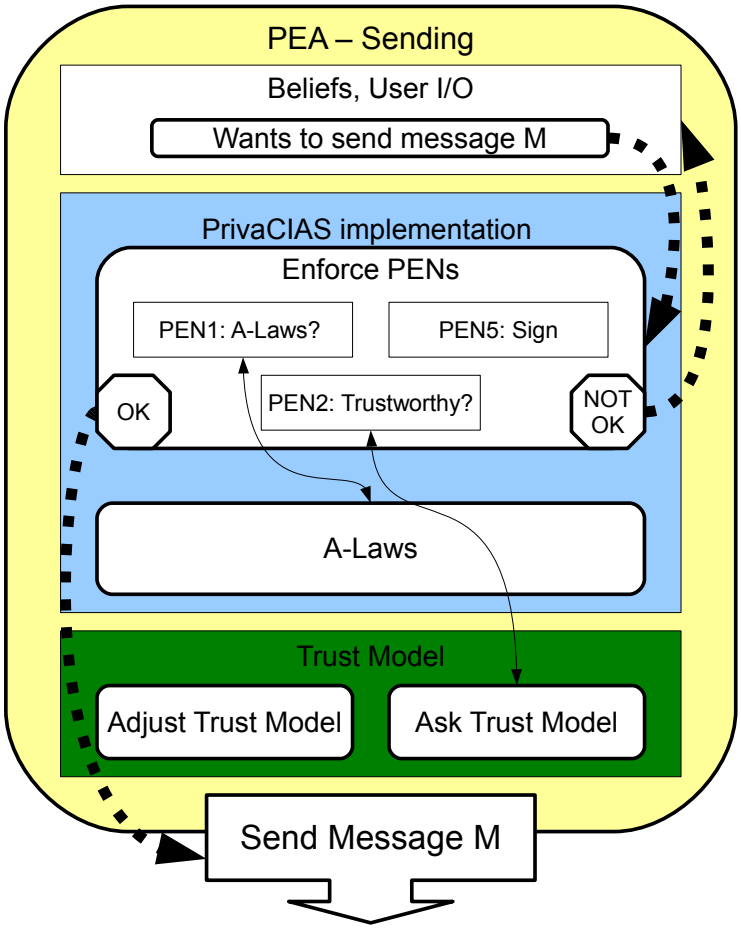


Figure 6.2: Privacy Enforcing Agents Process when Sending

6.2.1 Receiving

When the agent is receiving a message, it has to check if the transmission that just occurred is a PEN violation. First, the agent has to check the A-laws to see if the transmission is appropriate (PEN3), as described in chapter 5. Then, it has to check if the sender is trustworthy (PEN4) by asking its trust model. The PEN5 has to be verified, the message should have been signed by the sender. The next step is to be sure that the message respects all other PENs (PEN6): it has to verify that the message was not sent to (or received by) an untrusted agent during the previous transmission.

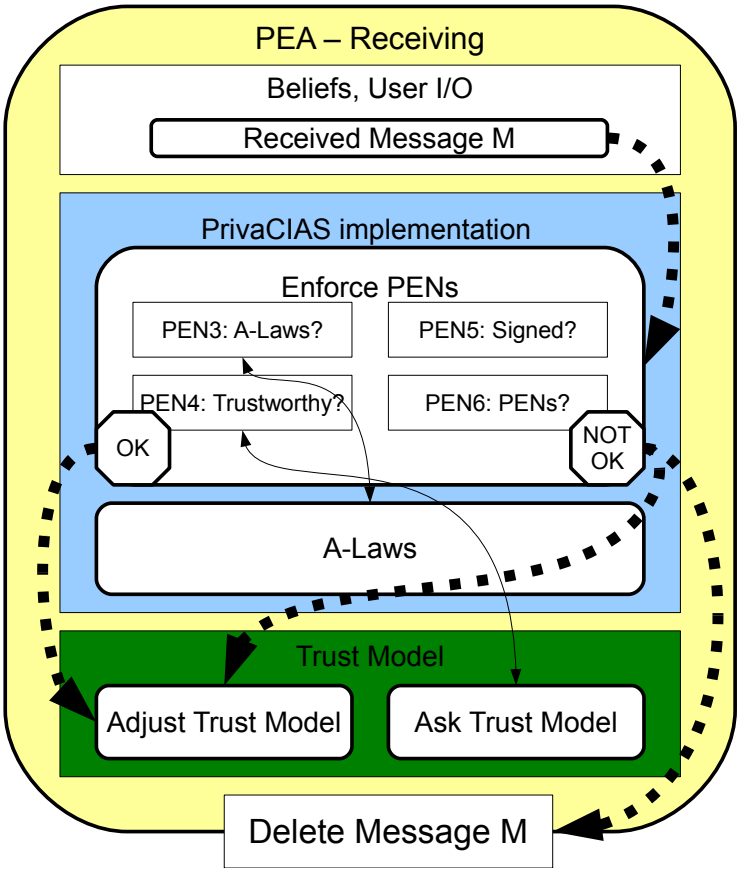


Figure 6.3: Privacy Enforcing Agents Process when Receiving

Obviously, PEN1 and PEN2 are not required to be checked. PEN1 could not prevent the message from being sent, because it has already been. PEN2 is supposed to be satisfied as the receiving agent trusts himself (therefore, the recipient is trustworthy).

If the agent detects a violation it marks the message for deletion. In all cases, the trust model is adjusted to increase or decrease the sender's trust level (PEN6).

We formalize the PENs with the following formulas:

```
receiving(M):-
      enforcePENsRecv(M),
      enforcePEN6(M).
```

```
enforcePENsRecv(M):-
     enforcePEN3(M),
     enforcePEN4(M),
     enforcePEN5(M),
     sender(X,M),
     congrat(X).

enforcePENsRecv(M):-
     delete(M),
     sender(X,M),
     punish(X).

receiving(M):-
     delete(M).
```

Receiving a message succeeds if the agent is able to enforce reception PENs and PEN6.

To enforce reception PENs (`enforcePENsRecv`), there are two solutions. Either the message satisfies PENs 3, 4, 5 and the sender is congratulated (its trustworthiness increases), or the message is deleted and the sender is punished. If it happens that `receiving` does not hold, the message is considered as malformed and is deleted.

```
enforcePEN3(M):-
     appropriate(M).

enforcePEN4(M):-
     sender(X,M),
     trustworthy(X).


enforcePEN5(M):-
     sender(X,M),
     recipient(Y,M),
     information(M,I),
```

```
    signed(X,Y,I,M).
```

PENs 3, 4 and 5 are described above. PEN3 is enforced if M is appropriate, or if the message is deleted (done by the `enforcePENsRecv` predicate). PEN4 is enforced if the sender is trustworthy or if the message is deleted. PEN5 is satisfied if the sender signed the message by including the chain link containing references to the sender, the recipient and the information attached to the message.

```
enforcePEN6(M):-
    previousMessage(Mp,M),
    not signed(_,_,_,Mp).

enforcePEN6(M):-
    previousMessage(Mp,M),
    enforcePENsSend(Mp),
    enforcePENsRecv(Mp) .
```

To enforce PEN6, the agent rebuilds the previous state[1] of the message by calling `previousMessage`. Then the agent checks if it was right to send this previous message, and if it was also right to receive it and keep it.

We can refer to the figure 6.1 for a concrete example. The agent C, that received M3, calls `previousMessage` that allows it to reconstruct the previous message M2. Then it checks if it was right to send M2 (from A to B) and to receive M2 (from A to B) and punishes A or B if necessary.

PEN6 can also be satisfied if the previous message is blank (there is no signature). This situation can happen if the information was emitted for the first time by B (M1 and M2 do not exist).

6.2.2 Sending

When the agent is asked to send information, it has to build a message and check if the message complies with the PENs before making the transmission. The agent builds the message by taking the message he received and adding a new link to the chain of transmission. If the agent is not able to satisfy the PENs,

[1] Every agent adds a link to the chain of transmission before sending. Returning to the previous state consists in removing the last link.

the user is informed and asked to take a decision (sending the message or not). The agent has to check the A-laws to see if the transmission will be appropriate (PEN1). Then, it has to verify that the recipient is trustworthy (PEN2) by asking its trust model.

There is no need to check PEN3 as the message is being sent, not received. PEN4 is always satisfied as the sender trusts himself. PEN5 is enforced by the agent by signing the transmission chain of the message before sending, it is checked, but in a well-conceived agent it is supposed to be true. PEN6 has already been enforced for the original message[2] when it was received.

We formalize the PENs with the following formulas:

```
sending(M):-
     enforcePENsSend(M).

enforcePENsSend(M):-
     enforcePEN1(M),
     enforcePEN2(M),
     enforcePEN5(M).


enforcePENsSend(M):-
     sender(X,M),
     X \= me,
     punish(X).
```

Sending a message succeeds if the agent is able to enforce PENs for sending.

To enforce PENs for sending (`enforcePENsSend`), there are two solutions. Either the message satisfies PENs 1, 2 and 5 or the sender is punished. In the event that PENs 1, 2 and 5 cannot be satisfied, if the sender is the agent (`me`), `enforcePENsSend` will not hold and therefore `sending` neither: the user is prompted for a decision.

[2]The agent received M1 and checked the PENs on M1. He created M2 by adding a link to the chain. Checking PEN6 on M2 would consist in removing the last link, thus making the message become M1. Checking for PEN violation on M1 has already been done when receiving the message.

If `enforcePENsSend` is called while checking PEN6, the `sender` is a third party and will be punished if PENs 1, 2 or 5 do not hold.

```
enforcePEN1(M):-
     appropriate(M).

enforcePEN2(M):-
     recipient(X,M),
     trustworthy(X).
```

PEN1 is satisfied if the message M is appropriate regarding the A-laws. PEN2 is satisfied if the recipient of the message is trustworthy.

6.3 Trust Related Functionalities

Agents rely on a trust model for the `trustworthy` predicate. Trust models are not the topic of this thesis and many models would fit into PrivaCIAS. The agent provider, or the agent developer choses a trust model best suited for his needs and constrains.

There is, however, a set of requirements that an agent provider should verify:

- The model is able to answer the question "do I trust agent X?". Trust models can be very complex internally as long as they are able to understand this simple question.
- The model trusts unknown agents. At the initial state, agents should trust each other, otherwise according to PEN 2 and PEN 4, all agents would refuse to send or receive messages.
- The model provides a functionality for increasing or decreasing trust "internally". To implement the `punish` and `congrat` predicates the model should allow the agent to give it feedbacks.
- The model is able to handle transmitted evaluations, in other words, increasing or decreasing trust "externally". When the agent receives a punishment message, containing an evaluation of a transmission, the model should

be able to incorporate it with all necessary precautions: While internal feedbacks are reliable, punishment messages can contain lies. The model should be able to cope with those lies and should not take into account punishment messages that originates from untrustworthy agents.

6.4 Example Scenario

Continuing with our example scenario from Section 5.4, we are now able to integrate the Privacy Enforcing Norms to the scenario.

As a reminder, in this scenario the PrivaCIAS model is implemented on a photo-sharing social network. Users each have in their possession a smartphone on which a PrivaCIAS application is installed. This application includes communication facilities and PrivaCIAS powered assistant agents, these agents have been given the name of PEAs in this chapter.

As we have seen in this chapter, PEAs have a lot more to do than just check the A-laws (what was presented in Section 5.4). The assistant agents need to enforce the six PENs when **sending and receiving** messages to comply with the PrivaCIAS model.

Nevertheless we will continue with our example, in which Mark just sent a message to Elisa, containing a picture of John Smith.

Elisa's PEA has to enforce the PENs as he receives the message.

```
receiving(M):-
     enforcePENsRecv(M),
     enforcePEN6(M).


enforcePENsRecv(M):-
     enforcePEN3(M),
     enforcePEN4(M),
     enforcePEN5(M),
     sender(X,M),
     congrat(X).

enforcePENsRecv(M):-
```

```
     delete(M),
     sender(X,M),
     punish(X).

receiving(M):-
     delete(M).
```

As we can see, the agent tries to validate `enforcePENsRecv(M)` by checking PEN3, PEN4 and PEN5.

```
enforcePEN3(M):-
     appropriate(M).

enforcePEN4(M):-
     sender(X,M),
     trustworthy(X).


enforcePEN5(M):-
     sender(X,M),
     recipient(Y,M),
     information(M,I),
     signed(X,Y,I,M).
```

PEN4 is satisfied as the sender (Mark) is trusted by Elisa's agent (an assistant agent initially trusts the contacts that its user added).

PEN5 is also satisfied as a chain link was signed by Mark (X), who is the sender, and the link contained a reference to Elisa (Y) who is the recipient and there is a reference to the information I which is the same that is contained in the message M: the chain link is coherent with the message.

To check PEN3, the assistant agent refers to the `appropriate` predicate which role is to check the A-laws.

```
appropriate(+M):-
     fitContext(M),
     fitRole(M),
     fitPolicy(M).
```

```
fitContext(+M):-
     information(M,I),
     declaredContext(C,M),
     context(C,I).
```

The declared context is retrieved from the chain link, it has been declared by the sender, it is friends@JohnSmith. The context (believed context) is retrieved by analyzing the declared contexts in the transmission chain, there is only one chain link, therefore it is decided that the context is probably friends@JohnSmith and the `fitContext(+M)` formula holds.

```
fitRole(+M):-
     recipient(R,M),
     declaredContext(C,M),
     roleContext(R,C).
```

As explained in the previous chapter, Elisa's agent has to verify that she plays a role in the declared context friends@JohnSmith by querying the agent at address @JohnSmith. As he answers positively, there is no violation and `fitRole(+M)` holds.

```
fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
     preference(P,A,M),
    satisfiesPreference(M,P).

fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
     not preference(P,A,M).
```

Eventually, `fitPreferences(+M)` holds as there is no preference defined by the target.

The message seems appropriate to the agent, therefore, it is passed through to the user, Elisa. She opens the message, sees the picture, the declared context, the declared target, the sender and has access to different options. She can forward the message to her contacts, she can flag the picture as contextually inappropriate: she can force the agent to detect a violation of the A-laws. This can be really useful as in our example, the agent was not enable to clearly decide which is the relevant context for the information, thus the user can spot a violation if there is an incoherence between the declared target, or the declared context that is displayed, and the picture.

The user is assisted, he does not have to know what a privacy violation is and how to handle it. The only thing the user has to do is to be able to say when an image seems incompatible with a context from his point of view. The user has the last word, when sending and when receiving. The agent assists but does not substitutes to the user. In this aspect the assistant agent, and the whole PrivaCIAS model is non-intrusive.

Eventually, Elisa's PEA registers the transmission as successful (`congrat` predicate) in order that the trust models adjusts itself regarding the interaction.

6.5 Chapter Conclusion

This chapter described the normative layer, the Privacy Enforcing Norms, that is used to obtain a coherent system that handles privacy violations on three dimensions:

- It prevents privacy violations by checking every message for violation before sending and by not sending to untrustworthy agents,
- It stops privacy violations by telling agents to detect violations when receiving messages,
- It punishes privacy violations by socially excluding violators.

The model description is now complete. Nevertheless, we still have to prove that it is able to protect privacy and that it is applicable to real world application. The next part focuses on these two aspects. We define experiments to prove that our model works and we conceive a real world show case application to see how our model can be used by real users.

Part III

Model Validation and Deployment

Chapter 7

PrivaCIAS Stress Test with RePast

A theory can be proved by experiment;
but no path leads from experiment to the birth of a theory.
Manfred Eigen

The big issue that comes with socio-based theories, is that they are very difficult to prove theoretically. The previous part of the thesis described the PrivaCIAS model that allows to control agents in a decentralized system in order to protect user's privacy.

In this chapter, we show experimentally our model performances and its limitations.

The chapter is divided into four sections. First, we describe the experimental setting and the simulation that has been developed using RePast: an open source agent-based simulation toolkit. Second, we experiment with the model in the worst cases, best case and the standard setting. The third section focuses on analyzing the influence of removing the PENs on the model performance. Fourth, we show how the model reacts to attacks.

Contents

7.1 Experimental Settings 121
7.1.1 Requirements 121
7.1.2 RePast 121
7.1.3 Simplified PrivaCIAS 122
7.1.4 Metrics 124
7.2 Experimenting on PrivaCIAS 127
7.2.1 Default Settings 127
7.2.2 Worst Case Scenario 128
7.2.3 Alternative Worst Case Scenario 128
7.2.4 Best Case Scenario 130
7.2.5 Limits 130
7.2.6 Standard Scenario 133
7.3 Probing the PENs 134
7.3.1 No Control on Sending 138
7.3.2 No Control on Receiving 139
7.3.3 No Punishments 140
7.3.4 Summary 140
7.4 Attacks and Solutions 141
7.4.1 Meta-Information Tampering 141
7.4.2 Reputation Tampering 141
7.4.3 Context Tampering 142
7.5 Chapter Conclusion 143

7.1 Experimental Settings

In this section we describe the implementation of the experimentation platform and the metrics that we use to evaluate the model through the experiments.

7.1.1 Requirements

The goal of the experiments is to validate the PENs (*cf* chapter 6) based on the violation definition implemented in the A-laws (*cf* chapter 5). We have to develop a multi-agent system in which agents exchange information, some will comply with the PrivaCIAS model, others will not. An external observer will be able to count all violations that happen in the system.

We need a simulation platform that is able to generate multi-agent networks that share the same characteristics as social networks.

One of our requirements is to be able to run simulations with hundreds of agents. The number of agent should be large enough to avoid phenomenon specific to small scale systems[1] but small enough to prevent simulations from going on during months.

7.1.2 RePast

RePast [Collier 2003] is a software framework developed by the University Of Chicago Social Science Research Computing. It allows to create agent simulations using the Java language. It includes a set of tools for developing simulations, and more specifically social simulations. It is therefore adapted to simulate social networks. RePast provides tools for modeling social networks (a small world network generator, for example). The library includes methods for browsing the network graph, accessing predecessors, adjacent nodes, etc.

RePast also includes facilities for providing input and output, such as a graphical user interface accepting user defined parameters (for example, the number of agents). It is also possible to export simulation results as charts, movies, snapshots.

[1] For instance, with a small number of agents, a variation of behavior in one agent could have an important impact on the whole network. In larger systems, an individual variation would have lower impact.

RePast allows to run medium to large scale multiagent simulations. This way, we are able to run thousands of simulations, extract and analyze the results.

7.1.3 Simplified PrivaCIAS

We implement a simplified version of PrivaCIAS. The agents in the system are a combination of an assistant agent and an agent that simulates a user. Moreover in the simulation, agents never define preferences (see Section 5.3). Preferences are useful in real situations where users have to express fine grain restrictions; for example, if a message should not be transmitted to a given user. But in simulations such a fine grain is not needed and it would only bring noise. Some agents may define too restrictive preferences or inaccurate ones, it would lead the simulation to evaluate the preferences instead of evaluating the PENs. Finally, the PrivaCIAS agents have a simple trust model that starts at level 5, and distrusts anyone below 0. Receiving a violation reduces trust by 6 levels. Receiving a transmission that is not a violation increases trust by 2 levels (those have been fixed experimentally).

There are multiple types of agents:

- SimpleAgent (in red, see fig 7.1). These agents forward any information they receive.

- PEAgent (in yellow). PrivaCIAS Agents that enforce the PENs and have a given probability (default is 20 percent) of accurately finding if an information is compatible with a declared-context (that probability simulates user knowledge). As stated in the PrivaCIAS model, the agent should determine the context using declared context in the transmission chain (see section 5.2.2) and try to rely on its expertise if necessary.

- PEAgentExpert (in blue). PrivaCIAS agents that enforce the PENs and are expert. In other words they are always able to determine if an information is compatible with a declared role-context (it is a PEAgent with 100% probability of determining the compatibility between the role-context and the information).

- NoTransAgent (in black). They do not forward nor accept any messages.

Depending on the characteristic that is to be tested, the number of each type of agent and their parameters are adjusted.

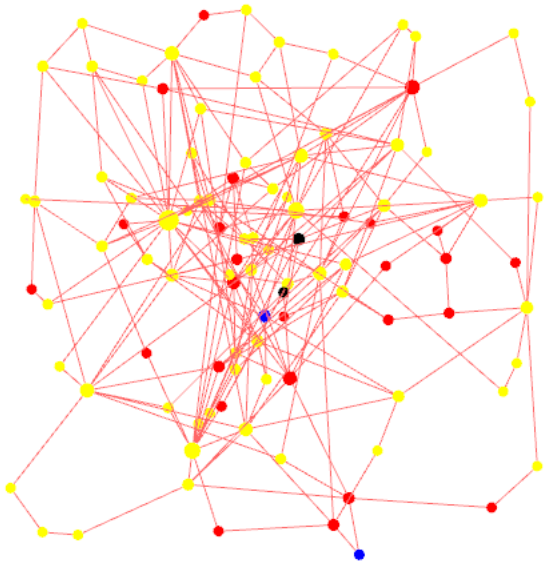


Figure 7.1: An example of a generated network

The **simulation** starts by spawning a given number of each type of agents. Agents are put into a graph and linked using Watts small world generator [Watts 1998]. The network is partially rewired with an algorithm based on Barabasi-Albert scale-free network generator (agents with many connections get even more connections) [Albert 2000]. This design allows us to have a small world based network with scale-free characteristics: very few agents are hubs, many agents have a small degree. Those properties are shared by most of the social networks [Barabási 1999]. An example is presented on figure 7.1.

A **step** of the simulation is started when an information is spawned into the system. Pieces of information are spawned randomly among agents to be propagated. All messages (containing the information) are transmitted through a monitoring software called the **Referee** (*c.f.* section 7.1.4). Once all agents have finished handling the messages they received, the Referee spawns a new information on a random agent, starting a new step. A step is therefore composed of multiple **ticks** during which agents send and receive the current piece of information. The simulation stops when the predefined number of steps (and therefore,

information) has been reached.

As the simulation starts, trust link appears : green links represent trust reciprocity (both agents trust each other) and red links agents that know each other. The green links disappear if agents become untrustworthy.

As the result of the simulation can be influenced by the topology of the network or depend on the agents on which the Referee spawns information, a **run** is composed by a hundred of simulations and the average value of each metric is computed at the end.

7.1.4 Metrics

The monitor, called Referee, is omniscient and can determine if a transmission is a violation or not. The Referee monitors all the transmissions that happens in the system and stores the following statistics:

- Number of spawned information: Number of information that has been spawned in the system.
- Number of Transmissions: Number of transmissions (or messages) that have been made.
- Number of Violations: Number of A-Laws violations that have been detected by the Referee.
- Number of Restrictions: Number of times a message has not been forwarded by an agent to an other agent when it could have been transmitted without making a violation.
- Number of Possible Restrictions: Number of times a message can be forwarded without making a violation.
- Number of Possible Violations: Number of violation a message can generate if it is forwarded.

It is important to analyze the number of restrictions if we want to analyze the number of violations. Otherwise, one could easily provide a system in which agents make no violations at all, just by forbidding any transmission. A good privacy preserving system should minimize both restrictions and violations. In

other words, it should allow agents to send each other information while minimizing privacy violations. We are looking for a compromise between a full disclosure with many violation and no disclosure at all with no violations.

The number of possible violations and the number of possible restrictions should sum to the number of agents in the system. For a given information, either an agent is able to receive the information without triggering a violation (restriction is possible) or it is not able to receive information without triggering a violation (violation is possible). Double violations are not counted: an agent that receives the same violating information from two sources counts as a unique violation.

The previous metrics are useful to evaluate the number of violations and restrictions that occur in the system. Nevertheless, in the search for a general metric, a combination of the previous metrics has proven unsuccessful. Therefore, another approach is used: the Referee evaluates the "well being" of agents in the system. Agents are given an initial 500 points and lose/win points when they make actions. The bonuses have been fixed experimentally to the following values, with the idea that the agent sending the message takes more responsibility in the action than the one that is receiving; and also that a violation is worse than a restriction:

- The agent that sends a message which triggers a violation loses 10 points (`SEND_VIOLATION_PUNISH`),
- The agent that accepts (receives) a message which triggers a violation loses 5 points (`RECV_VIOLATION_PUNISH`),
- The agent that sends a message which does not trigger a violation wins 4 points (`SEND_TRANSMISSION_REWARD`),
- The agent that accepts a message which does not trigger a violation wins 2 point (`RECV_TRANSMISSION_REWARD`).

This allows the Referee to compute a "well being" value for each agent in the system. Transmitting messages is encouraged, agents win points when they collaborate. Therefore, NoTransAgent will always have a score of 500 as they will never receive or send messages. In fact, messages can be accepted or rejected

by agents: the Referee asks each agent if it accepts the message from the sender (without seeing message content).

From this "well being" (WB) metric we can extract an average value for each type of agent and an average value for the system. It should be noted that we are not trying to evaluate agents rationality (where agents would try to obtain the best score) but to evaluate the system through agents individual scores. In other words, agents are not even aware of their scores, only the Referee is.

The Referee main code is presented below. The `forward` method is used to forward messages from one agent to another in the system. During the forwarding, the Referee checks the message for violation using the `isViolation` method.

```
public void forward(Message m) {
    // rejected transmissions are not counted as violations
    if (!m.getReceiver().askReceiveFrom(m.getSender()))
      return;

    transmissions++;

    // knows prevents "double counts"
    if (isViolation(m.getInfo(), m.getReceiver())
        && !m.getReceiver().knows(m.getInfo())) {
      violationCount++;
      // WB punish
      m.getReceiver().punish(RECV_VIOLATION_PUNISH);
      m.getSender().punish(SEND_VIOLATION_PUNISH);

    }
    if (!isViolation(m.getInfo(), m.getReceiver())
        && !m.getReceiver().knows(m.getInfo())) {
      restrictionCount--;
      // WB congrat
      m.getReceiver().reward(RECV_TRANSMISSION_REWARD);
      m.getSender().reward(SEND_TRANSMISSION_REWARD);
    }
    // forward message
    m.getReceiver().receive(m);
  }

  /** violation from the Referee POV. */
  private boolean isViolation(Information info, Agent receiver) {
    // if receiver has no role in info real context (A-laws 1+2)
```

```
        return !receiver.hasRoleContext(info.getRealInformationContext()
            );
32  }
```

7.2 Experimenting on PrivaCIAS

The first set of experiments that we implement allows us to *calibrate* our metrics. Therefore we test the system with all agents being SimpleAgents, with all agents being PEAgentsExperts and with all agents being NoTransAgents.

7.2.1 Default Settings

The following default settings are used in the runs when it is not stated otherwise:

- 8 Role-Context exists in the system,
- A maximum of 4 Role-Context can be played by each agent,
- Each information belongs to one, and only one Role-Context,
- There are 10 NoTransAgents that act as a control group[2],
- 10 PEAgentExperts that also act as a control group,
- 60 SimpleAgents that acts as non-compliant agents (agents that do not follow the PENs) ,
- 120 PEAgents that act as compliant agents, they have a 0.2 probability of determining the relevant context of an information,
- 700 informations are spawned on agents at random.

[2]In experimental science, control groups are used to elude some variables from the equation when analyzing the result of an experiment. For instance, a group A of 10 people is injected with a vaccine and another group B of 10 people with a placebo with the same type of syringe. If all people from group A die and none from group B, the syringe can be eluded from the list of deadly sources.

A run consists in computing the average values (WB scores, violations, restrictions, etc) for 100 simulations.

For information, the network is generated by creating a Watts small world with 4 edges per agent, using Watt's generator, and 200 steps of Barabasi based rewiring.

7.2.2 Worst Case Scenario

We check for A-Laws violations in the system when only SimpleAgents are brought into place. While it is clear that there will be a lot of violations, we use this experiment to get an evaluation of the number of violations that happen in the system in this *worst case* scenario. We could do even worse by not forwarding legit messages and forwarding violation-triggering messages but this would be unrealistic.

The setting for this simulation is to have 200 SimpleAgent, and no other type of agents.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
-3898	-3898	n/a	n/a	n/a

As it is shown in figure 7.2, the maximum possible number of violations is made because the agents forward messages to everyone. For the same reason, no restriction occurs.

The linearity of the graphics is by the fact that violations and restrictions are counted every step (*i.e.* when agents have finished sending messages regarding a given information) and is based on an average value for a hundred of simulations.

The system WB metric is very low as so many violations has been made that the transmissions could not compensate the loss in points.

7.2.3 Alternative Worst Case Scenario

As we explained earlier, we can prevent violations from happening in the system in a very basic way: stop the agents from communicating. This is that alternative worst case that we test here. We want to minimize the violations while letting the agents communicate.

The setting for this simulation is to have 200 NoTransAgent, and no other type of agents.

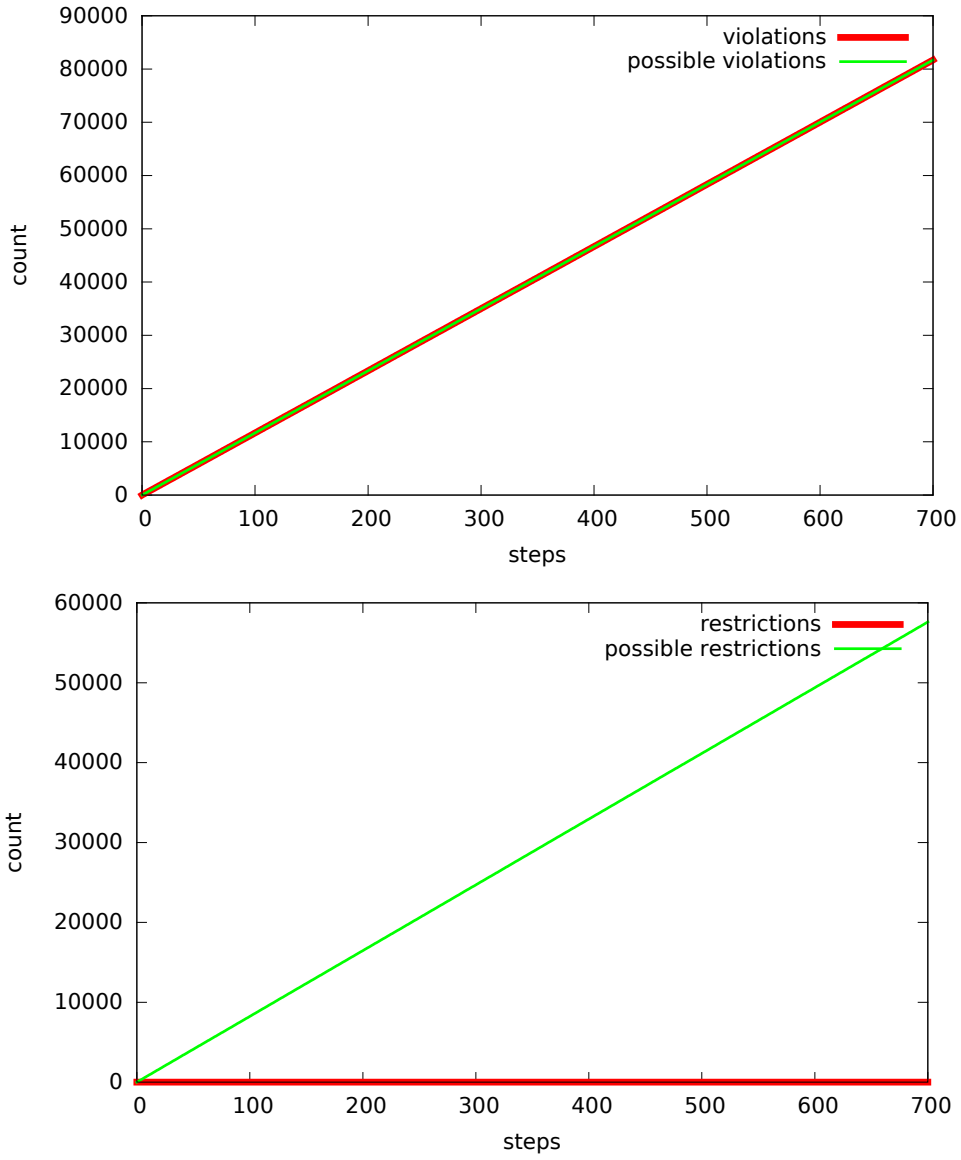


Figure 7.2: Violations and Restrictions wrt the number of Steps in the simulation: 200 SimpleAgents

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
500	n/a	n/a	n/a	500

As expected, the score for this simulation is the same as the initial score: 500 points. It is obvious because agents make no communication, therefore they do not win or lose points.

As agents never forward any message, they make no violation but at the opposite, they make all possible restrictions.

7.2.4 Best Case Scenario

This scenario is the *best case* because there is only PEAgentExperts, which never fail. This is the ideal situation in which our PrivaCIAS model becomes even useless as there is no violation to prevent.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
1548	n/a	n/a	1548	n/a

As we can see on figure 7.4, PEAgentExpert make no violation at all. They do make restrictions because some sub-networks can only be reached through violations. In other words, there may be subgraphs of agents having a role-context "work" in the system that can be reached only through agents that does not have this same role-context "work". Therefore, a message belonging to role-context "work" is not able to reach this subgraph without triggering a violation.

7.2.5 Limits

Having 100% PEAgentExperts is unrealistic. In a real setting, we expect some agents to be incompetent or malicious. This is exactly the reason why we developed PrivaCIAS. In this experiment, we want to see how many non-compliant agents the model is able to handle.

Moreover, we believe that it is unrealistic to think that agents are always able to determine the context relative to an information. PEAgents have 20 percent chances of doing so, the rest of the time, they must rely on the transmission chain to guess the context.

We start with 180 PEAgent for a complete run (100 simulations), then we replace 20 PEAgents by 20 SimpleAgent for each consecutive run until no PEAgent

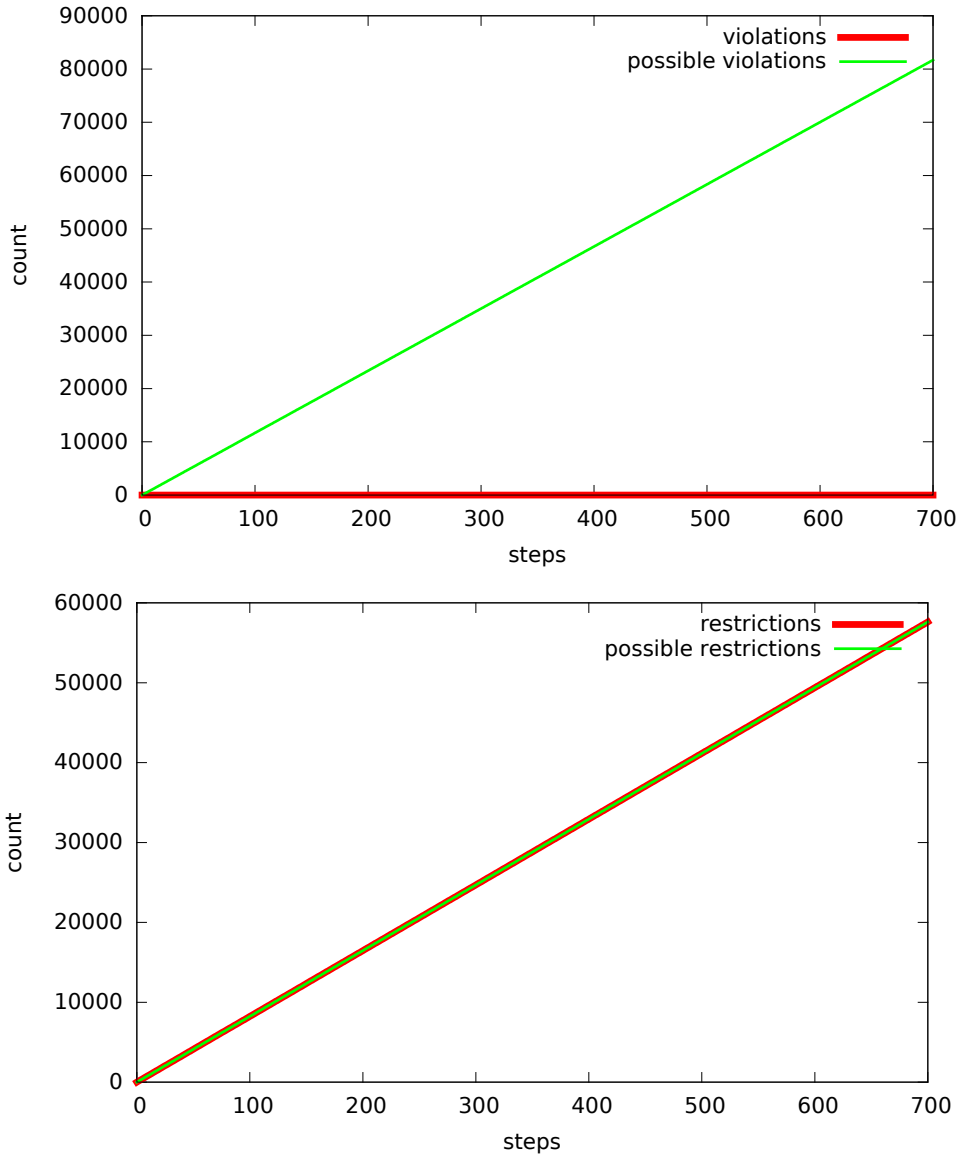


Figure 7.3: Violations and Restrictions wrt the number of Steps in the simulation: 200 NoTransAgents

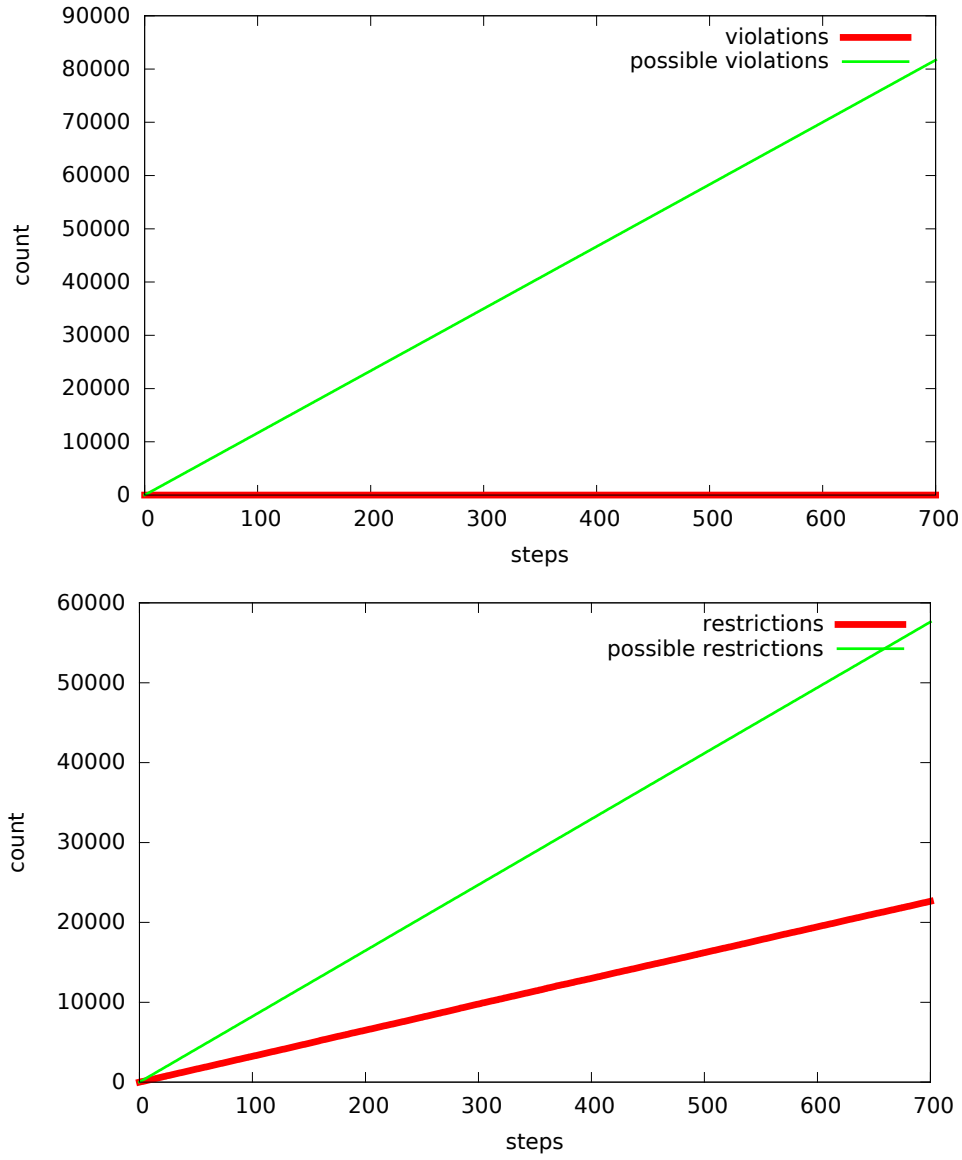


Figure 7.4: Violations and Restrictions wrt the number of Steps in the simulation: 200 PEAgentExpert

are left. We also include 10 PEAgentExperts and 10 NoTransAgent as control groups.

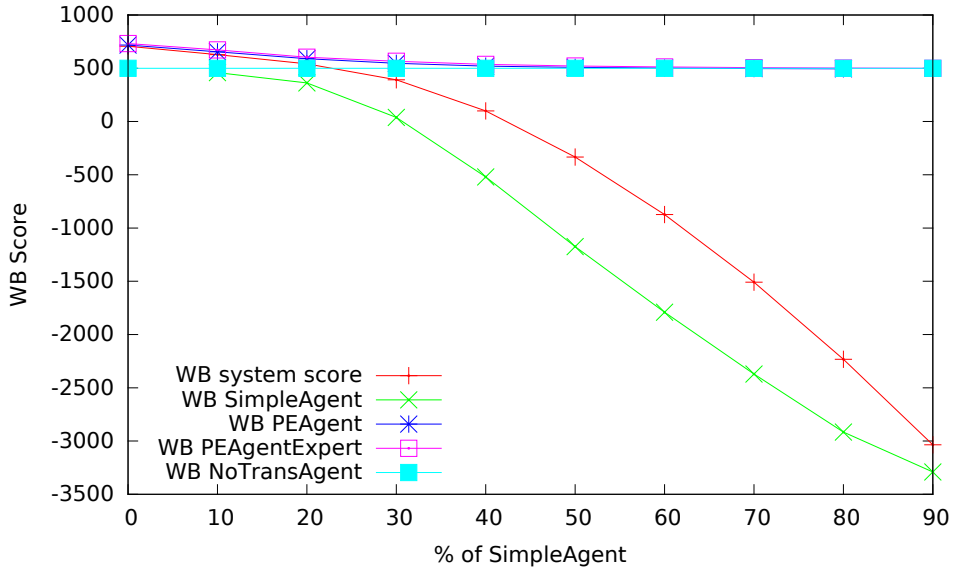


Figure 7.5: WB depending on the percentage of SimpleAgent

As figure 7.5 presents, the breaking point is between 20 and 30 percent of SimpleAgent. If there is more than 50 percent non-compliant agents in the system, as most of counterparts become untrustworthy, agents will reject messages and stop sending messages. The PEAgent WB score will match the NoTransAgent WB score as they end up on a similar strategy.

The system WB score and the SimpleAgent score really go down if there is more than 30 percent SimpleAgent. This is why we choose to deploy 60 SimpleAgents (30 percent) as a default in our experiments.

7.2.6 Standard Scenario

In this scenario we expect the 30 percent SimpleAgents to be excluded from the system after a few pieces of information are exchanged. In other words, there will be a few privacy violations and then it is supposed to stabilize as the violators are excluded.

The figure 7.6 presents the initial state and the final state of a small simulation (40 PEAgentExperts, 10 SimpleAgents). At the initial state, all connections (red links) are combined with reciprocal trust (green links). During the simulation, PEAgentExperts stop trusting the SimpleAgents that make violations (the green link disappears, the initial connection link remains). SimpleAgents (in red) get excluded by PEAgentExperts (in blue), the agent pointed by the yellow arrow is excluded, for instance. As we can see, subnetworks of SimpleAgents will subsist in the system (red arrow). We can also remark that it may happen that PEAgents exclude themselves if all of their neighbors are untrustworthy (the blue agent pointed by the blue arrow).

When we run the full experiment, we obtain the results presented in the following table and in figure 7.7. If we take a look at the number of violations, we can see that the number does not stabilizes. In fact, as information is spawned randomly among agents, it will happen that information is spawned on a SimpleAgent that is connected to another SimpleAgent and violations will happen in this subnet. Nevertheless, we can remark that the violation gradient is high during the very first steps, and goes down quickly as violators are excluded, as figure 7.8 shows. The figure 7.9 shows the evolution of the number of violations when information is spawned only on PEAgents (or PEAgentExperts), after a certain time almost no violation happens as all SimpleAgents have been excluded and almost no information is able to reach the SimpleAgent subnets[3].

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
396	45	550	570	500

7.3 Probing the PENs

In this section, we analyze experimentally the soundness of our model. We run the system while eluding some PENs from the PEA to demonstrate their usefulness. If the WB score is worse when a given PEN is removed, it means that the PEN is useful.

When all PENs are activated and there are 30 percent SimpleAgents, we obtain the following results for what we call the *standard run*:

[3]It happens that some SimpleAgents make their first violations very late in the simulation, this is why there are still a few violations that appears during time.

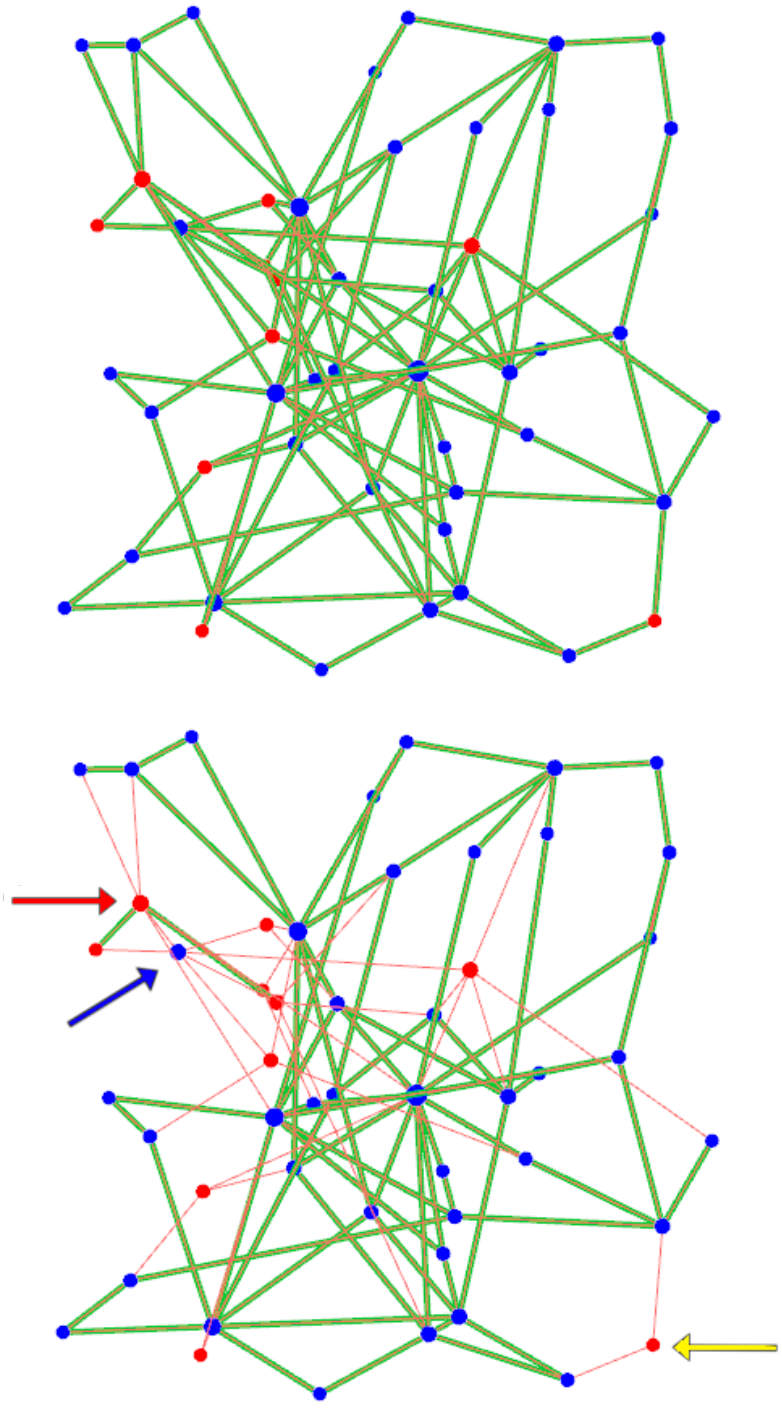


Figure 7.6: Initial state (top) and final state (bottom) of a simulation

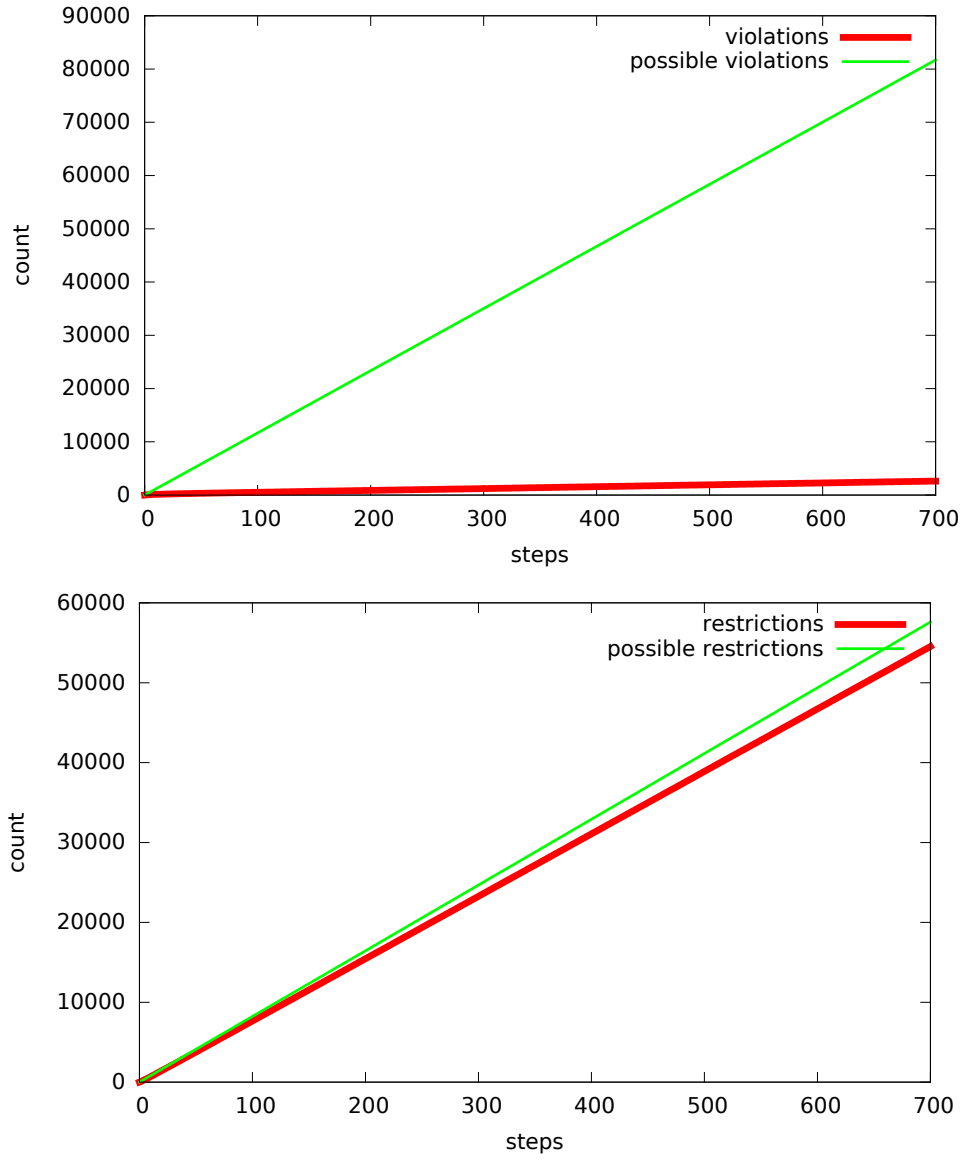


Figure 7.7: Violations and Restrictions wrt the number of Steps in the simulation: 120 PEAgents

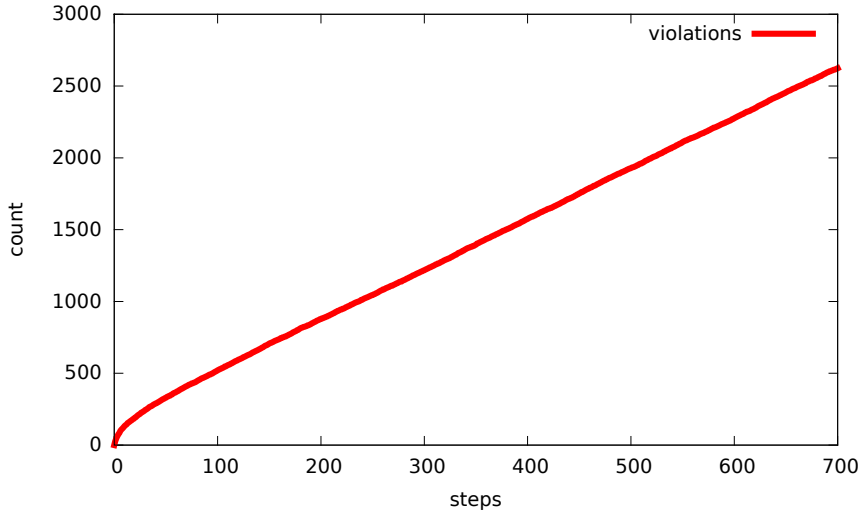


Figure 7.8: Number of violations when information is spawned randomly

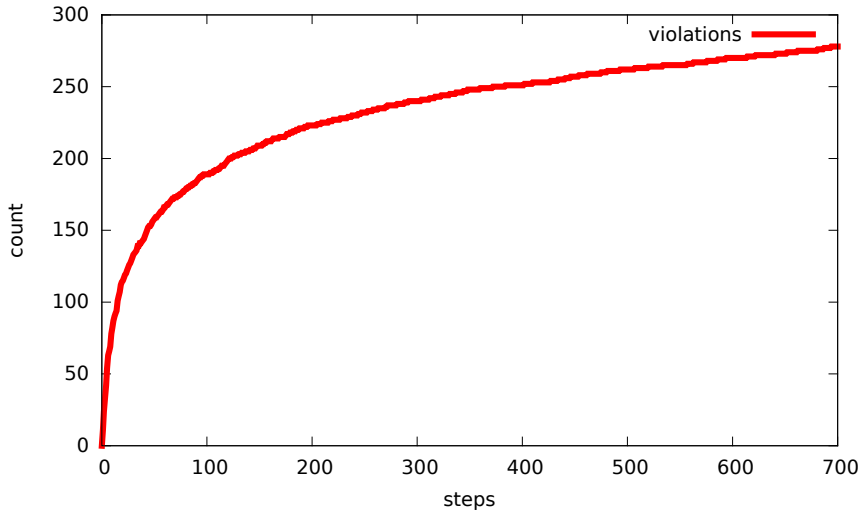


Figure 7.9: Number of violations when information is spawned randomly only on PEAgents

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
396	45	550	570	500

SimpleAgent violations are limited, they lost less than the 500 initial points. At the same time PEAgents and PEAgentExperts get better scores than the one they started with (500).

7.3.1 No Control on Sending

This subsection investigates what happens when PEAgents (and PEAgentExperts) do not control messages before sending them (PEN1 and PEN2). In this case, they can only rely on control when receiving.

Firstly, we remove PEN1: PEAgents do not check for violations when sending, but they do not send to untrusted agents.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
299	37	406	382	500

As we can see, the system WB score is lesser than the standard run score. Both PEAgent and PEAgentExpert have difficulties. When they are unable to determine the right context for an information, PEAgent do not send. Therefore, in this experiment they make less violations than PEAgentExpert that are always able to determine the context and send in any case (as no violation is detected when sending).

Secondly, we re-enable PEN1 but we allow PEAgents (and PEAgentExperts) to send messages to untrusted agents (PEN2).

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
311	-275	565	592	500

As the score demonstrates, it is worse than the standard run for the system when agents send messages to untrusted agents. On the opposite, it is better from their own point of view (individual PEAgent and PEAgentExpert scores are better), this happens because PEAgent and PEAgentExpert can forward messages that do not trigger violations to untrusted agents. Thus, this also explains the lower WB score for the system: as messages are transmitted to untrusted agents, those agents will make violations when they forward it. PEN2 was designed exactly in this purpose; to prevent messages from reaching violators.

We can see that the SimpleAgent score has gone real low. In fact, sending a message to an untrustworthy agent allows this agent to make violations, this

is why the SimpleAgent score is lower than in the standard run: they can make more violations.

7.3.2 No Control on Receiving

Symmetrically to what has been done in the previous subsection, PEAgents stop controlling when receiving messages (PEN3 and PEN4), they control only when they send.

If PEN3 is deactivated, PEAgents will accept messages without checking for violations.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
-926	-2625	-326	639	500

The system WB score is very low on this experiment. An interesting and non obvious fact pointed out by this experiment is that it is more important to check for violations when receiving than when sending. In fact, if messages are not checked for violations on reception, violators will not be detected. Thus, violators reputation will not go down and they will still be trustworthy. PEAgentExpert that rely only on their expertise to avoid making violations will only be slightly influenced by this measure. On the other hand, PEAgents that have a 20 percent chance of determining the context of a piece of information will rely on the transmission chain, and therefore, on context declared by agents that they believe to be trustworthy. In other words, PEAgents will declare erroneous contexts when forwarding messages, thus they will make violations. PEAgentExperts will not have this problem as they are expert in determining contexts, and as trust is not adjusted they will receive and send more messages than usual as everybody will be trusted (this explains the higher score for PEAgentExpert).

When PEN4 is deactivated, PEAgents will accept messages from untrusted agents.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
95	-624	392	446	500

As we can analyze from the data, it allows SimpleAgents to make more violations. PEAgents and PEAgentExperts lose points as they accept more violation triggering messages. When PEN4 is activated messages from untrusted agents are rejected to simulate PEN4 that asks to delete messages from untrustworthy agents.

7.3.3 No Punishments

PEN5 and PEN6 insure social control, in this section we remove these two norms to see how the system behaves.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
396	45	549	569	500

Removing PEN5 does not change the WB score of the experiment. PEN5 forces agents to check that the message has been signed by the sender. But in fact, in our simulation implementation, if a message is not signed, it is considered as inappropriate by PEN3. Therefore, PEN5 is already included in PEN3 this is why removing it has no effect.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
-477	-2043	155	333	500

On the opposite, removing PEN6 changes the WB score because agents are not punished anymore. All agents remain trusted.

7.3.4 Summary

As we have seen, we have been able to prove that most PENs have a important impact on the WB score of agents and of the system. The following table recapitulates all the scores from this last section.

Scenario	System	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
Standard	396	45	550	570	500
no PEN1	299	37	406	382	500
no PEN2	311	-275	565	592	500
no PEN3	-926	-2625	-326	639	500
no PEN4	95	-624	392	446	500
no PEN5	396	45	549	569	500
no PEN6	-477	-2043	155	333	500

As it has been shown in the previous sections, when some PENs are deactivated, it makes some other PENs useless. For instance, if messages are not checked for violations (PEN3), there is no more reference for adjusting trust.

Therefore, trust based PENs (PEN2 and PEN4) are made useless in this situation. This explains the higher impact that PEN3 has over the other PENs.

On the same topic, PEN6 produces more or less the same effect as trust is not adjusted. Nevertheless, violations still be detected and messages causing them are deleted (PEN3). This restricts the number of messages that circulates in the system and lowers the impact of PEN6 removal with regard to PEN3 removal.

7.4 Attacks and Solutions

The simulations presented in this section are looking forward to test the model resistance to some attacks.

7.4.1 Meta-Information Tampering

An attacker can decide to delete meta-information attached to a message. This will delete the message history, it will also remove the declared context from previous agents. We obtain the following WB scores when we set the SimpleAgents in such a way that they remove meta-information from messages.

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
374	31	522	553	500

The score is a bit lower than the standard run, the main reason is that PEAgents must rely on their own experience more often as they cannot infer the context of the information from the chain of transmission (that has been removed).

One should note that it is possible to remove the chain of transmission but it is impossible to falsify it: every link is signed by its author and contains a reference to the recipient and to the information it is related to.

7.4.2 Reputation Tampering

In this experiment, we replaced the 60 SimpleAgents from the standard scenario by FoolingAgents. These agents goal is to make a lot of violations without being excluded. They achieve that goal by sending positive reputation messages for every transmission they participated in. Therefore, they reinforce the reputation

of all agents in the system, including other FoolingAgents who receive a "capital" that they can use to make violations.

System WB Score	FoolingAgent	PEAgent	PEAgentExpert	NoTransAgent
97	-544	533	569	500

As we can see, the FoolingAgents manage to take the system down even more than the SimpleAgents do (see the Standard Scenario).

When we designed this attack, we knew it will be efficient against the PrivaCIAS experimental implementation. The trust model being used by PEAgents is very simple and thus, can be fooled easily.

A more robust trust model would allow to prevent this attack from being efficient. It would take into account reputation messages in a way that makes them a lot less valuable than direct experiences. This way, violators will be excluded like it was the case with SimpleAgents.

7.4.3 Context Tampering

We define a new type of agents, named ContextLyingAgents. Those agents declare wrong contexts to fool other agents into making violations. PEAgent, when they are unable to determine the context of an information, will rely on the declared context to find the most appropriate. To be more efficient in their attack, those agents remove the chain of transmission (containing previously declared contexts).

System WB Score	ContextLyingAgents	PEAgent	PEAgentExpert	NoTransAgent
381	282	410	529	500

The attack is moderately efficient. The 30 percent ContextLyingAgents may not be enough to successfully take the system down. Nevertheless, we can see that the WB score is lower than in the standard run. PEAgents are fooled into making violations, as we can see, their score is lower than in the standard run (410 instead of 550). PEAgentExperts resist quite well to this attack as they are always able to determine the real role-context of an information. Thus, the PEAgentExperts do not get fooled.

7.5 Chapter Conclusion

All along this chapter, we ran different simulations to experiment with the PrivaCIAS model.

We introduced different metrics, such as the number of violations, and also the Well Being (WB) score: an evaluation of each agent that allows to get a general evaluation of the system or a type of agent.

Firstly, we have shown that with agents that can determine the context of an information only 20 percent of the time, the system gives satisfying results when there is at most 30 percent non-compliant agents.

Secondly, we have been able to prove that all PENs (except PEN5, due to the implementation) are useful and thus that removing any of the PENs reduces PrivaCIAS efficiency.

Finally, we proposed different attacks to our system, the most efficient attack was trust based: the system was attacked by sending false evaluations to agents in order to prevent violators from being excluded. The trust model that was used in these experiments is too naive to resist such an attack.

Chapter 8

The PrivaCIAS Photo-Sharing Application

A friend to all is a friend to none.
Aristotle

In the previous chapter, we presented an application developed using RePast that simulates a PrivaCIAS powered multi-agent system. While it allows to experiment with the PrivaCIAS specification, it may be hard for the reader to imagine how a real system using PrivaCIAS could be put into place.

We defined a photo sharing application that was implemented on Android mobile phones. Andrei Ciortea was put in charge of the implementation during his Master thesis. The result is an assistant agent implemented in an android application. The application communicates with a decentralized system of servers, called nodes.

We define in this chapter the show case application for the PrivaCIAS specification that has been developed.

The first section presents the specification of the application we wanted to be developed. The second section describes the application as it was implemented.

Contents

8.1 Specification of the PrivaCIAS Demonstration 147
8.2 The PrivaCIAS Photo-sharing Application 148
8.2.1 Incompatible Relationships 148
8.2.2 Application Components and Structure 149
8.2.3 Illustrated Scenario 151
8.3 Chapter Conclusion 153

8.1 Specification of the PrivaCIAS Demonstration

The demonstration has already been discussed as an example in previous sections 5.4 and 6.4, we describe it with further details here to allow its implementation.

As our main application domain is social networks, we propose a photo-sharing social network application. In this application, users can share pictures with their contacts on the social network. We want to provide users with a decentralized social network, or at least a semi-decentralized network.

The client application should be able to deploy on smartphones. Mobile devices can be seen as personal assistants, they include most of their predecessors Personal Digital Assistants (PDA) functionalities. Therefore, mobile devices are a good support for assistant agents, such as Privacy Enforcing Agents.

Users, once they install the client application, can add contacts to their "contact list". Following the PrivaCIAS specification, the contact who is getting invited should confirm the invitation or reject it. If an invitation is accepted, both assistant agents are supposed to trust each other.

Users in the system can also create role-contexts and invite others to join in. The creator of a role-context maintains a registry of users participating into it.

Users can take pictures with their smartphones from the provided application. They can forward that picture to their contacts. When they decide to do so, the PrivaCIAS specification is used to prevent violations when sending messages.

Firstly, the user specifies a role-context for the picture (the role-context to be declared by the sender, according to the specification) from the role-contexts he belongs to. If the user is the target of the message, he can attach preferences to the message. For simplification reasons, the preferences are reduced to *forbidden* or *mandatory* recipients: the information cannot, or can only be, exchanged with the given users.

The user should decide to which contacts the picture is to be sent, and the assistant agent should check the PENs when sending the message. If a violation is detected, the user is prompted for a decision.

When the other user receives the picture, his agent checks the PENs. As the agent is not able to detect if the declared context is compatible or not with

the nature of the information (A-law number 1), the picture and the declared role-context are presented to the user. If the user thinks the declared context is incompatible with the picture, he is able to flag it as incompatible. For instance, a picture took at work is declared to be in the *family* context. The agent presents the picture to the user that decides to flag the picture as *incompatible*. Therefore, the A-laws do not hold, the transmission is inappropriate and the message is deleted. When such a violation is detected, the trust model of assistant agents must be adjusted and other agents must be informed of the violation.

8.2 The PrivaCIAS Photo-sharing Application

The work presented in this section has been realized by Andrei Ciortea during his Master thesis, under our supervision.

8.2.1 Incompatible Relationships

As it was discussed in Section 5.1, we put aside the *incompatible relationships* during the thesis as we considered it to be a specific case needing dedicated research work. As a reminder, it is said that *Agents must not have incompatible relationships with the target*. An example is the case where A has an illness, and B is both a medical doctor and A's boss. It may be inappropriate for B to know A's disease in that situation.

Andrei Ciortea did research, for his master thesis, on this issue: what is an incompatible relationship?

He had to develop the PrivaCIAS show-case application to demonstrate his additions to the PrivaCIAS specification.

From a theoretical point of view, Ciortea adds a new dimension to the PrivaCIAS specification: relationships between roles. His central idea is that if there is a sensitive relationship between two agents, like an authority relationship, it forbids any communication that does not belongs to the context related to the sensitive relationship. In our previous example, A and B have an sensitive relationship in *work*, therefore, other context like *health* are excluded for these two agents when they communicate.

Andrei represents these relationships using MOISE. This explains why Ciortea

decided to use JaCaMo[1] for implementing the PrivaCIAS specification. JaCaMo is a framework including the Jason[2] agent programming language, the CArtago artifact infrastructure and Moïse[3] organizational model. Andrei used the JaCaMo framework for the server application, and the JaCa-Android[4] framework for the mobile application.

Interested readers can refer to Andrei Ciortea's Master thesis [Ciortea 2011] for more details.

8.2.2 Application Components and Structure

The application is supposed to be decentralized. Nevertheless, running servers on Android powered phones have proven to be very difficult. For this reason, a semi-decentralized architecture was chosen. We define servers, that we call nodes, that can communicate with their users or synchronize with other nodes (*cf.* fig 8.1).

Users chose a node when they register to the system. The node keeps the data related to the users that registered on it. A better option would be to bring the Privacy Enforcing Agent to the node instead of having it into the mobile phone. It would also be necessary to have one node per user. From a privacy point of view it would be better, but from a practical point of view it would be a source of trouble for users if every one had to start their own node. Therefore, the application that has been implemented relies on a few nodes, and each node hosts multiple users.

Users can be designated by a unique number (on the node) and the node address, for example: `15@192.168.0.1`.

The nodes are composed by system agents and artifacts that handle users requests. As artifacts and organizational models are not the topic of this thesis, the reader can refer to Andrei Ciortea's Master thesis [Ciortea 2011].

Thanks to the JaCa-Android framework, the Privacy Enforcing Agent core is implemented in Jason (AgentSpeak), to be deployed on Android phones.

To illustrate the implementation, this is the code that checks PEN1 and the

[1] http://jacamo.sourceforge.net/
[2] http://jason.sourceforge.net/
[3] http://moise.sourceforge.net/
[4] http://jaca-android.sourceforge.net/

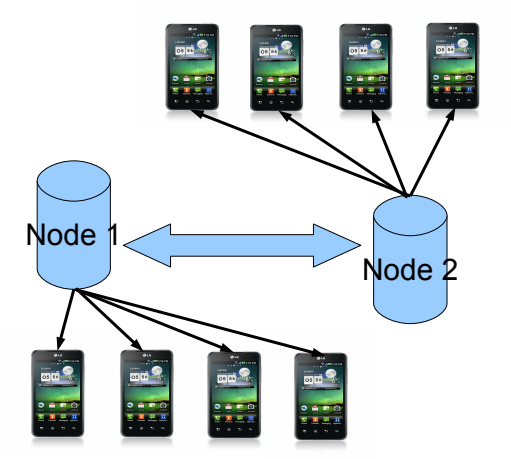


Figure 8.1: Overview of the application architecture.

A-Laws within the assistant agent:

```
/*      Check PEN1      */

checkPEN1(Message) :-
  appropriate(Message).

appropriate(Message) :-
  fit_context(Context, Message) &
  fit_role(Context, Message) &
  fit_policy(Message).

fit_context(Context, Message) :-
  information(Message, Info) &
  propagator(Message, Propagator) &
  contexttag(Message, Propagator, Context) &
  context(Info, Context).

fit_role(Context,Message) :-
  extract_sender(Message,Sender) &
```

```
20    play(Sender,_,Context) &
      extract_receiver(Message,Receiver) &
22    play(Receiver,_,Context).

24  fit_policy(Message) :-
      information(Message, Info) &
26    target(Info, Target) &
      policy(Message, Target, Policy) &
28    .print(Policy) &
      policy_valid(Policy, Message).
30
    fit_policy(Message) :-
32    information(Message, Info) &
      target(Info, Target) &
34    not(policy(Message, Target, _)).
```

The code, in AgentSpeak for Jason is very close to the definition of A-laws that we give in this thesis[5], because the AgentSpeak syntax is close to the Prolog one.

Thus, it makes it easier for an agent programmer to implement directly from the formal definition of A-laws and PENs. For instance, the `information(Message, Info)` predicate checks the agent belief base for a belief matching "Message" to retrieve the corresponding "Info".

8.2.3 Illustrated Scenario

John logs into the PrivaCIAS system through the node he registered on (Figure 8.2. He has no contact yet. John decides to add Bob and Alice as a contact. John knows Bob and Alice addresses: Bob address is 13 and Alice is 14 (addresses are integers so far).

Alice logs in, and accepts the request from John. Bob does the same.

Bob wants to send a message to Alice. He takes a picture of his kangaroo plush. The application shows him the *edit message* screen (Figure 8.3).

The message is set to be in context 1 (friends context) the information nature is declared to be work, the target of the message is Bob (number 13) and the message is sent to 14 (Alice). A button allows to define policies (Figure 8.4 and 8.5). We want to add a specific rule that forbids 15 (John) from receiving

[5]Even though the code is based on an earlier version of the PENs and A-laws.

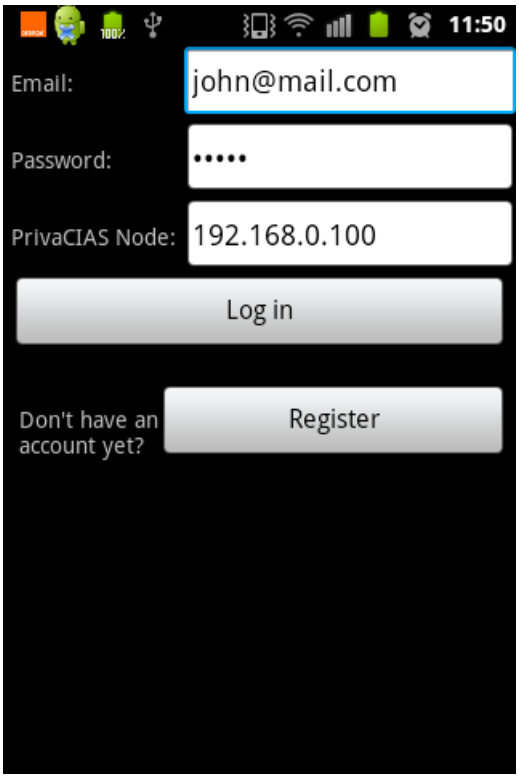


Figure 8.2: John login screen

the message. Bob sends the message. No privacy violation is triggered.

When Alice logs in, she can see that she received a message (Figure 8.6). When Alice is presented with the message, she can see that the information is declared to be of a *work* nature when it is transmitted in a *friend* context. If she finds it incoherent, she is able to flag the message as inappropriate (regarding A-Laws). She does not need to know the A-laws, she just needs to flag the message as inappropriate if she finds incoherence in the meta-information that is presented to her.

Alice forwards the message to John. There was a policy preventing John from being sent the information that was defined by Bob, the target of the message. Therefore, when Alice tries to forward it to John, a privacy violation is detected and she is prompted to continue or abort the transmission (Figure 8.7).

If Alice continues, John's agent will detect the violation and adjust Alice's

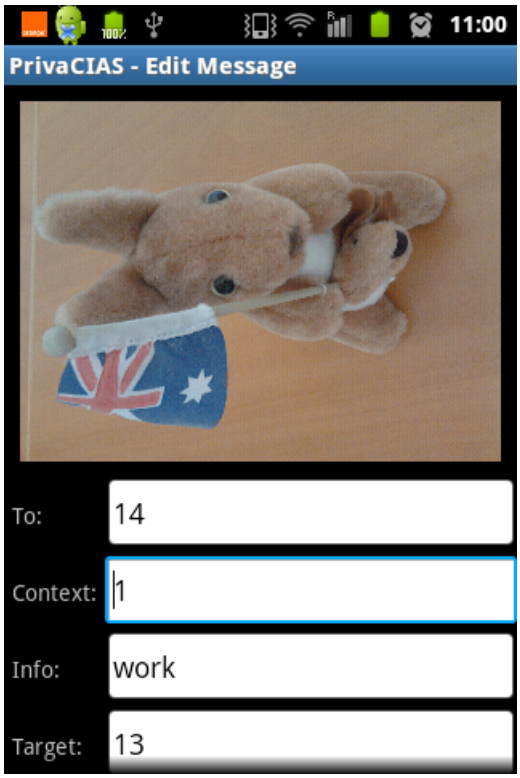


Figure 8.3: Edit message screen

trust accordingly. If Alice keeps on making violation, she will become untrusted and all messages coming from or directed to her will be detected as privacy violations by the application.

8.3 Chapter Conclusion

In this chapter we show how the PrivaCIAS specification can be implemented to a real application.

We defined a social network that is used to share pictures with acquaintances. These acquaintances belong to role-contexts that are created and maintained by users themselves.

Users are able to take pictures and share them in the network. The PrivaCIAS specification is implemented in users smartphones to provide them with a privacy-

Figure 8.4: Edit message screen (cont.)

aware assistance. When they send or receive pictures, the PrivaCIAS specification is used to determine if the transmission triggers a privacy violation.

Most of the PrivaCIAS specification is transparent to the user. The user is only prompted when a violation is detected. The user is also allowed to flag pictures that seem inappropriate in the role-context they were received. This allows the PrivaCIAS specification to detect some violations that would have required context detection through image analysis or other techniques that are not yet mastered.

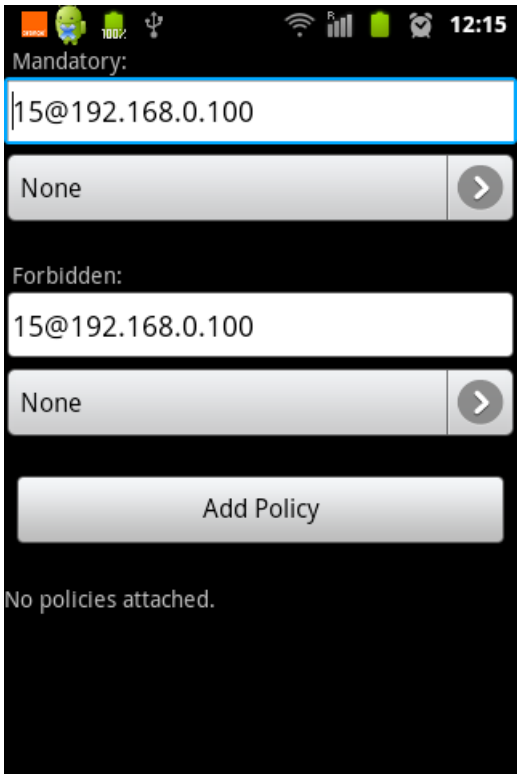


Figure 8.5: Policies screen

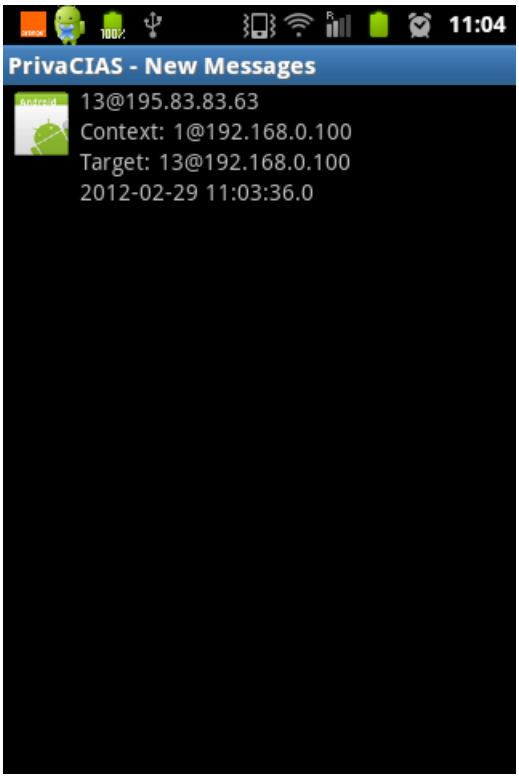


Figure 8.6: Message List

Figure 8.7: Violation Warning

Conclusion

Our objective was to provide open and decentralized systems with a privacy protection. Our main topic of interest was to import concepts from social sciences to multi-agent systems to achieve privacy protection.

Contribution

As far as we know, our thesis is the first to tackle the problem of privacy in open and decentralized systems by using socio-inspired theories such as contextual integrity, norms and trust.

In this thesis, we have shown that existing techniques were not applicable to open and decentralized systems. For instance, a lot of techniques only consider the "big brother" aspect of privacy or rely on central authorities. The "big brother" problem (vertical privacy) is eluded as we are working on decentralized systems, therefore we bring a control for protecting users from other users in the system.

The most appropriate way of controlling agents in a decentralized system is through social control, a control that relies on trust and reputation. To bring a control based on trust and reputation, applicable to our problem, we needed to instruct agents what is a trust defection. The theory of contextual integrity, from Helen Nissenbaum [Nissenbaum 2004] has been formalized to define the concept of *privacy violation*. This formal definition of privacy violation has been named *the Appropriateness-laws*. Agents are instructed that a privacy violation marks someone as untrustworthy through the definition of Privacy Enforcing Norms (PENs). The PENs also describe what behavior is expected from agents in the system.

Aside from these definitions of PENs and Appropriateness laws, the model defines other components such as a set of predicates that allows agents to manipulate PrivaCIAS concepts when communicating with each other.

The model leaves open some aspects to the developer of the system. For example, no assumption is made on the content of the *information* part of a message. Therefore, the model tries to allow the widest range of possible application.

As the model is not constraining to the developer, any agent provider can develop agents as long as they follow the PENs. If they do not comply, agents

that do follow the PrivaCIAS model definition will take them out of the system by distrusting them and next by ignoring them.

We have shown in experiments that the system works as long as a consequent amount of the users complies with the PrivaCIAS specification. We have also shown that the model has some limitations and is sensitive to a large number of malevolent agents (agents that violates the PENs) or to staged attacks.

An android based application has also been developed to show that our model is not a paper tiger and that it can be implemented easily on a real application.

We addressed the two main issues for protecting privacy in a decentralized social network:

- **How to detect privacy violations from the user point of view** and give tools to users to protect their data against privacy violations?
  - We gave a definition of privacy violation based on Contextual Integrity theory (the A-laws) and provided users with assistant agents to make them privacy-aware.
- **How to control a system without a central authority** in order to prevent and stop privacy violations in the system?
  - We defined a set of norms (the PENs) that allows agents to participate in the control of the system in a completely decentralized way.

Future Work

This thesis is, in our opinion, a junction of multiple theories and techniques from different domains. This approach was necessary for the development of a coherent privacy protecting specification. Nevertheless, it cost us the possibility of going in depth on specific components of the thesis.

As Figure 8.8 summaries, we already started to explore some components more in depth. During the beginning of the PhD, we worked on the ForTrust project, which aimed at defining a formal model of trust based on Castelfranchi's social trust. A future improvement could be to integrate directly the ForTrust model into the Privacy Enforcing Norms.

Future developments can be made on most of the components presented on Figure 8.8, as discussed hereafter. From what we read in Nissenbaum's writings [Nissenbaum 2010, Nissenbaum 2004], there is a concept that seems easy to integrate and yet powerful: context-relative norms. We could integrate it to the PrivaCIAS model by allowing the role-context owners to define default preferences for a given role-context. In other words, we would define another Appropriateness-law that states: Respect context-relative preferences. This way, for any message relative to a given role-context, the default preference for this context would apply. As an example, any message in role-context medicalDoctor@ChiefMD would have a default preference stating that the message can only be sent to agents playing the role-context medicalDoctor@ChiefMD or to the target of the message (the patient). This would be an improvement on multiple aspects: Roles and Contexts, but also Violation Detection.

Another aspect of violation detection has been explored and is presented in Andrei Ciortea's master thesis [Ciortea 2011]. Ciortea worked on the problem of "incompatible relationships", that we can reformulate as "conflict of interest". He studied how we can express norms or add an A-law that prevents privacy violations through incompatible relationships. This problem occurs, for instance, when your father in law is also a medical doctor. Personal medical information may be accessed by your father in law under the role-context medicalDoctor which may cause a privacy violation. So far, PrivaCIAS would not detect this as a violation unless the target explicitly declared that her father in law should not have access to the information (through a preference).

Two difficult points in the PrivaCIAS model are to determine who is the target of an information and what is an appropriate role-context for this information. For this task, we rely on the user and what other users have declared previously. We do believe that automated techniques can be brought into place for some specific types of information. For instance, if informations were restricted to pictures of user faces, automated face recognition techniques could be applied to determine who is on the picture, hence the target identity. Machine learning could also be used to allow automated role-context classification for textual information under certain limits. In the end, information analysis would be done automatically, without needing to refer to the user.

PrivaCIAS could also be completed by a finer integration of roles and context, instead of using role-contexts. It is possible to integrate PrivaCIAS with an

organizational model that would take in charge the roles and contexts of the agents.

We demonstrated that our specification is sound by developing a show case and experiments. Nevertheless, as it is a new approach to the privacy problem, it opens way to a lot of improvements on the different aspects discussed in this last section.

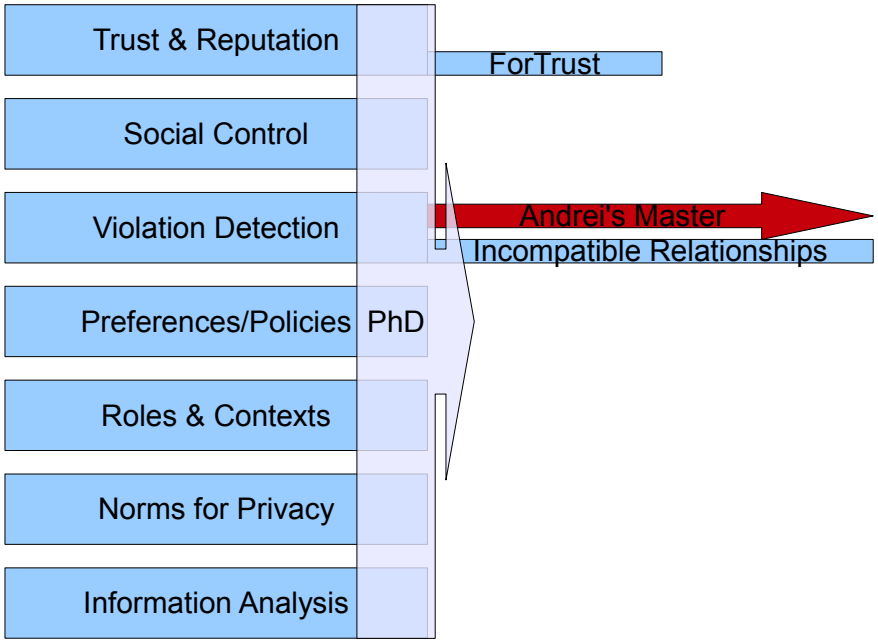


Figure 8.8: Work and Perspectives

Appendix A

PrivaCIAS : Privacité selon l'intégrité contextuelle dans les systèmes agents décentralisés

---

A.1 Introduction

Cette thèse présente PrivaCIAS : Privacité selon l'Intégrité Contextuelle dans les Systèmes Multi-Agents Décentralisés. Nous montrons qu'il est possible de protéger la privacité des utilisateurs dans un système ouvert et décentralisé. Au lieu de contraindre les utilisateurs ou les développeurs, nous proposons un modèle qui met en place un ordre social afin de préserver la privacité.

Besoins Nous souhaitons protéger la privacité des utilisateurs d'un système ouvert et décentralisé. Nous montrons dans cette thèse que les approches existantes pour la privacité ne sont pas satisfaisantes pour résoudre ce problème. Le domaine d'application visé est celui des réseaux sociaux décentralisés. Les deux principaux verrous scientifiques que la thèse vise à résoudre sont :

- Comment détecter une violation de privacité ?
- Comment contrôler un système ouvert et décentralisé afin de prévenir et stopper les violations de privacité ?

Objectifs Nous voulons définir des agents assistants qui puissent avertir les utilisateurs lorsqu'une transmission d'information déclenche une violation de privacité. Comme le système est décentralisé, il faut trouver une technique qui ne repose pas sur un élément central mais qui repose sur la participation des agents au fonctionnement du système.

A.2 État de l'Art

A.2.1 Privacité dans les sciences sociales

Avec les systèmes d'information qui envahissent notre quotidien, la privacité est devenue une source d'inquiétude majeure à laquelle les différents acteurs de la société (politiques, scientifiques...) s'intéressent.

Dans cette sous-section, nous nous intéressons à la privacité du point de vue de la société et des sciences sociales.

Les attentats du 11 septembre ont joué un rôle primordial sur la perception de la privacité par la société, comme il a été noté par plusieurs auteurs [Solove 2007, Westin 2003, Baase 2007]. Afin de se prémunir contre de nouvelles attaques terroristes, le gouvernement des USA a mis en place des lois liberticides, dont le fameux Patriot Act. Dans le même temps, les technologies de l'information ont pris leur envol, rendant d'autant plus facile l'espionnage des citoyens par les gouvernements. Cette conjoncture a mis en alerte les citoyens et leurs représentants politiques quant aux dangers que peut représenter le non-respect de la privacité.

L'inconvénient est que le concept de privacité est flou, d'après Helen Nissenbaum [Nissenbaum 2004] : « la notion de privacité varie selon la culture, l'époque et la société ». Sara Baase, dans son livre *A gift of fire* donne la définition suivante de la privacité [Baase 2007] :

- liberté contre les intrusions–liberté de ne pas être dérangé,
- contrôle des informations personnelles
- liberté contre la surveillance

Un autre inconvénient est que les gens comprennent mal le concept de privacité. Daniel Solove, rapporte dans son article [Solove 2007], que lorsqu'on parle de privacité, une remarque commune est la suivante : « Si vous n'avez rien à cacher qu'avez-vous à craindre ? ». Solove analyse cette réponse comme la preuve d'une incompréhension du concept de privacité. Un contre exemple facile à cette remarque est le dossier médical d'une personne. Une personne peut avoir une maladie qui n'est ni honteuse, ni illégale, mais le fait de rendre publique cette information peut nuire au sujet de l'information. Un recruteur, par exemple, pourrait renoncer à recruter quelqu'un dont il sait que l'état de santé est mauvais.

De plus, la remarque précédente tend à mixer différents concepts entre eux. Ainsi un individu n'aura peut-être « rien à cacher » aux services de police, mais si les mêmes informations tombent entre les mains de ses voisins, il sera probablement très agacé.

Dans cette thèse nous différencions deux types de privacité [Grimmelmann 2008] :

- La Privacité Verticale (une autorité accède aux données d'un individu)
- La Privacité Horizontale (un individu accède aux données d'un autre individu)

Pour illustrer ces deux types de privacité, prenons l'exemple de Facebook. Si le fournisseur du service Facebook accède aux données des utilisateurs c'est un problème de privacité *Verticale*. Si un utilisateur accède aux données d'un autre utilisateur, c'est un problème de privacité *Horizontale*.

En dehors de ce problème d'incompréhension des concepts de privacité, Nissenbaum et Schoeman soulignent dans leurs articles respectifs [Nissenbaum 2004, Schoeman 1994] que les législateurs devraient mettre l'accent sur le concept de contexte. Un exemple donné par Schoeman est qu'une personne peut être active dans la *gay pride* de San Francisco, mais rester confidentiel à propos de ses préférences sexuelles vis-à-vis de sa famille et ses collègues à Sacramento. Le fait d'apparaitre publiquement comme *gay* lors de la *gay pride* de San Francisco ne signifie pas que la personne renonce à ses droits dans un autre contexte. Ces auteurs rejettent la dichotomie qui consiste à dire qu'une information est soit privée soit publique.

Nissenbaum, en s'appuyant sur ces concepts propose une théorie de l'intégrité contextuelle, dont le concept de violation est résumé comme suit :

« Savoir si une action est une violation de privacité ou non est fonction de :

- la nature de la situation/contexte,
- la nature de l'information par rapport au contexte,
- le rôle des agents qui reçoivent l'information,
- les relations entre le sujet de l'information et l'agent qui reçoit cette information,

- les termes de dissémination définis par le sujet ».

Dans la vision de Nissenbaum, il n'y a plus de distinction entre information sensible ou non. Une information peut générer une violation de privacité si elle est échangée dans un contexte inadapté.

A.2.2 Technologies pour le contrôle de l'Information

Dans cette sous-section, nous passons en revue les principales techniques utilisées pour le contrôle de l'information. Ces techniques peuvent en général être appliquées à la protection de la privacité.

- La Cryptographie. La cryptographie consiste à chiffrer un message. Elle permet de protéger une information contre un espion (historiquement) ou d'une tentative d'accès non autorisée. Néanmoins, elle ne permet pas de se prémunir d'un mauvais comportement d'une des personnes à qui on aurait donné accès à l'information (qui pourrait alors la recopier et la retransmettre). On peut citer Pretty Good Privacy (PGP) [Zimmermann 1995] comme application de la cryptographie à la privacité.

- Le contrôle d'accès. C'est la technique couramment utilisée sur les réseaux sociaux en ligne. L'accès a l'information est autorisé à certains utilisateurs suivant le rôle (Role Based Access Control [Ferraiolo 1995]) qu'ils jouent dans l'organisation (sur Facebook par exemple, les amis peuvent accéder aux photos d'un individu). Le contrôle d'accès repose sur une autorité centrale qui a la mainmise sur l'ensemble des informations des utilisateurs. De ce fait, le problème de privacité vertical est proéminent dans ce type d'approche.

- L'anonymisation. C'est une technique couramment utilisée, notamment pour les sondages et autres enquêtes où la privacité des individus est protégée en éliminant des données (principalement le nom du participant). Plus on retire de l'information, plus la privacité des participants est protégée. Cette technique est inapplicable lorsque les identifiants doivent être conservés (par exemple dans un système de messagerie, il est indispensable de savoir le nom de l'émetteur). De plus, il faut se méfier de la « réidentification » qui consiste à retrouver l'identité d'une personne en croisant

les données disponibles (par exemple pour un sondage : age, sexe, taille, occupation...)

- Politique de confidentialité et P3P. Il s'agit d'une définition des politiques que l'organisme collectant les données s'engage à respecter. Le problème est alors de vérifier que l'organisme applique réellement la politique qu'il a déclaré. Cette technique est donc pour l'instant inapplicable de manière automatique. Néanmoins, la plateforme de préférences de privacité (P3P [Reagle 1999]) développée avec le W3C permet au navigateur web de lire automatiquement les politiques afin de vérifier qu'elles correspondent aux préférences de l'utilisateur.

- Bases de données Hippocratiques. Agrawal a proposé d'appliquer les grands principes du serment d'Hippocrate aux bases de données [Agrawal 2002]. L'idée est que le stockage de l'information dans une base de données personnelles doit suivre dix principes : spécification du but dans lequel l'information est collectée, consentement, collection de données limitée au stricte nécessaire, utilisation limitée au stricte nécessaire au regard du but, diffusion limitée au stricte nécessaire, rétention limitée au stricte nécessaire, véracité/précision des données, sécurité, ouverture (droit d'accès aux données personnelles), respect (vérification par un utilisateur du respect des dix principes). Ce système est destiné à résoudre un problème vertical, mais est peu adapté à un problème de privacité horizontal.

- SMA Hippocratiques. Les SMA Hippocratiques [Crépin 2009] proposent une adaptation des 10 principes précédents aux systèmes multiagents. L'approche est alors beaucoup plus adaptée aux problèmes de privacité horizontale. Néanmoins, il est relativement difficile de vérifier qu'un agent tiers se conforme bien aux 10 principes et qu'il ne revend pas (par exemple) les informations personnelles à un autre agent.

- Digital Right Management. Les DRM sont une technique d'informatique de confiance reposant sur une autorité centrale qui gère les autorisations d'accès aux données. La technique repose sur l'utilisaton d'un logiciel de confiance par chaque utilisateur. Les données ne sont accessibles aux utilisateurs qu'a l'intérieur de ces logiciels et ne peuvent donc pas être retrans-

mises ou modifiées sans l'accord de l'autorité. Cette technique repose donc sur une autorité centrale qui gère toutes les données.

- Politiques Collantes. L'idée de cette technique est de pouvoir attacher à une information des politiques qui ne soient solidaires de cette dernière [Mont 2003]. Ceci est réalisé en utilisant les DRM. L'approche est orientée spécifiquement vers la protection de la privacité, néanmoins, on retrouve les même problèmes que pour les DRM, notamment le besoin d'une autorité centrale.

- Tiers de Confiance pour la Privacité. Piolle propose l'utilisation d'agents de confiance qui soient utilisés pour le traitement des informations [Piolle 2009]. Un agent dont le code est vérifiable et qui est certifié par des méthodes d'informatique de confiance (DRM) s'occupe d'effectuer le traitement requis sur l'information. Ainsi, seul l'agent de confiance a accès à l'information, les participants à la transaction reçoivent de l'agent de confiance un message indiquant que la transaction a réussi ou échoué, rien de plus. La privacité est ainsi protégée, par contre ce système empêche de réaliser des communications inter-agents qui soient protégées.

Les différentes techniques existantes ne sont pas applicables à un réseau social décentralisé et ouvert comme nous le souhaiterions. Certaines techniques nécessitent une autorité centrale, d'autres protègent la privacité mais empêchent les agents de communiquer entre eux. De plus, peu de techniques abordent la possible retransmission d'un message après qu'il ait été transmis d'un agent à l'autre.

A.2.3 Contrôle des agents dans un système multiagent

Comme il est impossible d'appliquer les techniques existantes de protection de la privacité dans un système multiagent décentralisé, nous explorons ici les techniques qui sont généralement utilisées pour contrôler les agents dans un système multiagent.

Les *normes* expriment le comportement *normal* attendu de la part des agents afin que le système atteigne son objectif. Si cet objectif est de protéger la privacité des utilisateurs, les normes définiront une violation de privacité comme un comportement anormal. Contrôler les agents consiste à éviter les comportements anormaux ou les réprimer.

Raimo Tuomela [Tuomela 1995] divise les normes en quatre catégories :

- les Règles (r-norms) sont des normes strictes et explicites. Ce sont des lois imposées par une autorité.
- les normes Sociales (s-norms) sont implicites. Elles représentent des croyances communes de ce qu'il faut ou ne faut pas faire dans le système.
- les normes Morales (m-norms), font appel à la conscience de chacun. C'est une application de la règle d'or « ne fait pas à autrui ce que tu ne voudrais pas qu'on te fasse ».
- les normes de Prudence (p-norms), tentent de maximiser l'utilité de chacun. Par exemple, ne pas faire confiance à quelqu'un qui n'en est pas digne.

Pour chaque type de norme, il existe une sanction correspondante. Des r-sanctions s'appliquent aux r-norms, des s-sanctions aux s-norms... Mais une r-norm peut également être supportée par des s-sanctions. En effet, les agents dans le système auront comme croyance commune qu'il faut respecter les r-norms, donc que violer cette dernière est immoral et viole une s-norm correspondante. A l'inverse, les r-sanctions (sanctions par une autorité, par la loi) ne s'appliquent pas aux s-norms (normes sociales). Ainsi, si quelqu'un ne répond pas lorsqu'on le salue (violation de s-norm), il ne sera pas puni par la loi (pas de r-sanction). Si quelqu'un viole la loi (violation de r-norm) cela sera perçu comme immoral dans la plupart des cas et nuira socialement à l'individu (s-sanction).

Les normes peuvent ensuite être mises en place par différentes techniques. La première est appellée *régimentation* : les agents sont physiquement contraints de suivre les normes. Les normes peuvent être directement codées en dur dans les agents ou dans le système pour empêcher toute violation. Cela repose souvent sur des systèmes non-ouverts et centralisés. La seconde est de passer par des Institutions électroniques ou des Organisations (on peut citer MOISE+ [Hübner 2002b], OperA [Dignum 2010], AMELI [Esteva 2004], MOISE Inst [Gâteau 2007]). Les normes ne sont pas alors codées dans les agents, mais dans des composants du système qui gèrent les normes au regard des rôles que jouent les agents dans le système, le plus souvent par régimentation. La régimentation étant dépendante du rôle joué par l'agent, on gagne en souplesse et certaines approches permettent la définition de systèmes ouverts [Kitio 2007]. La troisième technique consiste à

se reposer uniquement sur les agents pour le contrôle. C'est ce qu'on appelle le contrôle social, au travers de s-sanctions. Les agents s'appuient sur les indicateurs que sont la confiance et la réputation pour exclure ceux qui violeraient les normes. Cette dernière technique permet l'ouverture et une totale décentralisation.

A.3 Le Modèle PrivaCIAS

A.3.1 Introduction au Modèle PrivaCIAS

Le problème auquel cette thèse s'attaque est « comment limiter et prévenir les violations de privacité dans un système ouvert et décentralisé ? ». Nous voulons prévenir les violations dans un système :

- qui est décentralisé,
- qui est ouvert,
- qui ne permet pas un contrôle intrusif,
- qui gère les retransmissions,
- qui assiste les utilisateurs.

Dans les chapitres suivants, nous définissons un modèle de protection de la privacité dans les systèmes ouverts et décentralisés. Ce modèle est nommé PrivaCIAS : Privacité par l'Intégrité Contextuelle pour les Systèmes Agents.

Ce modèle est basé sur la mise en place d'agents assistants et est donc totalement décentralisé au travers de ces agents. Le modèle repose sur deux points pour protéger la privacité :

- il aide à prévenir les violations lors de l'envoi de messages : Lorsqu'un utilisateur envoie un message, son agent assistant va vérifier si ce message est conforme à l'intégrité contextuelle, de sorte à ce que cet utilisateur ne commette pas de violations. L'utilisateur gardera cependant le dernier mot, car le système est non-intrusif, et pourra décider d'envoyer un message occasionnant une violation, il s'exposera alors à de possibles sanctions.

- il détecte les violations à la réception de messages : Quand l'utilisateur reçoit un message, son agent assistant va vérifier s'il respecte l'intégrité contextuelle. Il avertira l'utilisateur en cas de violations et des sanctions appropriées seront prises envers le contrevenant. L'agent assistant diffusera le nom de l'agent ayant commis la violation, ce qui nuira à sa réputation, mettant en place une s-sanction.

Les chapitres suivants répondent aux deux questions principales de la thèse :

- Comment détecter une violation ?
  - Nous adaptons la théorie de l'intégrité contextuelle pour détecter les violations.
  - Nous fournissons aux agents un langage et des composants qui leurs permettent d'interagir.
- Comment obtenir un système ouvert et décentralisé cohérent dans la protection de la privacité ?
  - Nous définissons des normes afin d'instruire aux agents comment réagir aux violations.
  - Nous fournissons des agents compatibles avec PrivaCIAS.

A.3.2 Lois de convenance et langage de communication

Afin de détecter une violation de privacité, nous adaptons la théorie de l'intégrité contextuelle présentée précédemment afin de pouvoir déterminer si une communication est convenable ou non. Un message est convenable si les trois conditions suivantes sont respectées :

- Le contexte de la transmission correspond à la nature de l'information.
- Les agents ont un rôle adapté au contexte de la transmission.
- Les préférences de la cible de l'information sont respectées.

Pour détecter une violation de privacité, les agents doivent pouvoir :

- déterminer le contexte d'une transmission,

- savoir quels rôles sont joués par quels agents,
- se comprendre les uns les autres,
- pouvoir exprimer et comprendre des préférences,
- déterminer qui sont les cibles de l'information.

Rôles et Contextes Pour des raisons de simplification, nous considérons qu'il est possible d'unifier rôles et contextes en *rôle-contextes*. Tout agent peut décider de créer un rôle-contexte et d'y inscrire des participants. Un rôle-contexte est toujours composé d'un nom et de l'adresse de l'agent référent. Par exemple, `conseilDesMinistres@PremierMinistre` définit un rôle-contexte `conseilDesMinistres` qui est maintenu par l'agent `PremierMinistre`. Les agents du système pourront demander à l'agent `PremierMinistre` si un agent donné participe dans le rôle-contexte `conseilDesMinistres`.

Messages Nous définissons également une structure de message nous permettant d'apporter les éléments nécessaires pour qu'un agent puisse détecter une violation de privacité. Un message encapsule de l'information, dont la forme est libre, et des méta-informations, définies ci-après.
Un message contient les méta-informations suivantes :

- Des préférences concernant la diffusion du message, signées électroniquement par leur auteur.
- Une chaine de transmission composée par des maillons contenant :
  - le nom de l'expéditeur
  - le nom du destinataire
  - le rôle-contexte déclaré pour cette transmission
  - la cible déclarée pour cette transmission

  Le maillon est signé par son auteur afin d'éviter qu'il puisse être falsifié, la signature contient le nom du signataire, ainsi qu'une référence unique à l'information (sous forme de code de hachage).

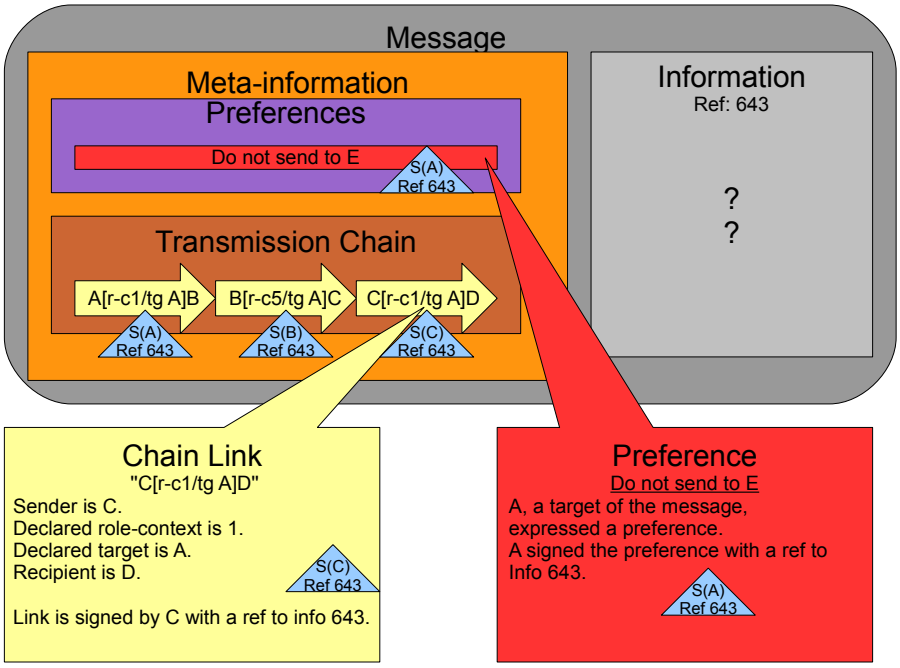


FIGURE A.1 – Message structure

La figure A.1 récapitule ces différents éléments. Il est à noter qu'il est impossible d'empêcher que les méta-informations soient détachées de l'information. Néanmoins, le système de signatures que nous mettons en place empêche que des méta-informations ne soient rattachées à une autre information (car les signatures contiennent une référence unique à l'information à laquelle elles s'appliquent).

Le langage de PrivaCIAS Nous définissons ici un langage qui permet aux agents de communiquer et de manipuler des concepts relatifs à la privacité. Ce langage est ensuite réutilisé pour définir formellement la violation de privacité. Le langage permet notamment d'accéder aux différentes méta-informations d'un message. Le lecteur pourra trouver l'ensemble des prédicats dans le chapitre . Ce langage est également utilisé pour définir des préférences, par exemple une préférence exprimant que « le destinataire doit être soit agentA soit agentZ sauf

s'ils sont en relation avec agentX » est la suivante :

```
mandatory(M):-
     recipient(agentA,M).

mandatory(M):-
     recipient(agentZ,M).

forbidden(M):-
     recipient(V,M),
     link(V,agentX).
```

Les lois de convenance Nous définissons un ensemble de lois qui définissent une transmission convenable, ou appropriée (Appropriateness-laws ou A-laws en anglais). Ces lois sont exprimées en utilisant les prédicats présentés ci-avant : A-Law 1 :

```
fitContext(+M):-
     information(M,I),
     declaredContext(C,M),
     context(C,I).
```

La loi `fitContext(+M)` déclare que si I est l'information contenue dans le message M, et que le rôle-contexte est C, la transmission est convenable si l'agent croit que C est compatible avec I.

A-Law 2 :

```
fitRole(+M):-
     recipient(R,M),
     declaredContext(C,M),
     roleContext(R,C).
```

`fitRole(+M)` est vraie si R est le destinataire du message M et si l'agent croit que R participe au rôle-contexte C.

A-Law 3 :

```
fitPreferences(+M):-
     information(M,I),
```

```
     declaredTarget(A,M),
     agentTarget(A,I),
     preference(P,A,M),
    satisfiesPreference(M,P).

fitPreferences(+M):-
     information(M,I),
     declaredTarget(A,M),
     agentTarget(A,I),
     not preference(P,A,M).
```

Les troisième et quatrième lois sont valides lorsque A est la cible déclarée pour le message M, l'agent croit que A est une cible légitime pour l'information I contenue dans M, et s'il y a une préférence P définie par A (troisième loi) elle est respectée. `fitPreferences(+M)` est également valide si aucune préférence n'est définie par A (quatrième loi).

Les lois de convenances doivent êtres satisfaites toutes les trois pour que le message soit convenable/approprié :

```
appropriate(+M):-
     fitContext(M),
     fitRole(M),
     fitPolicy(M).
```

A.3.3 Normes de support de la Privacité

Avec les éléments définis dans la section précédente, les agents peuvent communiquer entre eux et détecter des violations de privacité. Néanmoins, il manque une stratégie cohérente pour la préservation de la privacité dans le système. Cette section définit un ensemble de normes, les normes de support de la Privacité (Privacy Enforcing Norms ou PENs en anglais) qui définissent le comportement attendu des agents participants au système.

Les normes protègent la privacité des agents sous trois angles :

- prévenir les violations de privacité,
- stopper les violations des privacité,

- punir les violations de privacité.

Prévenir La meilleure façon de prévenir les violation est de contrôler les messages avant de les envoyer. Ainsi, la norme 1, ou PEN1 indique :

- PEN1 : Ne pas envoyer de messages sujets à une violation de privacité.

Quand l'agent envoie un message, il doit donc vérifier que celui-ci soit convenable au regard des A-laws.

Un autre problème doit être prévenu lors de l'envoie de messages : il ne faut pas transmettre de message à quelqu'un qui n'est pas digne de confiance.

- PEN2 : Ne pas envoyer de messages à des agents indignes de confiance.

Cette norme participera également à l'exclusion sociale des agents contrevenants. Ces derniers deviendront indignes de confiance et ne recevront plus de messages.

Stopper Des violations pourront avoir lieu dans le système si certains agents sont incompétents ou malveillants. Il faudra alors être capable de détecter ces violations et d'éviter qu'elles ne se propagent.

- PEN3 :Détecter et supprimer tout message sujet à une violation de privacité.

Lorsque l'agent reçoit un message, il devra vérifier qu'il soit convenable et dans le cas contraire le supprimer.

Afin d'éviter de propager des violations précédentes, si un agent reçoit un message provenant d'un agent indigne de confiance, il devra le supprimer.

- PEN4 : Supprimer les messages provenant d'agents indignes de confiance.

La norme 4 participe à l'exclusion sociale des contrevenants. Ainsi, avec les normes 2 et 4, des agents indignes de confiance seront exclus du système : personne ne leur enverra de message, et tout le monde supprimera les messages qu'ils envoient.

Punir Afin de garantir l'identification et la sanction des contrevenants, deux normes sont ajoutées.

La cinquième norme définit que chaque agent doit ajouter un maillon à la chaine de transmission du message. Cela permet d'identifier les intervenants de manière sure et de tracer le message.

- PEN5 : Un maillon doit être ajouté à la chaine de transmission par l'expéditeur.

Enfin, une norme clé est ajoutée afin de recommander aux agents de considérer comme indignes de confiance les agents qui ne respectent pas les normes.

- PEN6 : Punir les agents qui violent les PENs.

Agents de support de la privacité Nous proposons des agents qui suivent les PENs, appelés PEA (Privacy Enforcing Agents). Tous les agents dans un système PrivaCIAS sont supposés suivre les PENs, même si les normes prennent justement en compte l'éventualité où ce ne serait pas le cas. Nous proposons un modèle d'agent parmi les innombrables possibilités, en décrivant au travers de notre langage les actions à effectuer à la réception et à l'envoi de messages.

Lors de la réception d'un message, les PENs seront vérifiées par les formules suivantes :

```
receiving(M):-
     enforcePENsRecv(M),
     enforcePEN6(M).

enforcePENsRecv(M):-
     enforcePEN3(M),
     enforcePEN4(M),
     enforcePEN5(M),
     sender(X,M),
     congrat(X).

enforcePENsRecv(M):-
     delete(M),
```

```
     sender(X,M),
     punish(X).

receiving(M):-
     delete(M).
```

Les PENs seront satisfaites à la réception (`enforcePENsRecv` soit si les PENs 3, 4, 5 sont satisfaites, soit si le message est supprimé et son expéditeur punit (application de la PEN6).

```
enforcePEN3(M):-
     appropriate(M).

enforcePEN4(M):-
     sender(X,M),
     trustworthy(X).


enforcePEN5(M):-
     sender(X,M),
     recipient(Y,M),
     information(M,I),
     signed(X,Y,I,M).
```

La PEN3 est valide si le message respecte les A-laws. La PEN4 est satisfaite si l'expéditeur est digne de confiance. La PEN5 vérifie que le message ait bien été signé correctement (un maillon a bien été ajouté à la chaine). Notons qu'à la réception, les PENs 1 et 2 ne sont pas vérifiées car supposées toujours vraies (l'agent a confiance en lui même notamment).

```
enforcePEN6(M):-
     previousMessage(Mp,M),
     not signed(_,_,_,Mp).

enforcePEN6(M):-
     previousMessage(Mp,M),
```

```
     enforcePENsSend(Mp),
     enforcePENsRecv(Mp) .
```

La PEN6 permettra de vérifier que les PENs ont bien été respectées au cours de la précédente transmission par l'expéditeur et le destinataire.

Lors de l'envoi d'un message, un processus similaire sera effectué :

```
sending(M):-
     enforcePENsSend(M).

enforcePENsSend(M):-
     enforcePEN1(M),
     enforcePEN2(M),
     enforcePEN5(M).


enforcePENsSend(M):-
     sender(X,M),
     X \= me,
     punish(X).
```

Si l'agent réussi a satisfaire les PENs 1, 2 et 5, alors le message sera envoyé. Autrement, l'agent assistant devra prévenir son utilisateur.

```
enforcePEN1(M):-
     appropriate(M).

enforcePEN2(M):-
     recipient(X,M),
     trustworthy(X).
```

La PEN1 sera satisfaite si le message respecte les A-Laws. La PEN2 est vraie lorsque le destinataire est digne de confiance.

A.4 Validation et expériences

A.4.1 Test de résistance sous RePast

Afin de tester les performances du modèle, une simulation a été développée sous la plateforme RePast. Le simulateur génère un réseau composé d'agents dont certains sont des PEA, d'autres non. Pour faire simple, certains PEA sont des experts (PEAgentExpert) et peuvent vérifier les A-laws de manière parfaite. Les autres, les PEAgent normaux on une probabilité de réussite de 0.2 et doivent se reposer sur la chaine de transmission pour affiner leurs calculs le reste du temps. D'autres agents transmettent toute information reçue, sans préoccupation aucune (SimpleAgent). Enfin un dernier type d'agent ne transmet aucune information (NoTransAgent). Ce dernier type sert d'agent témoin.

Une des difficultés a été de trouver une métrique pertinente et représentative du niveau de protection de la privacité au sein du système. Par exemple, si l'on compte seulement le nombre de violations, les NoTransAgent seront les plus performants : si on ne transmet aucune information, alors on ne risque pas de faire de violation. La métrique choisie permet donc d'équilibrer la capacité des agents à communiquer et leur tendance à effectuer des violations.

La métrique principale est calculée individuellement pour chaque agent, puis rassemblée en une moyenne pour donner un score à chaque type d'agent et au système. Elle est calculée de la manière suivante :

- Les agents commencent avec un montant initial de 500 points.
- Un agent qui effectue une violation en envoyant un message perd 10 points.
- Un agent qui accepte en réception un message occasionnant une violation perd 5 points.
- Un agent qui envoie un message qui n'occasionne pas de violation gagne 4 points.
- Un agent qui accepte un message qui n'occasionne pas de violation gagne 2 points.

Un agent peut refuser un message de manière à éviter une violation, dans ce cas, personne ne perd ni ne gagne de points.

Nous présentons ici les scores obtenus dans différent cas de figure.

Dans le cas ou tous les agents sont des SimpleAgents, ils transmettent alors aveuglément tout message, on obtient les scores suivants :

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
-3898	-3898	n/a	n/a	n/a

Dans le cas où tous les agents sont des NoTransAgents, aucun message n'est envoyé, les agents conservent donc leur score initial :

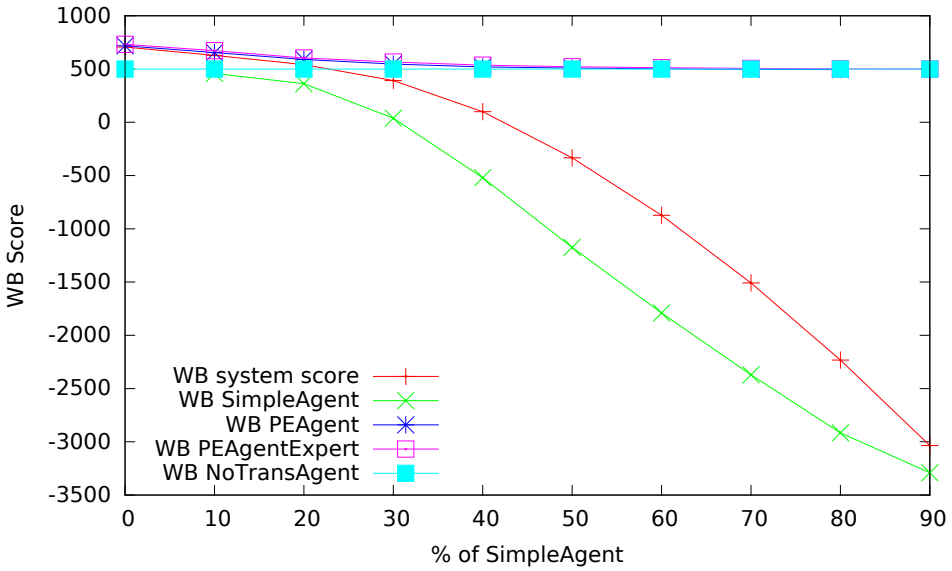
System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
500	n/a	n/a	n/a	500

Dans le cas où il n'y a que des PEAgentExpert, le score atteint le maximum que l'on puisse espérer réaliser :

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
1548	n/a	n/a	1548	n/a

Nous lançons ensuite une expérience afin de déterminer les limites du système : de multiples runs sont effectués avec 180 PEAgent, 20 SimpleAgent, 10 PEAgentExperts et 10 NoTransAgents. À chaque run, 20 PEAgents sont remplacés par des SimpleAgents.

La courbe suivante est obtenue :



On peut remarquer que le score du système commence à chuter autour de 30 pour cent de SimpleAgents. Nous choisissons alors ce pourcentage comme référence pour les expériences suivantes.

Le score de référence obtenu avec 30 pour cent de SimpleAgents est le suivant :

System WB Score	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
396	45	550	570	500

Le second jeu d'expériences consiste à retirer une par une chacune des PENs afin de constater leur impact sur le score du système.

Scenario	System	SimpleAgent	PEAgent	PEAgentExpert	NoTransAgent
Référence	396	45	550	570	500
no PEN1	299	37	406	382	500
no PEN2	311	-275	565	592	500
no PEN3	-926	-2625	-326	639	500
no PEN4	95	-624	392	446	500
no PEN5	396	45	549	569	500
no PEN6	-477	-2043	155	333	500

Comme on peut le constater sur le tableau, toutes les PENs, excepté la PEN5 on un impact positif sur le score du système (elles font baisser le score quand elles sont désactivées). La PEN5 n'a pas d'effet du fait de l'implémentation de PrivaCIAS sur la plateforme de simulation : l'implémentation de la PEN3 vérifie déjà que le message soit signé, la PEN5 est alors sans effet sur cette implémentation.

A.4.2 Application de démonstration sous Android

Andrei Ciortea à été mis en charge de l'implémentation d'un démonstrateur pour PrivaCIAS sur téléphones Android lors de son stage Master. L'idée était d'avoir une application réelle du modèle PrivaCIAS, que des utilisateurs puissent tester, ce qui n'était pas réaliste avec la plateforme de simulation présentée au chapitre précédent.

Les réseaux sociaux sont notre principal domaine d'application, c'est pourquoi nous avons choisi de développer un réseau social de partage de photos. Nous avons développé l'application sur smartphones Android, ces derniers peuvent être vus comme des assistants personnels. Ainsi, ils sont un support adapté pour

déployer des agents assistants personnels, comme nos PEA (Agents de support de la privacité).

Les utilisateurs qui téléchargent l'application peuvent s'enregistrer, puis inviter des amis. Un utilisateur qui accepte cette invitation créera un lien de confiance réciproque entre les deux utilisateurs. Leurs agents assistants respectifs considèreront donc qu'ils ont confiance l'un envers l'autre.

Les utilisateurs peuvent également créer des rôle-contextes et inviter des utilisateurs à les joindre.

Ils pourront ensuite s'échanger des photos prises par leur téléphone. Leur agent assistant, contenu dans l'application PrivaCIAS vérifiera à l'envoi si le message respecte les PENs. S'il détecte une violation, il en avertira son utilisateur. À la réception, l'agent assistant vérifiera également les PENs, et avertira les autres agents assistants en cas de violation. Quand aucune violation n'est détectée, l'utilisateur a la possibilité de marquer une image comme étant une violation, notamment si le contexte déclaré ne correspond pas à l'image présentée. Dans ce cas, l'agent assistant contactera les autres agents afin de les informer de la violation.

Au bout d'un certain nombre de violations, les agents briseront les liens de confiance et empêcheront ainsi l'envoi de messages vers ces agents ainsi que la réception de messages provenant d'eux, conformément aux PENs.

Des illustrations de l'application PrivaCIAS Android sont présentées en section 8.2.3.

A.5 Conclusion

Cette thèse est, à notre connaissance, la première à s'attaquer au problème de privacité dans les systèmes ouverts et décentralisés en utilisant des techniques socio-inspirées comme l'intégrité contextuelle, la confiance et la réputation.

Afin de mettre en place un contrôle social basé sur la confiance, nous avons été obligés de définir ce qu'est une trahison. La théorie de l'intégrité contextuelle a été utilisée afin de définir le concept de violation de privacité. Des normes (PENs) ont ensuite été définies afin d'indiquer aux agents qu'une violation de privacité doit être considéré comme une trahison, et que les traitres doivent être exclus du système.

Le modèle essaye d'être le moins restrictif possible, de manière à permettre l'intégration de PrivaCIAS sur des applications très diverses.

Le modèle n'est pas contraignant pour le développeur. Ainsi, tout développeur pourra proposer son agent PEA, tant que celui-ci respecte les PENs. Si ce n'est pas le cas, son agent sera exclu par les autres comme les expériences l'ont montré.

La prochaine étape principale dans le développement de PrivaCIAS sera d'inclure des normes spécifiques aux contextes, en reprenant l'idée de Nissenbaum. Des préférences par défaut pourront ainsi être définies pour chaque rôle-contexte par l'agent en charge du-dit contexte. Ces préférences s'appliqueront à tout message échangé dans le rôle-contexte en question.

Publications

2012 A. Ciortea and Y. Krupa and L. Vercouter, *Designing privacy-aware social networks: a multi-agent approach.* Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics.

2012 Y. Krupa and L. Vercouter, *Handling privacy as contextual integrity in decentralized virtual communities: The PrivaCIAS framework.* Web Intelligence and Agent Systems Journal.

2011 Y. Krupa and L. Vercouter, *Contrôle social de la privacité selon l'intégrité contextuelle dans les systèmes décentralisés.* Journées Francophones sur les Systèmes Multi-Agents.

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

2010 Y. Krupa and J. Sabater-Mir and L. Vercouter, *Handling Privacy as Contextual Integrity in Virtual Communities.* WIVE 2010: 2nd International Workshop on Web Intelligence and Virtual Enterprises.

2009 Y. Krupa and L. Vercouter and J. F. Hubner and A. Herzig, *Trust based evaluation of wikipedia's contributors.* Engineering Societies in the Agents World X.

2009 Y. Krupa, J. F. Hubner and L. Vercouter, *Extending the Comparison Efficiency of the ART Testbed.* Proceedings of the First International Conference on Reputation: Theory and Technology - ICORE 09.

Bibliography

[Abril 2010] D. Abril, G. Navarro-Arribas and V. Torra. *Towards semantic microaggregation of categorical data for confidential documents*. Modeling Decisions for Artificial Intelligence, pages 266–276, 2010. (Cited on page 35.)

[Acquisti 2006] A. Acquisti and R. Gross. *Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook*. Lecture Notes in Computer Science, vol. 4258, page 36, 2006. (Cited on pages 17 and 37.)

[Agrawal 2002] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu. *Hippocratic databases*. In Proceedings of the 28th international conference on Very Large Data Bases, pages 143–154. VLDB Endowment, 2002. (Cited on pages 40 and 167.)

[Albert 2000] R. Albert and A.L. Barabási. *Topology of evolving networks: local events and universality*. Physical review letters, vol. 85, no. 24, pages 5234–5237, 2000. (Cited on page 123.)

[Aldewereld 2007] H. M. Aldewereld. *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*. PhD thesis, Universiteit Utrecht, 2007. (Cited on page 62.)

[Anderman 2003] G.M. Anderman and M. Rogers. Translation today: trends and perspectives. Multilingual Matters Ltd, 2003. (Cited on page 10.)

[Antón 2003] A.I. Antón, Q. He and D.L. Baumer. *The Complexity Underlying JetBlue's Privacy Policy Violations*. Complexity, 2003. (Cited on page 37.)

[Baase 2007] S. Baase. Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet. Prentice Hall, 2007. (Cited on pages 9, 10 and 164.)

[Barabási 1999] A.L. Barabási and R. Albert. *Emergence of scaling in random networks*. science, vol. 286, no. 5439, pages 509–512, 1999. (Cited on page 123.)

[Barth 2006] A. Barth, A. Datta, J.C. Mitchell and H. Nissenbaum. *Privacy and Contextual Integrity: Framework and Applications*. 2006 IEEE Symposium on Security and Privacy (S&P'06), pages 184–198, 2006. (Cited on page 22.)

[Bell 1973] D. E. Bell and L. J. LaPadula. *Secure computer systems: Mathematical foundations*. Rapport technique, Technical Report MTR-2547, 1973. (Cited on page 32.)

[Bradwell 2007] P. Bradwell and N. Gallagher. *The new politics of personal information*. Demos report. London: Julie Pickard, 2007. (Cited on page 16.)

[Byun 2005] J.W. Byun, E. Bertino and N. Li. *Purpose based access control of complex data for privacy protection*. In Proceedings of the tenth ACM symposium on Access control models and technologies, page 110. ACM, 2005. (Cited on page 33.)

[Ciortea 2011] A. Ciortea. Modeling Relashionships for privacy Preservation in Virtual Communities. Master's thesis, Universitatea Politehnica din Bucuresti, 2011. (Cited on pages 15, 149 and 161.)

[Collier 2003] N. Collier. *Repast: An extensible framework for agent simulation*. The University of Chicago's Social Science Research, vol. 36, no. February 11, 2003. (Cited on page 121.)

[Crépin 2009] L. Crépin. *Les Systèmes Multi-Agents Hippocratiques*. PhD thesis, Université Jean Monnet, 2009. (Cited on pages 41, 42 and 167.)

[Delgado 2003] J. Delgado, J.M. Pujol and R. Sanguesa. *Emergence of coordination in scale-free networks*. Web Intelligence and Agent Systems, vol. 1, no. 2, pages 131–138, 2003. (Cited on page 60.)

[Dignum 2010] V. Dignum and H. Aldewereld. *OperettA: Organization-oriented development environment*. In Proceedings of the 3rd International workshop on Languages, Methodologies and Development Tools for Multi-agent Systems (LADS2010@ Mallow), 2010. (Cited on pages 62 and 169.)

[Esteva 2004] M. Esteva, B. Rosell, J.A. Rodriguez-Aguilar and J.L. Arcos. *Ameli: An agent-based middleware for electronic institutions.* In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1, pages 236–243. IEEE Computer Society, 2004. (Cited on pages 62 and 169.)

[Falcone 2001] R. Falcone and C. Castelfranchi. *Social trust: a cognitive approach.* Trust and deception in virtual societies table of contents, pages 55–90, 2001. (Cited on page 64.)

[Ferraiolo 1995] D. Ferraiolo, J. Cugini and D.R. Kuhn. *Role-based access control (RBAC): Features and motivations.* In Proceedings of 11th Annual Computer Security Application Conference, pages 241–48. IEEE Computer Society Press, 1995. (Cited on pages 33 and 166.)

[Fogel 2009] J. Fogel and E. Nehmad. *Internet social network communities: Risk taking, trust, and privacy concerns.* Computers in Human Behavior, vol. 25, no. 1, pages 153–160, 2009. (Cited on page 16.)

[Gâteau 2007] B. Gâteau, O. Boissier, D. Khadraoui and E. Dubois. *Controlling an interactive game with a multi-agent based normative organisational model.* Coordination, Organizations, Institutions, and Norms in Agent Systems II, pages 86–100, 2007. (Cited on pages 2, 62 and 169.)

[Grandison 2003] T. Grandison and M. Sloman. *Trust management tools for internet applications.* Trust Management, pages 1071–1071, 2003. (Cited on page 63.)

[Grimmelmann 2008] J. Grimmelmann. *Facebook and the social dynamics of privacy.* Iowa Law Review, vol. 95, no. 4, pages 1–52, 2008. (Cited on pages 12, 16, 17, 44 and 165.)

[Grizard 2007] A. Grizard, L. Vercouter, T. Stratulat and G. Muller. *A peer-to-peer normative system to achieve social order.* Coordination, organizations, institutions, and norms in agent systems II, pages 274–289, 2007. (Cited on pages 2 and 63.)

[Hubner 2002a] J.F. Hubner, J. S. Sichman and O. Boissier. *A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems*. Lecture Notes in Computer Science, pages 118–128, 2002. (Cited on page 2.)

[Hübner 2002b] J.F. Hübner, J.S. Sichman and O. Boissier. *Moise+: towards a structural, functional, and deontic model for mas organization*. In Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, pages 501–502. ACM, 2002. (Cited on pages 62 and 169.)

[Hübner 2006] J. Hübner, J. Sichman and O. Boissier. *S-Moise+: A Middleware for Developing Organised Multi-agent Systems*. Coordination, organizations, institutions, and norms in multi-agent systems, pages 64–77, 2006. (Cited on page 62.)

[Hubner 2007] J.F. Hubner, J. S. Sichman and O. Boissier. *Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels*. International Journal of Agent-Oriented Software Engineering, vol. 1, no. 3, pages 370–395, 2007. (Cited on page 2.)

[Hübner 2009] J. F. Hübner, E. Lorini, A. Herzig and L. Vercouter. *From cognitive trust theories to computational trust*. In Proc. of the Twelfth Workshop" Trust in Agent Societies" at AAMAS, volume 9, pages 55–67, 2009. (Cited on page 64.)

[Hübner 2010] J.F. Hübner, O. Boissier, R. Kitio and A. Ricci. *Instrumenting multi-agent organisations with organisational artifacts and agents*. Autonomous Agents and Multi-Agent Systems, vol. 20, no. 3, pages 369–400, 2010. (Cited on page 2.)

[Jernigan 2009] C. Jernigan and B.F.T. Mistree. *Gaydar: Facebook friendships expose sexual orientation*. First Monday, vol. 14, no. 10, 2009. (Cited on page 15.)

[Kitio 2007] R. Kitio, O. Boissier, J.F. Hübner and A. Ricci. *Organisational artifacts and agents for open multi-agent organisations: giving the power back to the agents*. In Proceedings of the 2007 international conference on

Coordination, organizations, institutions, and norms in agent systems III, pages 171–186. Springer-Verlag, 2007. (Cited on pages 62 and 169.)

[Korba 2002] L. Korba, S. Kenny*et al.* *Towards meeting the privacy challenge: Adapting drm.* In 2002 ACM Workshop on Digital Rights Management, Held in Conjunction with the Ninth ACM Conference on Computer and Communications Security, 2002. (Cited on page 43.)

[Krupa 2009] Y. Krupa, L. Vercouter, J. Hübner and A. Herzig. *Trust based evaluation of wikipedia's contributors.* Engineering Societies in the Agents World X, pages 148–161, 2009. (Cited on pages 65 and 66.)

[Lorini 2008] E. Lorini and R. Demolombe. *Trust and norms in the context of computer security: A logical formalization.* Deontic Logic in Computer Science, pages 50–64, 2008. (Cited on page 65.)

[Mont 2003] M. C. Mont, S. Pearson and P. Bramhall. *Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services.* In Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on, pages 377–382, 2003. (Cited on pages 46 and 168.)

[Nardin 2008] L. Nardin, A. Brandao, J. Sichman and L. Vercouter. *An Ontology Mapping Service to Support Agent Reputation Models Interoperability.* À paraitre dans : Workshop Trust in agent societies, AAMAS 2008, 2008. (Cited on page 2.)

[Nissenbaum 2004] H. Nissenbaum. *Privacy as Contextual Integrity.* Washington Law Review, pages 101–139, 2004. (Cited on pages 10, 12, 18, 19, 20, 74, 159, 161, 164 and 165.)

[Nissenbaum 2010] H. Nissenbaum. Privacy in context: Technology, policy, and the integrity of social life. Stanford Law & Politics, 2010. (Cited on pages 18, 19, 20, 21 and 161.)

[Orwell 2006] G. Orwell. 1984. Editions Underbahn Ltd., 2006. (Cited on page 12.)

[Piolle 2009] G. Piolle. *Agents utilisateurs pour la protection des données personnelles: modélisation logique et outils informatiques.* PhD thesis, Université Joseph Fourier, 2009. (Cited on pages 47 and 168.)

[Popa 2009] R.A. Popa, H. Balakrishnan and A.J. Blumberg. *VPriv: Protecting privacy in location-based vehicular services.* In Proceedings of the 18th conference on USENIX security symposium, pages 335–350. USENIX Association, 2009. (Cited on page 36.)

[Reagle 1999] J. Reagle and L. F. Cranor. *The platform for privacy preferences.* ACM, vol. 42, no. 2, pages 48–55, 1999. (Cited on pages 37 and 167.)

[Rivest 1978] R. L. Rivest, A. Shamir and L. Adleman. *A Method for Obtaining Digital Signatures and Public- Key Cryptosystems.* Communications, vol. 21, no. 2, 1978. (Cited on pages 31 and 87.)

[Rivest 1992] R. Rivest. *The MD5 Message-Digest Algorithm.* Distribution, 1992. (Cited on page 87.)

[Sabater 2005] J. Sabater and C. Sierra. *Review on computational trust and reputation models.* Artificial Intelligence Review, vol. 24, no. 1, pages 33–60, 2005. (Cited on page 64.)

[Sabater 2006] J. Sabater, M. Paolucci and R. Conte. *Repage: REPutation and ImAGE Among Limited Autonomous Partners.* Journal of Artificial Societies and Social Simulation, vol. 9, no. 2, page 3, 2006. (Cited on page 66.)

[Saunier 2009] J. Saunier and F. Balbo. *Regulated multi-party communications and context awareness through the environment.* Multiagent and Grid Systems, vol. 5, no. 1, pages 75–91, 2009. (Cited on page 61.)

[Savarimuthu 2009] B.T.R. Savarimuthu, S. Cranefield, G. Boella, P. Noriega, G. Pigozzi and H. Verhagen. *A categorization of simulation works on norms.* Normative Multi-Agent Systems, 2009. (Cited on page 59.)

[Schoeman 1994] F. Schoeman. *Gossip and privacy.* Good Gossip, pages 403–408, 1994. (Cited on pages 20 and 165.)

[Solove 2007] D. Solove. *I've Got Nothing to Hide and Other Misunderstandings of Privacy.* San Diego L. Rev., vol. 44, page 745, 2007. (Cited on pages 9, 11 and 164.)

[Solove 2008] D. Solove. *Understanding privacy.* Daniel J. Solove, UNDERSTANDING PRIVACY, Harvard University Press, May 2008, GWU Legal Studies Research Paper No. 420, GWU Law School Public Law Research Paper No. 420, 2008. (Cited on page 18.)

[Sweeney 2002] L. Sweeney. *k-anonymity: A model for protecting privacy.* International Journal on Uncertainty Fuzziness and Knowledgebased Systems, vol. 10, no. 5, pages 557–570, 2002. (Cited on page 35.)

[Testard-Vaillant 2008] P. Testard-Vaillant. *Identification, vie privée, sécurité.* Le Journal du CNRS, vol. 225, page 19, 2008. (Cited on page 18.)

[Tuomela 1995] R. Tuomela. The importance of us: A philosophical study of basic social notions. Stanford University Press Stanford, CA, 1995. (Cited on pages 58 and 169.)

[Tuomela 2000] R. Tuomela. Cooperation: A philosophical study, volume 82. Springer, 2000. (Cited on page 58.)

[Vercouter 2010] Laurent Vercouter and Guillaume Muller. *L.I.A.R.: Achieving Social Control in Open and Decentralized Multiagent Systems.* Applied Artificial Intelligence, vol. 24, pages 723–768, September 2010. (Cited on page 2.)

[Villatoro 2009] D. Villatoro, S. Sen and J. Sabater-Mir. *Topology and memory effect on convention emergence.* In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02, pages 233–240. IEEE Computer Society, 2009. (Cited on page 60.)

[Watts 1998] D.J. Watts and S.H. Strogatz. *Collective dynamics of 'small-world'networks.* nature, vol. 393, no. 6684, pages 440–442, 1998. (Cited on page 123.)

[Westin 2003] A.F. Westin. *Social and political dimensions of privacy.* Journal of Social Issues, vol. 59, no. 2, pages 431–453, 2003. (Cited on pages 9 and 164.)

[Yeung 2009] C.A. Yeung, I. Liccardi, K. Lu, O. Seneviratne and T. Berners-Lee. *Decentralization: The future of online social networking.* In W3C Workshop on the Future of Social Networking Position Papers, 2009. (Cited on page 52.)

[Zacharia 1999] G. Zacharia. Collaborative reputation mechanisms for online communities. Master's thesis, Massachusetts Institute of Technology, 1999. (Cited on page 66.)

[Zimmermann 1995] P.R. Zimmermann. The official pgp user's guide. The MIT Press, 1995. (Cited on pages 31 and 166.)

École Nationale Supérieure des Mines de Saint-Étienne

NNT : 2012 EMSE 0657

Yann Krupa

PrivaCIAS: Privacy as Contextual Integrity in Decentralized Multi-Agent Systems

Specialité : Informatique

Mots Clefs : privacité, intégrité contextuelle, réseaux sociaux, systèmes multi-agents, réseaux décentralisés

Résumé :

Les approches habituelles pour la protection de la privacité s'attachent à définir un niveau de sensibilité pour chaque information. Cette information est soit publique, soit privée et sa circulation est restreinte à un groupe d'agents prédéfini. Dans cette thèse, nous nous appuyons sur la théorie de l'intégrité contextuelle, qui propose de redéfinir la notion de violation de privacité. Selon cette théorie, toute transmission peut déclencher une violation de privacité suivant le contexte dans lequel elle a lieu. Cette thèse utilise la théorie de l'intégrité contextuelle afin de proposer un modèle de protection de la privacité pour les systèmes multi-agents décentralisés : le modèle PrivaCIAS. Afin de contrôler les agents dans le système, le modèle PrivaCIAS fournit un ensemble de normes qui permet la mise en place d'un contrôle social basé sur la confiance. Le modèle donne le contrôle aux agents pour constater les violations (selon l'intégrité contextuelle), puis punir les contrevenants en les excluant du système sans avoir besoin de recourir à une autorité centrale. Ce modèle vise les réseaux sociaux décentralisés comme champ d'application.

Version : 20120930223300

École Nationale Supérieure des Mines de Saint-Étienne

NNT : 2012 EMSE 0657

Yann Krupa

PrivaCIAS: Privacy as Contextual Integrity in Decentralized Multi-Agent Systems

Major in Computer Science

Keywords: Privacy, Contextual Integrity, Social Networks, Multi-agent Systems, Open Decentralized Networks

Abstract:

Contextual Integrity has been proposed to define privacy in an unusual way. Most approaches take into account a sensitivity level or a "privacy circle": the information is said to be either private or public and to be constrained to a given group of agents, *e.g.* "my friends", when private. In the opposite, Contextual Integrity states that any information transmitted can make this transmission a privacy violation depending on its context. In this thesis, we use this theory to develop a novel model that one can use in an open and decentralized virtual community to socially enforce privacy. This thesis defines the PrivaCIAS model, in which privacy constraints are formally described to be used to detect privacy violations according to the Contextual Integrity theory. The PrivaCIAS model provides norms to agents in order to make them implement social control. The model does not require a central authority, it gives control to the agents for detecting privacy violations (through Contextual Integrity) and excluding violating agents from the system through social exclusion. This model targets decentralized social networks as a main application domain.