Chapter 14

Security, Privacy and Trust

Federico Bergenti

14.1 Introduction

The literature about trust in multiagent systems collects a huge number of works that analyse almost any facets of this concept from nearly every point of view. Nevertheless, an accepted and stable formal model of trust in agent societies is still missing. In this chapter, we address this remarkable flaw of the current research by reporting the main contribution of the CASCOM project on this topic: (i) a stochastic model of trust that measurably captures trust in two-party interactions, and (ii) a general-purpose framework that the CASCOM platform provides to enable the realization of secure, privacy-aware and trust-aware multiagent systems.

Interaction is a key feature of agenthood (actually, "the" key feature, we may say) and secure, trusted and privacy-aware interactions are what we truly want from real-world multiagent systems. While it is easy to identify a minimum set of requirements capable of providing guarantees of security in multi-party interactions, e.g., authorization and authentication, we are not yet ready to identify similar requirements for trusted and privacy-aware interactions.

The work done in the CASCOM project on these problems is along the lines of the research that is trying to identify a set of abstractions and mechanisms to guarantee trust- and privacy-awareness in multi-agent interactions. In particular, the final objective of our work is about providing the CASCOM platforms with a set of facilities allow developers to easily and intuitively create not only secure but also privacy- and trust-aware multiagent systems. In order to achieve our goal, we developed a stochastic model of trust capable of formally showing that interactions that are mediated by a trusted third party, that we call *guarantor*, are rationally convenient over direct interactions. This result ensures that privacy- and trust-awareness can be obtained by mediated interactions and it provides a solid base for the design of the a framework for privacy and trust awareness that we integrated in the CASCOM platform.

This chapter is organized as follows: the next section frames the problem that we address in order to focus on the ideas and the abstractions behind our stochastic model. Section 14.3 provides the foundations of our model and quantifies the increment of the utility that agents perceive because of the mediation of a guarantor. Then, Section 14.3.3 deals with the decision-making strategies of rational agents and it shows a worst-case specialization of our model that justifies why agents are more likely to choose guarantor-mediated over direct interactions. Section 14.4 reports on how our stochastic model is concretized into the CASCOM platform by means of a framework that facilitates the realization of trust- and privacy-aware multiagent systems. This framework relies on secure messaging within the CASCOM platform and Section 14.4 also provides some technical details on this. Finally, Section 14.5 summarizes the lessons learnt from this work.

14.2 Two-Party Interactions

Most of the work reported in this chapter is about the study of the interaction between only two agents, X and Y. This study is very generic and its results can be applied in many situations. In any case, we need to focus our work on a special case of general interest in order to devise a formal framework for our study. This is the reason why we take the assumption that, from the point of view of security, trust and privacy, we can always reduce any two-party interaction to the special case of two agents mutually signing a contract. With no loss of generality, from now on we will always refer to the joint act of signing a contract as a means to study any other form of two-party interaction.

Having said this, we can state our working scenario as follows: X is interested in signing a contract with Y and it is in the process of deciding whether to do it directly or through the mediation of a trusted third party, the guarantor G, that can act as a middleman of this transaction. We take a rational standpoint and we assume that X discriminates between direct and mediated interaction on the basis of its utility function. Moreover, we assume incomplete information and we say that X cannot take a fully-informed decision. Rather, it has to face some risks.

This scenario models some interesting properties of real-world interactions and it provides a sufficiently simple case to allow for a formal analysis. Moreover, we believe that many interactions that are possibly occurring in nowadays multiagent systems can be approximated with acceptable accurancy to a network of two-party interactions. The comprehensive study of scenarios involving many jointly interacting agents is still work-in-progress and it is subject for a future work.

The two-party scenario that we use to define our stochastic model of trust relies on an underlying assumption that is worth some discussion. In particular, we always assume that agents exchange the terms of the contract under negotiation using individuals of a known and shared ontology, which is then described in some known and shared logic formalism, e.g., OWL [15]. This assumption allows agents

to manage the information contained in the contract in a friendly way and to reason about the contract with a reasonable accurancy.

All in all, the assumption of modelling contracts between negotiating agents in terms of individuals of known and shared ontologies is absolutely general and has some remarkable advantages. First, complex contracts can be described using a combination of simple — ontologies, with a potential reduction of the complexity of published ontologies. We can freely compose simple ontologies into complex descriptions of contracts, thus avoiding duplication of definitions and possible ambiguities. The second advantage that we see in using ontologies for modelling contracts is that it greatly simplifies the creation and validation of proposals and agreements. The creation of a proposal is reduced to the creation of one or more individuals of known ontologies. The control of the suitability of a proposal reduces to checking whether a candidate proposal actually belongs to the family of admissible proposals described in the referenced ontology. The problems of creation and validation of individuals of ontologies are both well-studied and they are largely supported by a number of available tools, e.g., reasoners [16] and query engines [11]. We need no special-purpose instrument to manage contracts: any available tool for processing ontologies are suitable for the purpose. Finally, ontologies expressed in common formats are easily mapped into human-readable documents for a subsequent inspection of the agreements that agents may have autonomously signed.

Nevertheless, the obvious assumption of using ontologies to describe the terms of a contract has some important drawbacks that we need to consider carefully. In fact, any attempt to use them in real-world scenarios immediately encounters a problem: How an agent could trust the constraint of a new ontology? Suppose that a seller requires possible customers to sign contracts using an ontology that it provides and that it made available in some public repository. This ontology may model some property as being "required by local laws." How could customers trust this requirement if they have no trust relationship with the seller that created this ontology? Could a customer — in some sense — validate the ontology to decide whether to trust it or not? Obviously, there is no way to validate the adherence of an ontology to real-world laws without involving highly specialized jurists. No potential customer would be in the position of performing this sort of validation.

Another facet of this problem occurs in the case of an ontology that is partially non-disclosed to final users. Let us suppose that the aforementioned seller splits its ontology into two parts: a public part describing valid proposals and agreements, and a private part used to model the policies that it employees to fix prices and accept orders, i.e., the policies that it uses to reason on proposals. This last part contains background knowledge on the marketing strategies of the seller and it is vital not to disclose it to potential competitors. In this case, a full fledged reasoning on the ontology could be done only by accessing the whole ontology, and only partial reasoning is possible for customers.

All in all, these exemplified facets of the same problem all roots in the re-

quirement that ontologies used to model contracts must be provided by trusted and liable signers.

Unfortunately, this last requirement is not sufficient to provide a solid base for modelling the trust relationships in real-world contracts. In fact, we need to take into account legal validity in a larger scope and therefore the problem of checking the identities of involved agents is obviously crucial. Unfortunately, a simple static control of identities by means of certificates [7, 8] is inadequate because, e.g., certificates can be revoked and keys can be stolen. We definitely need a dynamic approach to validating identities, i.e., the identification of agents in a secure, privacy- and trust-aware multiagent system can be performed only through a set of runtime services capable of validating certificates, and thus performing a trusted source of identification.

The problem of checking identities is closely related to the concrete representation of identities. For example, in Italy persons are uniquely identified by an alphanumeric code that groups the full name of the person, his/her birth date, his/her birthplace and a checksum. Similarly, corporations are designated with their VAT identification number. The identification code is the only means that we have to validate the identity of a legal person, whether physical or not. Therefore, one of the very basic issues that we have to tackle is how to represent identities in an agent-processable way. In our model, we decided to design an ontology describing legal persons and their attributes and to associate this ontology with a set of general-purpose services for addressing the majority of problems related to identification. The connection between this ontology and its services is reinforced by the necessity of a common trusted signer.

It is worth noting that, in order to fully exploit the possibility of having runtime services capable of providing some sort of guarantees regarding sensible tasks on an ontology, both the ontology and its associated services must have the same levels of trust and security. In fact, we have — at least — two interesting cases. In the first, suppose that two negotiating agents both trust the publisher of the ontology. They exchange proposals until an agreement is reached and they mutually check their identities using an untrusted service. Since they do not trust the identity-verification service, they can both suppose that they are signing an agreement with an unknown party. The symmetric case may also arise: suppose that we have an identity-verification service that receives an ontology and an identity as input, and that it verifies the identity in a secured database. What happens if someone gives formally valid — compliant with the ontology — but legally void identity? Since the given identity matches the record in the database of identities, the service would return an affirmative answer, but this identity is legally void and therefore unusable in signing contracts.

These two simple examples show that both ontology and associated services must be trusted and secured. If any of the two has not a suitable level of trust and security, their combined use will not result in a secure and trusted interaction.

14.3 A Model of Mediated Interactions

This section presents a set of abstractions and accounts for their relationships in order to setup a stochastic model of interactions between agent X, agent Y and (possibly) guarantor G. It is worth noting that this model is symmetrical for X and Y.

14.3.1 Abstractions

The problem of providing a quantitative definition of trust in societies of rational agents has been addressed in many different ways, e.g., see [14]. While we recognize the critical importance of cognitive models of trust, e.g., see [6], we date back to the abstract and coarse-grained definition of trust given in [9] to come to a stochastic interpretation of this notion. In particular, if we recall that:

> "Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends,"

it is quite reasonable to model trust as an estimation of the probability by which B will perform the target action. Many factors contribute to this estimation [12, 13]; nonetheless we prefer to adopt a blackbox approach that discards all these factors and we model trust as a random variable $\mathbf{t}$ that ranges in the interval $[t_{min}, t_{max}]$. Clearly t_{min} and t_{max} are both between zero and one and we assume $t_{max} \geq t_{min}$ with no loss of generality.

Then, we assume the rationality of agent A and we require that the estimation of the probability by which B will perform the target action is done using some reasonable amount of information regarding B and its actual intention of performing the action. This guarantees that the real probability of B performing the action lays in $[t_{min}, t_{max}]$, with t_{min} and t_{max} reasonably close around it.

Having said this, our model of two-party interactions is based on the following quantities, where X and Y are agents and c is a contract:

- $p_{c,X}$: the probability that X would carry out successfully all the obligations stated in c.
- $t_{c,X,Y}$: the measure of trust that X has in Y with respect to c, i.e., an estimation of $p_{c,Y}$ from the point of view of X.

The study of all different forms of contracts is the subject of a large literature and even restricting it to the types of contracts that we normally consider in multiagent systems [4], the diversity of possibilities is remarkable. We acknowledge this literature but, for the sake of simplicity and for the need of quantitative tractability, we stick on a very simple model of contract. This model involves only two signers, X and Y, and it is totally described by two triples — each signer knows only one of the two triples. In detail, from the point of view of agent X —

the notation is symmetrical for Y — a contract c is described by a triple, that we call *subjective evaluation*, that contains:

- A *reward* $R_{c,X}$ that agent X receives upon success of contract c;
- An *investment* $I_{c,X}$ that agent X makes in contract c, i.e., a certain assured value that it releases when signing contract c; and
- A *penalty* $P_{c,X}$ that agent X receives if the contract fails because of the other party.

Such values are not restricted to be monetary, rather they quantify the level of satisfaction of X. All in all, such quantities are subjective and therefore we cannot assess any mathematical relations between values of the triples of two different agents, even though they refer to the same contract.

More in details, a contract c has the following properties from the point of view of X:

- If the contract is honoured, agent X will receive $R_{c,X}$ with probability one; and
- If the contract fails because of agent Y, agent X will receive $P_{c,X}$ with probability one.

Another assumption concerns the relative ordering of reward, investment and penalty in a single subjective evaluation. We are interested in contracts whose parameters are ordered as follows:

$$P_{c,X} \leq I_{c,X} \leq R_{c,X} \tag{14.1}$$

This inequality captures the essence of risky contracts. Moreover, it implies that we are interested in agents that sign contracts with the intent of honouring them. Any failure in honouring a contract turns into a loss of utility (see later on): $I_{c,X} - P_{c,X}$. Furthermore, agents in our model do not consider their failure in honouring a contract, they just assume that they can honour all contracts they sign; nevertheless the uncertainty about the other signer remains.

The abstraction of *guarantor* was introduced and discussed in details in [3, 2]. For the sake of completeness, we recall here that guarantors are sources of highly trusted information and they are sort of trust catalysts, i.e., they are trusted by other agents and they form connecting nodes in the network of trust. If agent X requests a piece of information from guarantor G, it assigns a correctness probability of one to the received response. Nevertheless, we introduce some failure probability in order to account for the idea that the use of additional information, i.e., the information that guarantor G provides, always introduces some risk, even though the information source is highly trusted and reliable.

14.3.2 Expectation of the Utility of Agents

We now analyse the utility that agents estimate in the process of signing a contract. This utility is considered in two forms: with and without the mediation of a guarantor. We refer to the first case as *mediated* interaction, and we say that the second case is a *direct* interaction. We start with the formalization of direct interactions because their treatment is obviously simpler.

Direct Interaction

Taking into account the abstractions that we previously defined, we can explicitly write the expected value of the utility that agent X receives from signing a contract with agent Y as:

$$\overline{U}^{r}_{X,c} = R_{c,X} \cdot p_{c,Y} + P_{c,X} \cdot (1 - p_{c,Y}) \tag{14.2}$$

where the superscript "r" indicates that the real probability is used in this equation, and not an estimation of its value.

Unfortunately, this utility is not available to any agent since $p_{c,Y}$ is not observable. Instead, agent X estimates the expected utility using its trust in the other party (agent Y):

$$\overline{U}^{e}_{X,c} = R_{c,X} \cdot t_{c,X,Y} + P_{c,X} \cdot (1 - t_{c,X,Y}) \tag{14.3}$$

Taking into account that agent X invests a certain value when it signs the contract, and that any contract has some probability $p^{s}_{c,X}$ of being finally signed, the total average utility perceived by agent X perceives is:

$$\begin{aligned} \overline{U}^{r}_{X} &= \overline{U}^{r}_{X,c} \cdot p^{s}_{c,X} + I_{c,X} \cdot (1 - p^{s}_{c,X}) \\ &= [R_{c,X} \cdot p_{c,Y} + P_{c,X} \cdot (1 - p_{c,Y})] \cdot p^{s}_{c,X} + I_{c,X} \cdot (1 - p^{s}_{c,X}) \end{aligned} \tag{14.4}$$

As before, the agent can only estimate the total utility, obtaining:

$$\begin{aligned} \overline{U}^{e}_{X} &= \overline{U}^{e}_{X,c} \cdot p^{s}_{c,X} + I_{c,X} \cdot (1 - p^{s}_{c,X}) \\ &= [R_{c,X} \cdot t_{c,X,Y} + P_{c,X} \cdot (1 - t_{c,X,Y})] \cdot p^{s}_{c,X} + I_{c,X} \cdot (1 - p^{s}_{c,X}) \end{aligned} \tag{14.5}$$

Mediated Interaction

We can adapt the previous equations to the case in which the contract is evaluated using additional information obtained from a guarantor. In this case, the failure probability that we associate with a guarantor has to be considered. This failure probability accounts for the possible uncertainty of the information that the guarantor provides. In particular, we assume that an error of a guarantor may cause a failure of the contract. In this case agent X receives $P_{c,X}$. This risk is acceptable if we assume that in the case of an error, the guarantor itself, and not contractors, pays the penalty.

Under this assumption, the new expected value of the utility of signing contract c is:

$$\overline{U}^{G,r}_{X,c} = R_{c,X} \cdot P\{c \text{ honoured}\} + P_{c,X} \cdot P\{c \text{ not honoured}\} \tag{14.6}$$

where the G superscript indicates that some information from the guarantor is considered when signing the contract.

Under the assumption that p^G_k is the probability of the guarantor to provide erroneous information and that any error of the guarantor immediately causes the contract to fail, it is possible to express the total contract success and failure probabilities:

$$\begin{aligned} P\{c \text{ honoured}\} &= p_{c,Y} \cdot p^G_k & (14.7)\\ P\{c \text{ not honoured}\} &= (1 - p_{c,Y}) \cdot p^G_k + 1 - p^G_k \\ &= 1 - p_{c,Y} \cdot p^G_k & (14.8) \end{aligned}$$

where the latter is obtained by means of:

$$\begin{aligned} P\{c \text{ not honoured}\} = {} & P\{c \text{ not honoured}|\text{Guarantor succeeds}\} + \\ & P\{c \text{ not honoured}|\text{Guarantor fails}\} \end{aligned} \tag{14.9}$$

Using Equations (14.7), we can rewrite Equation (14.4) as:

$$\overline{U}^{G,r}_{X,c} = R_{c,X} \cdot p_{c,Y} \cdot p^G_k + P_{c,X} \cdot (1 - p_{c,Y} \cdot p^G_k) \tag{14.10}$$

Then, exploiting this equality in (14.4), we obtain the total average utility of signing the contract using information from a guarantor as:

$$\begin{aligned} \overline{U}^{G,r}_{X} &= \overline{U}^{G,r}_{X,c} \cdot p^s_{c,X} + I_{c,X} \cdot (1 - p^s_{c,X}) \\ &= [R_{c,X} \cdot p_{c,Y} \cdot p^G_k + P_{c,X} \cdot (1 - p_{c,Y} \cdot p^G_k)] \cdot p^s_{c,X} + \\ &\quad + I_{c,X} \cdot (1 - p^s_{c,X}) \end{aligned} \tag{14.11}$$

Since agents give a trust of one to their guarantors, most of the estimations of agent X are not changed by the mediation. In particular, the estimation of the contract success probability remains unchanged; therefore the estimation of the average utility of the contract does not change. Also, the estimation of the expected utility as a function of the probability of signing (Equation 14.5) is not influenced. As explained later on, the mediation of the guarantor influences only the decision making strategy.

14.3.3 Decision Making Strategy

Using the previous results, we can introduce a rationality principle in our model by means of a decision making strategy that exploits a utility to discriminate on the inclusion of the mediation of a guarantor into an interaction.

Trust PDF and the Risk Factor

As we said in the introductory text of Section 14.3, we model trust from the point of view of an agent as the estimation of the probability of having a contract honoured by its counterpart. An underlying assumption of this definition is that this estimation, and the real probability of the contract being honoured, both lie in the interval $[t_{min}, t_{max}]$. In essence, trust is a random variable $\mathbf{t}$ whose *Probability Density Function* (*PDF*) depends on decision making strategies of the agents involved in the contract.

Taking the variable $\mathbf{t}$ and a rationality principle into account, it is easy to define the probability that agent X would sign a given contract c. In particular, we can state our rationality principle as follows:

> X decides to sign a contract c with Y if the estimated expected utility that it perceives is greater than the investment required to sign the contract

Which leads immediately to the following:

$$\begin{aligned} p^s_{c,X} &\doteq P\{\overline{U}^e_{X,c} > I_{c,X}\} \\ &= P\{R_{c,X} \cdot t_{c,X,Y} + P_{c,X} \cdot (1 - t_{c,X,Y}) > I_{c,X}\} \end{aligned} \tag{14.12}$$

A further elaboration of this equation yields:

$$\begin{aligned} p^s_{c,X} &= P\{t_{c,X,Y} \cdot (R_{c,X} - P_{c,X}) > I_{c,X} - P_{c,X}\} \\ &= P\left\{t_{c,X,Y} > \frac{I_{c,X} - P_{c,X}}{R_{c,X} - P_{c,X}}\right\} \end{aligned} \tag{14.13}$$

Where we supposed that $R_{c,X} - P_{c,X}$ is not zero. Now, if we define:

$$\kappa_{c,X} \doteq \frac{I_{c,X} - P_{c,X}}{R_{c,X} - P_{c,X}} \tag{14.14}$$

it is possible to express $p^s_{c,X}$ as:

$$p^s_{c,X} = P\{t_{c,X,Y} > \kappa_{c,X}\} \tag{14.15}$$

This last equation indicates that agent X signs contract c if its trust in the counterpart with respect to c exceeds $\kappa_{c,X}$, that we call *risk factor*. This factor depends only on X's subjective evaluation of contract c and it describes the risk that X perceives in signing contract c. This, allows to rephrase the decision making strategy as:

> *Agent X signs a contract c with a counterpart Y if and only if its trust in Y for contract c is greater then the risk factor of c.*

It is worth noting that risk factor $\kappa_{c,X}$ is a number between zero and one. Furthermore, it is the quotient of two quantities that has a precise meaning on its own:

- The numerator $N_{c,X} = I_{c,X} - P_{c,X}$ expresses the gain that agent X obtains when rejecting contract c, in comparison to the case in which the contract is accepted but actually not honoured.
- The denominator $H_{c,X} = R_{c,X} - P_{c,X}$ represents the gain that the contract yields in case of success with respect to failure.

Then, e.g., if we consider the boundary cases:

- $\kappa_{c,X} = 1$ means that the contract will never be signed, because the investment equals the utility, but the first is guaranteed while the second is not.
- $\kappa_{c,X} = 0$ means that the contract has no risk, since the investment equals the penalty (which is assured with probability one). Therefore the contract will always be accepted.

In particular, if $\kappa_{c,X} \leq t_{min}$ the contract is always rejected, while if $t_{max} \leq \kappa_{c,X}$ the contract is always accepted. This consideration accounts also for the boundary cases analysis explained above.

Having introduced the risk factor $\kappa_{c,X}$, it is possible to rewrite Equation (14.4) putting some emphasis on it. In particular:

$$\begin{aligned} \overline{U}^r_X &= [R_{c,X} \cdot p_{c,Y} + P_{c,X} \cdot (1 - p_{c,Y})] \cdot p^s_{c,X} + I_{c,X} \cdot (1 - p^s_{c,X}) \\ &= [(R_{c,X} - P_{c,X})p_{c,Y} + P_{c,X}] \cdot p^s_{c,X} + I_{c,X}(1 - p^s_{c,X}). \end{aligned} \tag{14.16}$$

Now, explicitly showing $p^s_{c,X}$ and subsequently $(R_{c,X} - P_{c,X})$:

$$\begin{aligned} \overline{U}^r_X &= [(R_{c,X} - P_{c,X})p_{c,Y} + P_{c,X} - I_{c,X}] \cdot p^s_{c,X} + I_{c,X} \\ &= (R_{c,X} - P_{c,X}) \cdot (p_{c,Y} - \kappa_{c,X}) \cdot p^s_{c,X} + I_{c,X}. \end{aligned} \tag{14.17}$$

This last equation gives the possibility to draw some interesting considerations. First, $\overline{U}^r_X$ is bounded between $P_{c,X}$ and $R_{c,X}$. Furthermore, $\overline{U}^r_X$ is a linear function of $p^s_{c,X}$, and its slope is $(R_{c,X} - P_{c,X}) \cdot (p_{c,Y} - \kappa_{c,X})$. Since $(R_{c,X} - P_{c,X})$ is non negative because of Equation (14.1), the sign of the slope is influenced by $(p_{c,Y} - \kappa_{c,X})$ only. This ultimately means that the risk factor is an indicator of convenience in terms of average utility:

- If the success probability of the contract is greater than $\kappa_{c,X}$, then the average utility (of X) increases with the probability of signing the contract, i.e., the contract is advantageous.
- If the risk factor is lower than $\kappa_{c,X}$, the contract is disadvantageous and the average utility decreases with $p^s_{c,X}$.
- If $\kappa_{c,X} \equiv p_{c,Y}$, the average utility is constant.

Role of the PDF of Trust

The only working assumption that we made up to now is that $\mathbf{t}$ is a random variable bound by t_{min} and t_{max}. Here, we further elaborate on trust as a random variable and, without breaking our blackbox approach, we go for the worst case and we assume that $\mathbf{t}$ is uniformly distributed in interval $[t_{min}, t_{max}]$. This new assumption allows us to study the influence of the mediation of a guarantor on the average utility perceived by agents.

In accordance with Equation (14.15), we can express the signing probability as the probability that $t_{c,X,Y} \geq \kappa_{c,X}$. Therefore:

$$p^s_{c,X} = P\{t_{c,X,Y} > \kappa_{c,X}\} = \int_{\kappa_{c,X}}^{+\infty} f(t_{c,X,Y})\, \mathrm{d}t_{c,X,Y} \tag{14.18}$$

Then,

$$p^s_{c,X} = \begin{cases} 1 & \kappa_{c,X} \leq t_{min} \\ \frac{t_{max} - \kappa_{c,X}}{t_{max} - t_{min}} & t_{min} < \kappa_{c,X} < t_{max} \\ 0 & t_{max} \leq \kappa_{c,X} \end{cases} \tag{14.19}$$

Now, we focus our analysis of the utility on the case in which $t_{min} \leq \kappa_{c,X} \leq t_{max}$, i.e., we exclude the edge cases. Moreover, we assume symmetric PDF of $\mathbf{t}$. Introducing (14.19) in (14.17) we obtain the average utility as a function of t_{min} and t_{max}:

$$\overline{U}^r_X = (R_{c,X} - P_{c,X}) \cdot (p_{c,Y} - \kappa_{c,X}) \cdot \frac{t_{max} - \kappa_{c,X}}{t_{max} - t_{min}} + I_{c,X}. \tag{14.20}$$

Then, using a symmetric PDF of $\mathbf{t}$ with width δ it is possible to rewrite (14.19) as:

$$p^s_{c,X} = \begin{cases} 1 & \kappa_{c,X} \leq t_{min} \\ \frac{t_{max} - \kappa_{c,X}}{t_{max} - t_{min}} & t_{min} < \kappa_{c,X} < t_{max} \\ 0 & t_{max} \leq k \end{cases} \tag{14.21}$$

And then:

$$p^s_{c,X} = \begin{cases} 1 & \kappa_{c,X} \leq t_{min} \\ \frac{p_{c,Y} + \delta - \kappa_{c,X}}{2\delta} & t_{min} < \kappa_{c,X} < t_{max} \\ 0 & t_{max} \leq k \end{cases} \tag{14.22}$$

which expresses $p^s_{c,X}$ as a function of δ. Substituting this equation in Equation (14.20) and excluding the edge cases, it yields:

$$\overline{U}^r_X = (R_{c,X} - P_{c,X}) \cdot (p_{c,Y} - \kappa_{c,X}) \cdot \frac{p_{c,Y} + \delta - \kappa_{c,X}}{2\delta} + I_{c,X}. \tag{14.23}$$

This equation expresses the average utility as a function of the width of the probability density function of $\mathbf{t}$. Since the utility is a hyperbolic function of δ, any small decrease of δ implies a much higher increase in the average utility and vice versa.

Equation (14.23) has the following interesting consequence on the behaviour of the utility. If agent X takes its decisions of signing a contract c using a symmetric PDF for $\mathbf{t}$ centred in $p_{c,Y}$ and if the contract does not fail because of X, then for all $\delta \in \Re$ such that $\delta \geq 0$ and $t_{min} - \delta \geq 0$ and $t_{max} + \delta \leq 1$, we have that $\overline{U}^r_X(\delta)$ is non-increasing. In fact, $\overline{U}^r_X$ is piecewise differentiable and the differentiation of (14.23) for $t_{min} < \kappa_{c,X} < t_{max}$ yields:

$$\begin{aligned} \frac{\partial \overline{U}^r_X}{\partial \delta} &= (R_{c,X} - P_{c,X}) \cdot (p_{c,Y} - \kappa_{c,X}) \cdot \frac{2\delta - 2(p_{c,Y} - \kappa_{c,X} + \delta)}{4\delta^2} \\ &= -\frac{(R_{c,X} - P_{c,X}) \cdot (p_{c,Y} - \kappa_{c,X})^2}{2\delta^2} \end{aligned} \tag{14.24}$$

Taking into account that a subjective evaluation is well formed if $R_{c,X} \geq P_{c,X}$, the partial derivative is always non-positive, i.e., an increment of the estimation (which introduces uncertainty), worsens the performance of the agent's decision strategy and its relative utility.

The explicit choice of a PDF for trust $\mathbf{t}$ allows elaborating on the inclusion of mediation into an interaction. The two parameters $\kappa_{c,X}$ and $p_{c,Y}$ are kept fixed, since the mediation of a guarantor does not change and/or influence them. On the contrary, the total error probability is modified to account for the additional probability of error that the guarantor brings. Using Equation (14.7), it is possible to directly substitute $p_{c,Y}$ with $p_{c,Y}\, p^G_k$ to express the total success and failure probabilities, thus obtaining the equivalent of (14.23) for the case of mediated interactions. To stress the fact that the width of the estimation is different when introducing a guarantor in the interaction, we use the notation δ^G instead of δ:

$$\overline{U}^{G,r}_X = \begin{cases} H_{c,X} \cdot M^G + I_{c,X} & \kappa_{c,X} \leq t_{min} \\ H_{c,X} \cdot M^G \cdot \frac{p_{c,Y} + \delta^G - \kappa_{c,X}}{2\delta^G} + I_{c,X} & t_{min} < \kappa_{c,X} < t_{max} \\ I_{c,X} & t_{max} \leq \kappa_{c,X} \end{cases} \tag{14.25}$$

Where we defined (see later on) $M^G = (p_{c,Y}\, p^G_k - \kappa_{c,X})$.

Worst-Case Analysis

In order to study the effect of mediation in our model, we recall that our main working assumption is that guarantors provide additional information to agents, thus allowing for a more precise — narrower — estimation of probability $p_{c,Y}$. Anyway, guarantors, although highly reliable, introduce additional error probability that must be compensated by improvements in the estimation of trust.

In order to quantify the performance of a guarantor as a middleman in an interaction between agent X and Y, we calculate the amount of additional information that a guarantor needs to provide in order to keep the average utility of agent X fixed.

The comparison of the two utilities expressed in Equations (14.23) and (14.25) allows to calculate the width of guarantor-mediated estimation of trust for which

the utility equals the case without mediation. If we introduce $\hat{\delta}^G$ as a function of δ and p_k^G where $\hat{\delta}^G \doteq \delta_G \in \left[0, \frac{1}{2}\right]$ such that:

$$\overline{U}^r_X(\,\delta\,,\,p_{c,Y}\,) = \overline{U}^{G,r}_X(\,\delta_G\,,\,p_{c,Y}\cdot p_k^G\,) \tag{14.26}$$

we can compare Equations (14.23) and (14.25) to obtain:

$$(p_{c,Y}-\kappa_{c,X})\cdot\frac{p_{c,Y}+\delta-\kappa_{c,X}}{\delta} = (p_{c,Y}\,p_k^G-\kappa_{c,X})\cdot\frac{p_{c,Y}+\hat{\delta}^G-\kappa_{c,X}}{\hat{\delta}^G} \tag{14.27}$$

where we subtracted $I_{c,X}$ on both sides and multiplied by $\frac{2}{R_{c,X}-P_{c,X}}$. Then, introducing $M = (p_{c,Y}-\kappa_{c,X})$ and $M^G = (p_{c,Y}\,p_k^G-\kappa_{c,X})$:

$$M\cdot\frac{\delta+M}{\delta} = M^G\cdot\frac{\hat{\delta}^G+M}{\hat{\delta}^G} \tag{14.28}$$

and dividing by M^G yields:

$$\frac{M}{M^G}\cdot\frac{\delta+M}{\delta} = \frac{\hat{\delta}^G+M}{\hat{\delta}^G} \tag{14.29}$$

This last equation allows to make $\hat{\delta}^G$ explicit:

$$\hat{\delta}^G = \frac{M}{\frac{M}{M^G}\cdot\frac{\delta+M}{\delta}-1} = \frac{M\,M^G\,\delta}{M(\delta+M)-M^G\,\delta} \tag{14.30}$$

that holds if $M^G \neq 0$.

It should be quite clear that $\hat{\delta}^G$ is the breakeven point that makes agent X choose to go for a mediated interaction rather than for a direct interaction:

- If the guarantor provides enough information to restrict the estimation of trust to a width less than $2\,\hat{\delta}^G$, the use of the mediation is advantageous.
- If the estimation remains larger than $2\,\hat{\delta}^G$, the error probability introduced by the guarantor decreases the average utility.

It is worth noting that this decision strategy is purely ideal because agent X does not know p_k^G.

In order to ground our model in everyday experience, we recall that we are interested in guarantors that introduce a very low error probability, and therefore we study the behaviour of $\hat{\delta}^G$ as p_k^G tends to one. What we obtain from this study is that if agent X makes its decisions using a symmetric PDF and that the contract does not fail because of X, $\forall\delta \in \Re : 0 \leq \delta \leq \frac{1}{2}$, $\delta - \hat{\delta}^G$ tends to zero in a hyperbolic way as p_k^G tends to one. Due to editorial reasons, we cannot provide details on the demonstration of this result. Anyway, this result shows that if a guarantor introduces a (sufficiently) low probability of error, the use of its mediation is advantageous and the advantages that it brings rapidly increase as the probability of error decreases.

14.4 Integration in the CASCOM Platform

Security, privacy- and trust-awareness are crosscutting features of the CASCOM platforms that are spread across all its layers to provide the application developer with different services at different layers. For the sake of readability, we somehow oversimplify the description here and we split these features into the IP2P Network Layer and the Service Coordination Layer only — see other chapters in this book for an indepth description of these layers and of the overall architecture of the CASCOM platform.

14.4.1 IP2P Network Layer

The IP2P Network layer provides the application developer with most of the standard security features that we need to guarantee an adequate level of security for real-world communications between software entities. In detail, this layer transparently accommodates cryptography and non-repudiability of pairwise communications. Application developers are not involved in securing communications and, once activated, the transport security mechanism is in charge of the whole process of encryption and decryption for all inbound and outbound messages.

Basically, messages are filtered using a standard asymmetric cryptography method and security and non-repudiability are guaranteed because agents are requested to complete the registration of their services with Directory Facilitators with an X.509 certificate to be used in communications. Client agents accessing the services of service-provider agents will provide their X.509 certificate in the act of requesting the service.

The adopted cryptography scheme may be very demanding in terms of communication bandwidth and this is not always acceptable in a mobile environment. In order to provide application developers with a fine-grained control over the cryptography overhead, we allow the level of security to be customized on message basis. For each and every single outbound message, an agent can choose to (i) encrypt the message or not and for encrypted messages to (ii) choose between encrypting the whole ACL message or only its SL content.

The IP2P Network Layer security module is embedded in the core of the CASCOM platform as a pluggable service and it is ubiquitously deployable because it is both MIDP and J2SE compliant. It uses a downsized version of the Bouncy Castle Crypto API [1] and its memory footprint and runtime requirement are compatible with nowadays mobile devices. Two new `ACLCodec` and `SLCodec` [10] implementations are provided to guarantee the possibility of fine tuning the overhead of encryption on message basis.

14.4.2 Service Coordination Layer

The Service Coordination layer sits on top of the IP2P Network layer and it exploits its services to provide novel, agent-layer services with a high level of

abstraction. This is the reason why we use to say that the Service Coordination layer raises the level of abstraction of the secure messaging of the IP2P Network layer towards the realization of full privacy- and trust-awareness services. In detail, such services provide:

- Guarantor-mediated ACL messaging that ensures trustworthy and possibly anonymized communications.
- Privacy-awareness of storage and of possibly anonymized communications.

Unfortunately, we cannot provide developers with a transparent tool like we did for secure messaging at the IP2P Network layer. Policies for ensuring privacy-awareness have to take into account the intended usage and the inherent nature of transmitted data for being correctly applied. At the service coordination layer, transmitted data is no longer an opaque stream of bytes, rather it is a source of possibly classified information that agents should protect against malevolent usage. ACL messages containing sensitive data are easily interleaved with messages that do not contain them and we need application developers to classify which message is potentially privacy-critical. Therefore, the CASCOM privacy- and trust-awareness services are agent-level services that developers must explicitly address. This is the reason why we developed a framework for CASCOM-based applications capable of providing a direct support to developers in the classification of data for communication and storage.

We designed our framework to match a set of fundamental requirements, that resulted in strict development guidelines, as follows.

- Security. All communications must be secured and directed to trusted parties.
- Traceability. Messages must be signed by the sender, while responses must be signed, directly or indirectly, by a guarantor. The framework transparently enforces this property and it provides a transparent tracing service that logs all communications.
- Locality. The number of trusted parties involved in a communication must be kept minimal.
- Transparency. The use of guarantors in trustworthy communications must be transparent to the application developer — he/she is not directly involved in the use guarantors' services.
- Ease of use. The framework must provide high level procedures to perform common tasks, as well as low level, more specific procedures devoted to fine-grained and less common tasks.
- Standardization. Information exchange, including messages and certificates, must be performed using well-known and accepted formats.

The design of the CASCOM framework for privacy and trust awareness is split into two views: (i) a *Client view* that groups the classes that a client agent can

use to access the services of the privacy- and trust-awareness framework, and (ii) a *Guarantor view* that contains the components that the guarantors use to implement their functionality. These views are connected through a Java interface, named `Guarantor`, that plays the logical role of a remote interface that guarantors implement and that client exploit to communicate with guarantors. This uses a classic *half object plus protocol* design pattern and application developers are only interested in the use of the stubs of interface `Guarantor`.

It is worth noting that the client view represents a mandatory interface while the guarantor view is only one of the possible internal implementation of guarantors. Obviously, the client view plays a substantially more important role in this design.

The use of the services that the client views of the framework provides always starts with an authentication phase that clients perform to achieve a mutual recognition with a guarantor. Once a client is authenticated with a guarantor, it can exploit the mediation of the guarantor to request for services in a trustworthy and possibly anonymized way. In detail, a client can perform three kinds of requests for services:

- Direct requests, i.e., requests for services whose outcome is used by the client itself;
- Indirect requests, i.e., requests that are performed on behalf of some other client.

Direct requests are ordinary requests for services, except for the following two constraints:

- Parameters and results are transported on a secured channel;
- The guarantor acts as a middleman of this request and it is responsible for tracing the request to guarantee non-repudiability;
- The client is responsible for providing a distributed timestamp to allow for traceability of complex interactions.

Indirect requests are a delegation mechanisms that allow a client (B, delegated) to have a service performed on behalf of another client (A, delegator). Indirect requests are implemented with the following steps:

- Client A requests its guarantor to grant indirect requests to client B;
- If the guarantor can honor the request of A, it accepts requests from B and serve them as if they were requested by A;
- The guarantor stops serving indirect requests from B when the delegation expires, e.g., because the maximum number of requests from B is reached.

This starts when A creates a delegation token for B using the `Guarantor` interface. If the guarantor can grant the delegation, a globally unique token that identifies

the delegation is created and provided to the B. The delegated client B uses this token to finally access the services through the guarantor with no further authentication, i.e., there is no mutual recognition between B and the guarantor.

Indirect requests allows chaining trust and constructing a network of trust on the fly, thus avoiding any static structure that mutual recognition of guarantors might imply. Moreover, they are a good way to allow a third party having a service done without explicitly requesting mutual recognition, i.e., it is a good way to carry out anonymized communications.

The guarantor view of the framework describes how guarantors implement their services with the requested level of security and privacy awareness. Every guarantor may decide its own optimized approach to provide services as long as the `Guarantor` interface is honoured and therefore the guarantor view of the framework is only one of the possible ways to implement guarantors. Anyway, high-quality guarantors are not easy to implement because they need to deal with somehow tricky issues, e.g., the global uniqueness of the generated tokens and the correct tracing of invocations, and therefore the guarantor view of the framework plays an important role in real applications.

One of the principal parts that build the guarantor view of the framework is the so called *sensitive data store*. This is a general-purpose means for data storage that is meant to allow for a seamless treatment of sensitive data. It is worth noting that every Nation in the European Community adopted laws to provide guarantees to citizens regarding the treatment of their sensitive data. Such laws are all rooted in a note of the European Commission and they all contain strict technical requirements that databases of sensitive data must follow. The sensitive data store ensures that the basic principles of the correct treatment of sensible data are respected and it ensures the possibility of fine tuning its policies to comply with national laws.

The second part of the guarantor view of the framework is the so called *message tracer*. This component provides all methods for tracing direct and indirect requests served or rejected by the guarantor. Such requests are stored in a sensitive data store that saves all information regarding requests. It is worth noting that the use of a distributed time stamp in indirect requests allows to correlate traces of different guarantors, and therefore it allows backward tracing communications across a complex network of clients and guarantors.

14.5 Summary

The work reported in this chapter has two main objectives: (i) first it studies trust from a quantitative point of view and demonstrates that mediated interactions are mandatory to achieve privacy- and trust-awareness in real-world multiagent systems; (ii) it shows an overview of the infrastructure that CASCOM platform provides to provide support for these abstractions. Such an infrastructure provides notable features that are not discussed here, that play a fundamental role from

the point of view of scalability and reliability (see other chapters in this book). Then, in many cases, the additional utility that mediation provides to agents is considerable even through guarantors are not error-free.

This work is not meant to be conclusive and many points remain open. One of the major planned developments regards the study of concrete trust estimators, and the introduction of the resulting PDFs in our model. Another very important open point regards the study of the effects of delegation of tasks and goals through a chain of delegated guarantors.

Furthermore, the study of one of the main features of guarantors, i.e., the possibility of anonymising interactions, is still in search of a formalization — and of a stochastic model — even though its characteristics and possible uses are clearly understood [2]. This kind of interaction allows to prevent unwanted spread of sensible information; as such, its study remains central in the evaluation of the agent's benefits from the guarantor infrastructure.

References

[1] Bouncy Castle Crypto API Web site. `http://www.bouncycastle.org`

[2] F. Bergenti, R. Bianchi and A. Fontana: Secure and Trusted Interactions in Societies of Electronic Agents. In Proceedings of *The* 4^{th} *Workshop on the Law and Electronic Agents (LEA 2005)*, 1–12. Bologna, Italy. 2005. Wolf Legal Publishers.

[3] R. Bianchi, A. Fontana and F. Bergenti: A Real–World Approach to Secure and Trusted Negotiation in MASs. In Proceedings of *The* 4^{th} *International Joint Conference on Agents and Multi-Agents Systems (AAMAS)*, 1163–1164. Utrecht. The Netherlands. 2005. ACM Press.

[4] R.W.H. Bons: *Designing Trustworthy Trade Procedures for Open Electronic Commerce*. Ph.D.diss., EURIDIS and Faculty of Business Administration, Erasmus University, Rotterdam, The Netherlands. 1999.

[5] CASCOM Web site. `http://www.ist-cascom.org`

[6] C. Castelfranchi and R. Falcone: Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification. In Proceedings of *The International Conference of Multi-agent Systems (ICMAS)*, 72–79. Paris, France. 2005. ACM Press.

[7] C. Ellison: *SPKI Requirements*. IETF RFC 2692. 1999.

[8] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas T. Ylonen: *SPKI Certificate Theory*. IETF RFC 2693. 1999

[9] D. Gambetta (Ed.). *Trust: Making and Breaking Co-operative Relations*. Basil Blackwell, Inc. 1988.

[10] JADE Team. *JADE Programmers Guide*. Available at `http://jade.tilab.it`

[11] JENA Web site. `http://jena.sourceforge.net`

[12] N.R. Jennings, S. Parsons, C. Sierra and R. Faratin: Automated Negotiation. In Proceedings of *The* 5^{th} *International Conference on the Practical Application of Intelligent Agents and Multi-Agents Systems (PAAM-2000)*, 23–30. Manchester, UK. 2000.

[13] S. Marsh: *Formalising Trust as a Computational Concept*. Ph.D. dissertation, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK. 1994.

[14] MINDSWAP Team. *A Definition of Trust for Computing with Social Networks* Technical report, University of Maryland, College Park, February 2005.

[15] OWL Web site. `http://www.w3.org/2004/OWL`

[16] Racer Web site. `http://www.sts.tu-harburg.de/~r.f.moeller/racer`