THE EXPLOITATION-RESISTANT TRUST (ERT) MODEL FOR OPEN DISTRIBUTED SYSTEMS

by
Amirali Salehi-Abari

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

MASTER OF COMPUTER SCIENCE

School of Computer Science

at

CARLETON UNIVERSITY

Ottawa, Ontario
April, 2009

Your file Votre référence
ISBN: 978-0-494-52047-5
Our file Notre référence
ISBN: 978-0-494-52047-5

Table of Contents

List of Tables

List of Figures

Abstract

Artificial societies - distributed systems of autonomous agents - are becoming increasingly important in open distributed environments, especially in e-commerce. Agents require trust and reputation concepts in order to identify communities of agents with which to interact reliably. Many different definitions for trust and reputation have been proposed that incorporate multiple sources of information. Much of the research on trust and reputation models deals with unrealistic attack models. We have noted in real environments that adversaries tend to focus on exploitation of the trust and reputation model. Then, we have noted the exposure of such models to individual and collusion attacks. These vulnerabilities reinforce the need for new evaluation criteria for trust and reputation models called exploitation resistant which reflects the ability of a trust model to be unaffected by agents who try to manipulate the trust model.

We introduce a decentralized Exploitation-Resistant Trust (ERT) model. ERT incorporates multiple sources of information to assess the trustworthiness of agents and is exploitation resistant against both individual and collusion attacks. We propose a new type of individual attack called the con-man attack and formally model it. As a representative of collusion attacks, we model a witness-based collusion attack.

To evaluate our proposed trust model, we describe the design of a decentralized game-theoretic trust and reputation environment model (testbed). Not only is the proposed environment model compatible with the characteristics of open distributed systems, but also allows agents to have different types of interactions in this environment model. Besides direct, witness and introduction interactions, agents in our environment model can have a novel type of interaction called a reporting interaction which represents a decentralized reporting mechanism in distributed environments.

We empirically demonstrate the vulnerability of three well-known trust models against the con-man attack and show how ERT is resistant against the con-man attack. Our experiments show how unidimensional trust models are vulnerable to the witness-based collusion attack while ERT show the robustness against that attack because of being multi-dimensional.

Acknowledgements

First and foremost, I am grateful to my thesis supervisor, Prof. Tony White. It is as a result of his excellent guidance and support that this research has been possible. I am unable to quantify the importance of his academic guidance and valuable advice regarding my studies at Carleton. I will not forget his guidance and advice. His tenacity and persistence in our weekly discussions and his conscientious nature demonstrate the excellence of his character. Moreover, I am greatly appreciative of the financial support that he provided during my studies and also for supporting me in attending several academic conferences.

Thanks should also go to the members of my thesis committee, Prof. Paul van Oorschot and Prof. Thomas Tran for their guidance and for their carefully considered review of this thesis. The document is better as a result of their thorough examination of it.

Thanks to the administrative staff of the School of Computer Science at Carleton University, especially Claire Ryan and Sharmila Namasivayampillai, who helped me in many ways over the course of my studies.

Finally, I thank all of my wonderful friends and family for their support. I am so grateful that I have such wonderful parents and I owe all of my education and success from my childhood to now to their effort and support. I know there is no way for me to return their multi-faceted support in my life but I will never forget all the things that they have done for me. I especially thank Omid, who always cares about my success. Last but certainly not least, I would like to offer my special thanks to Julie not only for her indefatigable academic discussions but also for her immense support.

Chapter 1

Introduction

1.1 Introduction

Recently, many computer applications are open distributed systems in which the components are located on a large-scale network. These systems are decentralized and subject to change over the system's lifetime. E-business systems, peer-to-peer systems [55], web services [48], the Semantic Web [8], and pervasive computing [67] fall into the category of open distributed systems. With the growth of these open distributed systems through the Internet, artificial societies have been formed in these environments. Furthermore, the society of intelligent agents, which may eventually interact on the behalf of their users in e-commerce marketplaces [17], can be viewed as these artificial societies. As a consequence, real-world assumptions and the whole range of possible social behaviors need to be taken into account in these artificial societies.

By analogy with human societies in which trust is one of the most crucial concepts driving decision making and relationships, *trust* is indispensable when considering interactions among entities (individuals) in these artificial societies. According to Jarvenpaa et al. [33], trust is an essential aspect of any relationship in which the trustor does not have direct control over the actions of a trustee, the decision is important, and the environment is uncertain.

Personal experience with others might build up trust. We use the experience that we gain in interacting with others to judge how they will perform in similar situations. However, when we need to assess our trust in someone of whom we have no direct personal experience, we often ask others regarding their personal experience with this individual. This collective opinion of others regarding the specific individual is known as an individual's reputation.

Trust and reputation have been studied and used in various fields from different

perspectives. For instance, Nowak and Sigmund have explained why selfish individuals cooperate using reputation concepts [53]. The concept of trust in economics and business was discussed first by Akerlof when he introduced "the market of lemons problem" [2]. He identified certain severe problems of markets characterized by asymmetrical information. Economists have used trust and reputation to explain *irrational* behavior of players in repeated economic games [41, 46]. Computer scientists have used trust and reputation for modeling trustworthiness of entities and individuals in open distributed systems (e.g., online marketplaces, multi-agent systems, and peer-to-peer systems) [58, 52]. This thesis concentrates on trust and reputation models for open distributed systems. Moreover, it is the view of this thesis that complicated trust and reputation models are not universally implemented because of the possibility of exploitation of those models.

Amazon [3] and eBay [21] are important practical examples of reputation management systems. In these systems, the sellers list their items for sale and buyers bid for these items. Users are allowed to rate sellers and submit textual comments. The overall reputation of a seller is the average of the ratings obtained from his customers. Several researchers have postulated that seller reputation has significant influence on prices, especially for high-valued products in the ebay market [30, 59]. Similarly, Brainov and Sandholm [11] have studied the impact of trust on contracting in e-commerce marketplaces. Their approach shows the amount of trade and agents' utility functions are maximized when the seller's trust is equal to the buyer's trustworthiness. Moreover, they show that advanced payment contracts can eliminate inefficiency caused by asymmetric information about trust and improve the trustworthiness between sellers and buyers. These studies all imply the importance of trust and reputation models in open distributed systems, especially e-commerce marketplaces.

Similar to eBay, one approach to building a trust or reputation model is to have a central agency that keeps records of the recent activity of the users in the system, very much like the scoring systems of credit history agencies (e.g., Kasbah [17]). However, these centralized approaches require considerable overhead on behalf of the providers of the online community and failure of the agency causes failure in all parts

of the system. Moreover, they are not compatible with most of the characteristics and limitations of open distributed system. Generally, since there is no central authority in a pure open distributed system, a centralized trust model is not suitable for these systems. For example, there is no trusted introducer service as is found in some distributed systems. That is why the scope of this thesis is decentralized trust and reputation models.

The majority of open distributed computer systems can be modeled as multi-agent systems (MAS) in which each component acts autonomously to achieve its objectives [34]. An important class of these systems is one that is *open* in terms of joining and leaving the system. Huynh et al. [32] pointed out three interesting features of these systems: (1) the agents are likely to be self-interested and may be unreliable; (2) no agent can know everything about its environment. In other words, there is no global perspective and (3) no central authority can control all the agents due to different ownership. A key component of these open MAS is the interactions that certainly have to take place between agents. Furthermore, since agents have incomplete knowledge about their environments and other agents, trust and reputation plays crucial roles in these interactions.

To reach its goals, an agent usually requires resources that only other agents can provide. The agent benefits from choosing the agents with which it interacts such that they can provide those resources. In this light, the agent can minimize the risk of unsuccessful interactions and failure by predicting the outcome of interactions, and avoiding risky (unreliable) agents. Modeling the trustworthiness of the potential interaction partners enables the agent to make these predictions. Furthermore, analogous to the legal systems in which a rule violation generates a legal punishment for the offender, in the social world the penalty for a violator who violates a social norm is a bad reputation [15]. Specifically, trust and reputation provide a form of social control in open distributed systems in which agents are likely to interact with unvisited and unknown agents.

It is important to note that the notion of trust in cryptography (hard trust) is out of the scope of this thesis. Specifically, this thesis does not address the issue of agent identification; i.e., the authentication problem. Hard trust often refers to

mechanisms to verify that the source of information is really who the source claims to be [4]. For these problems, cryptographic algorithms such as digital signatures and encryption/decryption mechanisms [49] should allow the receiver of the information to verify the source or sender of that information. In other words, hard trust approaches, which are based upon provable properties such as is found in cryptography, help to verify that the partner you are interacting with is authenticated and authorized to take various actions. They do not ensure that the party will behave appropriately based on your expectation of service delivering. The scope of this thesis is limited to decentralized computational trust and reputation models.

1.2 Goals

The main utility of trust and reputation models can be summarized as minimizing the the risk of interacting with others. To reach this goal, an agent must be able to model trustworthiness of potential interaction partners and make decisions based on those models. Therefore, research objectives in this thesis firstly include defining and evaluating trust models with desirable characteristics for open distributed systems – as explained below – and secondly defining agent policies based on its trust models that assist agents in isolating the untrustworthy (undesirable and unreliable) agents from the society [80].

1.3 Problem Statement

Fullam et al. [26] has defined the following set of criteria to evaluate trust and reputation models: (1) the model should be multi-dimensional; (2) converge quickly; (3) precisely model the agent's behavior; (4) be adaptive: the trust value should be adapted if the target's behavior changes; (5) be efficient in terms of computation.

Although trust and reputation models have a strong foundation on the assumption that agents may attempt to exploit each other, there is little consideration regarding the possibility that agents may attempt to exploit the trust and reputation model itself. In this regard, we believe that in addition to the criteria explained above, *exploitation resistance* is a crucial feature of trust models. Exploitation resistance

reflects the ability of a trust model to be impervious to agents who try to manipulate the trust model and who aim to abuse the presumption of trust. More precisely, exploitation resistance implies that adversaries cannot take advantage of the trust model and its associated systems parameters even when they are known or partially known to adversaries.

We categorize the possible exploitations into two groups: individual attacks and collusion attacks. Individual attacks are concerned with attacks that are mounted by only one agent while collusion attacks are usually mounted by the collaboration of a group of agents. In this thesis, we introduce a decentralized **Exploitation-Resistant Trust (ERT)** agent model. ERT incorporates multiple sources of information to assess the trustworthiness of agents and is exploitation resistant in the two dimensions of individual attacks and collusion attacks.

This thesis solves the following problem: What trust models and policies can be created for a heterogeneous agent population such that even if their trust models are known to adversarial agents, employing a given set of attacks, the community of untrustworthy agents can be determined?

1.4 Contributions

Our contributions in this thesis include the following:

- The design of a decentralized game-theoretic trust and reputation environment model (testbed). The proposed environment model is compatible with the characteristics of open distributed systems mentioned in Section 1.1. Agents can have different types of interactions and consequently have access to different sources of information for assessment of other agents. Moreover, the proposed environment model provides the facility to define agents with various behaviors and is flexible enough to accommodate a variety of adversarial behaviors.

- We have introduced a novel type of interaction called the Reporting Interaction and its relevant trust dimension; this interaction facilitates a decentralized reporting mechanism in distributed environments.

- The introduction and proposal of the Exploitation-Resistant Trust (ERT) model which incorporates multiple sources of information to assess the trustworthiness of agents. This model incorporates six desirable characteristics of trust and reputation models described in Section 1.3.

- We introduced a new type of individual attack called the con-man attack. We formally model the con-man attack and empirically demonstrate the vulnerability of three well-known trust models against it.

- We proposed the desirable characteristics of con-resistant trust models and empirically evaluate a con-resistant component for ERT [65].

- We modeled witness-based collusion as a representative of collusion attacks through the introduction of the concept of a naive agent.

- We empirically analyzed the impact of naive agents on an agent society [64].

1.5 Organization

The remainder of this thesis is structured as follows. Chapter 2 reviews important background material used in this thesis. We discuss the state-of-the-art related work in Chapter 3 highlighting areas where the research reported in this thesis is intended to provide improvements. Chapter 4 and Chapter 5 describe the details of the environmental model and agent model of ERT respectively. The empirical experiments and corresponding results demonstrating the improvements highlighted in Chapter 3 are described in Chapter 6. Finally, we conclude the thesis with a summary of key messages, highlighting the contributions made, and discuss potential future work in Chapter 7.

Chapter 2

Background

2.1 Introduction

This chapter provides brief reviews of Multi-agent Systems, Game Theory and Reinforcement Learning to help readers better understand the material contained in this thesis. The explanations provided are not comprehensive, and we encourage readers to refer to the provided references for further exploration of these concepts.

The remainder of this chapter is organized as follows. Section 2.2 describes intelligent agents, agent-based models and multi-agent systems. Required game theory concepts and Prisoner's Dilemma are discussed in Section 2.3. A brief review of reinforcement learning is presented in Section 2.4. Finally, we summarize this chapter in Section 2.5 and indicate where the material is used within the following chapters.

2.2 A Brief Review of Multi-agent Systems and Simulation

2.2.1 Intelligent Agent (IA)

An intelligent agent (IA) is defined as an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals (i.e., it is goal-oriented) [61]. Jennings et al. [35] have a similar definition: "An agent is a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives."

Intelligent agents might have various intrinsic features. They might learn or use knowledge to achieve their goals. They can be very simple or very complex. In this light, Russell & Norvig [61] categorize agents into five classes based on their degree of intelligence and capability:

- **Simple reflex agents:** These agents act only on the basis of the current percept.

- **Model-based reflex agents:** Model-based reflex agents keep track of the current state of the world using an internal model. They then choose actions in the same way as the reflex agents.

- **Goal-based agents:** This kind of agent is a model-based agent which stores information regarding desirable situations. This allows the agent to choose among multiple possibilities to reach its goal state.

- **Utility-based agents:** This type of agent is similar to a goal-based agent with the difference that the agent employs a utility function and its goal is maximizing that.

- **Learning agents:** A learning agent has the capability of operating in unknown and noisy environments and to become more competent than its initial knowledge equipped it.

2.2.2 Agent-based Simulation/Agent-based Model (ABS/ABM)

For the design of complex and dynamic systems, simulation is commonly recognized as a valuable aid in analyzing these systems. Generally speaking, simulation is a decision support tool rather than a decision making tool. A simulation model consists of a set of rules that specify how a system changes over time by considering the current state of the system. When a simulation model is run, the changes of the system can be observed at any point in time which provides insight into system dynamics.

The key features of simulation modeling are abstraction and simplification. In this sense, a simulation model can only be an approximation of the target system and embraces those characteristics of the target system that are important for study and analysis. The purpose of simulation is either to better understand the operation of a target system, or to make predictions about a target system's utility.

Agent-Based Simulation (ABS) or Agent-based Models (ABM) are simulation methods which researchers employ to study and analyze complex systems. The models simulate the simultaneous operations of multiple agents, in an attempt to model and predict the actions of a complex system. The individual agents are supposed to act in a way that they perceive as being in their own interests, such as reproduction,

economic benefit, or social status, while acting upon limited knowledge [6]. ABM agents may be able to learn, adapt, and reproduce [9].

In an ABM the researcher explicitly describes the decision processes of an agent at the micro level. Structures emerge at the macro level as a consequence of the agents' interactions with other agents and the environment. ABM is a bottom-up approach and is used for environments with heterogeneous entities, where individual variability between the agents cannot be neglected and each agent might have personal motivations and incentives. ABM embraces elements of different domains such as: game theory, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming.

2.2.3 Multi-agent Systems (MAS)

While an agent is an entity with domain knowledge, goals and actions, multi-agent systems (MAS) comprise a set of agents which interact in a common environment. Using multi-agent systems, complex systems involving multiple agents are constructed and agent coordination can be addressed. Broadly speaking, a multi-agent system consists of autonomous interacting agents that coordinate their actions so as to achieve its goal(s) jointly or competitively [76]. In this sense, the Grid [24], the Semantic Web [8], pervasive computing [67], and peer-to-peer systems [55] can be viewed as multi-agent systems (MAS) where entities act in an autonomous and flexible way in order to achieve their objectives.

Multi-agent systems can be used to solve problems which are difficult for an individual agent to solve. Examples of problems which are appropriate for multi-agent systems research include online trading, and modeling social structures. Multi-agent systems can demonstrate complex behaviors, self-organization, emergence even when the individual strategies of all their agents are simple. The agents in a multi-agent system have the following characteristics [76]:

- **Autonomy:** The agent senses the environment, and acts on it over time in pursuit of its own agenda (goal).
- **Local views:** No agent has a full global view of the system (i.e., there is no global perspective).

- **Decentralization:** There is no one controlling agent. In other words, no central authority can control all the agents due to different ownership.

The study of multi-agent systems is closely connected with the development and analysis of sophisticated artificial intelligence problem solving techniques. The crucial research topics in MAS include:

- Cooperation
- Coordination
- Communication
- Negotiation

A key component of multi-agent systems is the interactions that certainly have to take place between agents. Furthermore, since agents have incomplete knowledge about their environments and other agents, trust and reputation plays crucial roles in these interactions. Cooperation and coordination are influenced by trust and reputation. It is the research problems inherent in agent trust and reputation in a decentralized system that motivates the work reported in this thesis.

2.3 A Brief Review of Game Theory

Game Theory is the branch of applied mathematics which has found applications in economics, evolutionary biology, sociology, political science, philosophy and computer science [43]. Recently, game theory has been used to model interactive computations and provides a theoretical basis for the field of multi-agent systems [69]. Game theory tries to mathematically capture behavior in strategic situations, where an individual's success depends on the choices of others.

Each individual (agent or player) in game theory is supposed to behave rationally. Rationality implies that each player tries to maximize its payoff. In this sense, each player has to decide among a set of moves which are in accordance with the rules of the game and which maximize his/her rewards. A game can be formally defined with the following components:

- Set of players: $D = \{P_i | 1 \leq i \leq n\}$
- Set of rules: $R = \{r_i | 1 \leq i \leq m\}$
- Set of Strategies S_i for each player P_i
- Set of Outcomes: $O = \{o_i | 1 \leq i \leq k\}$
- Payoff $u_i(o)$ for each player i and for each outcome $o \in O$

2.3.1 Types of Games

Cooperative vs. Non-cooperative

The games can be either of cooperative or non-cooperative. In a cooperative game, the players are able to form binding commitments (e.g., the legal system forces them to stick to their promises) and the basic modeling unit is the group; hence the game is a competition between groups of players, rather than between individual players. On the other hand, in a non-cooperative game, binding commitment is not possible and the basic modeling unit is the individual (e.g., his beliefs, preferences, and possible actions) thus players make decisions independently.

Symmetric vs. Asymmetric

A symmetric game is a game where the payoffs for playing a particular strategy depend only on the other strategies played, not on the identity of the player. In other words, a player does not have any distinct roles in the game, and the payoff for players does not depend on their identities. Otherwise the game is asymmetric.

Zero Sum vs. Non-zero Sum

A game is a zero-sum game where the total payoff for all players in the game (for every combination of strategies) always adds up to zero. Generally speaking, if the total gains of players are added up, and the total losses are subtracted, they will sum to zero.

Simultaneous vs. Sequential

In simultaneous games, the players move simultaneously. If they do not move simultaneously, the later players do not know the players' earlier moves. In contrast, sequential games are games where later players have some knowledge about earlier actions.

Perfect Information vs. Imperfect Information

A perfect information game is a game in which all players are aware of the previous moves of all other players. Therefore, only sequential games can be games of perfect information. In contrast, players are unaware of the other player's move in imperfect games. Interestingly, most games studied in game theory are imperfect information games.

2.3.2 Representation of Games

Games are represented differently based on their types. Cooperative games are usually presented in characteristic function form, while the extensive and normal forms are used to define non-cooperative games. We will explain extensive and normal forms in the following subsections.

Extensive Form

Using the extensive form, games are presented as trees where a vertex (node) represents a point of choice for a player and out-going edges of a given vertex represent possible actions for that player. The player is specified by a number listed above each vertex. The payoffs are specified at the bottom of the tree.

Normal Form

The normal form uses a matrix to present the game. The matrix includes the players, strategies, and payoffs. Generally, it can be represented by any function that associates a payoff for each player with every possible combination of actions. If a game is presented in normal form, it is presumed that each player acts simultaneously and

the game provides imperfect information. If players have some information about the choices of other players (i.e., a sequential game), the game will be presented in the extensive form.

2.3.3 Classical Prisoner's Dilemma

The Prisoner's Dilemma, a problem in game theory, was originally described by Merrill Flood and Melvin Dresher in 1950 while Albert W. Tucker formalized the game with prison sentence payoffs [57]. The prisoner's dilemma forms a non-zero-sum, non-cooperative and simultaneous game in which two players may each "cooperate" with or "defect" from the other player. Similar to other games in game theory, the goal of each individual is maximizing his/her payoff, without any concern for other player's payoff.

The prisoner's dilemma is formalized as follows. Two prisoners are arrested by the police for a crime and each has a choice of confessing to the crime or remaining silent. As the police do not have enough evidence to convict them, the police have separated them (i.e., no negotiation between criminals) and have the following deal for each: if one testifies against the other and the other remains silent, the betrayer goes free and the silent person go to the jail for 10 years. If they both choose to remain silent, the police will not be able to prove their case and they will stay in prison for a short term, say 1 year, for minor offenses. If each of the criminals betrays the other, each receives a five year sentence. Each prisoner must make the choice of whether to betray (defect) the other or to remain silent (cooperate with the other). However, since the prisoners are kept in separate rooms and cannot communicate with each other, neither prisoner knows what choice the other prisoner will make.

Clearly, there are four total outcomes depending on the choices made by each of the two prisoners. We can present this game in the normal form via the following two-by-two matrix.

P_1/P_2	**Cooperate (silent)**	**Defect (betray)**
Cooperate (silent)	1,1	0,10
Defect (betray)	10,0	5,5

Table 2.1: Payoff Matrix of the Prisoner's Dilemma

As shown in Table 2.1, each of the two prisoners P_1 and P_2 has two possible strategies: to "betray" or to remain "silent". The two strategies of prisoner P_1 correspond to the two rows and the two strategies of prisoners P_2 correspond to the two columns of the matrix. The entries of the matrix are the costs incurred by the players (left entry for the row player and the right entry for the column player).

In the Prisoner's Dilemma, cooperating is strictly dominated by defecting since the game will only be played once between individuals.

2.3.4 Iterated Prisoner's Dilemma

When two players play the Prisoner's Dilemma game repeatedly and they are able to remember the previous moves of their opponent, the game is called the Iterated Prisoner's Dilemma (IPD) [5]. In IPD, the players can change their strategy based on their observations.

It can be proved by induction that when the game is played exactly N times [5], the dominant strategy for both players is to defect N times. But if the player is unaware of the number N, and if the probability that two players play another game is high, then cooperation is the stable strategy.

2.4 A Brief Review of Reinforcement Learning

Machine learning, a subfield of artificial intelligence, focuses on the design and development of algorithms that help computers (programs) improve their performance over time (i.e., "learn") based on the perceived data. There are some similarities between animal and machine learning as many techniques in machine learning derive from the efforts of psychologists who propose computational learning models from their theories of animal and human learning.

There is a wide range of machine learning algorithms including Genetic Algorithms, Genetic Programming, Neural Networks, Swarm Intelligence, and Reinforcement Learning. Reinforcement Learning (RL) has attracted rapidly increasing interest in the multi-agent systems community. According to Mitchell [50], "Reinforcement learning addresses the question of how an autonomous agent that senses and acts in its environment can learn to choose optimal actions to achieve its goals". In other

words, its promise is to program agents by reward and punishment without needing to specify how the task is to be achieved. Reinforcement learning algorithms try to find a policy mapping the actions the agent can take to the states of the environment. Reinforcement learning has been successful in different domains, most notably in games [73].

There are two main streams for solving reinforcement learning problems [39]. The first is to search the space of behaviors and select the one that performs well in the environment. The second is to use statistical techniques and dynamic programming methods in order to predict the utility of taking actions in particular states of the world.

In the classic reinforcement learning model, an agent is connected with its environment through perception and action. The environment is typically formulated as a finite-state Markov decision process (MDP). The agent perceives the current state of the environment and then chooses an action. The action changes the state of the environment. Afterward, the value of this state transition is sent to the agent through a scalar reinforcement signal. Formally, the reinforcement learning model consists of [39]:

- A discrete set of environment states: $\mathbf{S}$
- A discrete set of agent actions: $\mathbf{A}$
- Reinforcement signals in $\Re$: $\mathbf{r}$

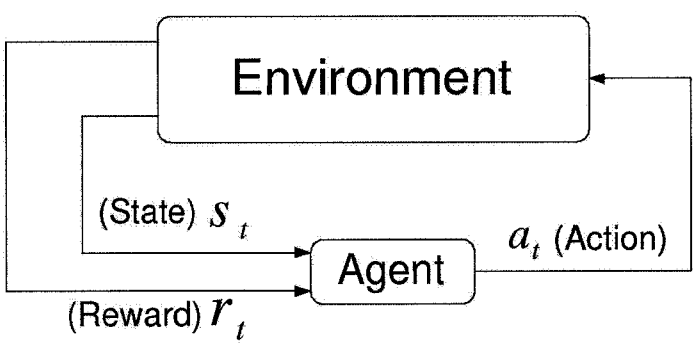


Figure 2.1: The Standard Reinforcement Learning Model.

As shown in Figure 2.1, when the agent perceives its state $s_t \in S$ at time t, amongst the set of possible actions $A(s_t)$, the agent chooses an action $a_t \in A(s_t)$. At time $t+1$, the agent receives from the environment the new state s_{t+1} and a reward r_{t+1}. Based on these interactions, the reinforcement learning agent must develop a policy $\pi : S \longrightarrow A$ which maximizes the quantity $R = r_0 + r_1 + \ldots + r_n$ for MDPs with a terminal state, or the following quantity for MDPs without a terminal state.

$$R = \sum_{t=0}^{n} \gamma^t r_t \tag{2.1}$$

Here, $0 \leq \gamma \leq 1$ is the discounting factor. Comprehensive reviews of reinforcement algorithms are presented in [39, 70].

2.5 Summary

We reviewed several key concepts important to this thesis in this chapter. The review topics include multi-agent systems, game theory concepts, and reinforcement learning.

The majority of open distributed computer systems can be modeled as multi-agent systems (MAS) in which each component acts autonomously to achieve its objectives [34]. A key component of multi-agent systems is the interactions that certainly have to take place between agents. These interactions influence cooperation and coordination in agent systems. Furthermore, since agents have incomplete knowledge about their environments and other agents, trust and reputation plays crucial roles in these interactions. We have used multi-agent systems in our proposed environment model as presented in Chapter 4.

As our proposed model (see Chapter 4) utilizes two extensions of the Prisoner's Dilemma, and some game-theoretic notions, we reviewed the necessary game-theoretic concepts and prisoner's dilemma game in this chapter.

Reinforcement learning is deployed in some existing trust and reputation models such as Tran and Cohen [74] and Fullam and Barber [25]. Moreover, our proposed adaptive update mechanism (see Section 5.6) is strongly influenced by reinforcement learning. This usage of RL motivates us to include a brief review of reinforcement learning in this Chapter.

Chapter 3

Related Work: State of The Art

3.1 Introduction

Trust and reputation have been studied and used in various fields from different perspectives. For instance, Nowak and Sigmund have explained why selfish individuals cooperate using reputation concepts [53]. The concept of trust in economics and business was discussed first by Akerlof when he introduced "the market of lemons problem" [2]. He identified certain severe problems of markets characterized by asymmetrical information. Economists have used trust and reputation to explain *irrational* behavior of players in repeated economic games [41, 46]. Computer scientists have used trust and reputation for modeling trustworthiness of entities and individuals in open distributed systems (e.g., online marketplaces, multi-agent systems, and peer-to-peer systems) [58, 52].

The remainder of this chapter proceeds as follows. We present several existing definitions of trust and reputation that have appeared mostly in e-commerce and computer science literature in Section 3.2. Trust model components and various classifications of trust and reputation models are discussed in Sections 3.3 and 3.4 respectively. We present various possibilities for exploiting trust and reputation models in Section 3.5. Reviews of trust and reputation testbeds, and centralized trust and reputation models, are presented in Sections 3.6 and 3.7 respectively. We review decentralized trust and reputation models using one information source in Section 3.8. A review of decentralized trust and reputation models incorporating multiple information sources is presented in Section 3.9. Finally, we summarize this chapter in Section 3.10 with a comparison of decentralized trust and reputation models incorporating multiple information sources.

3.2 Trust and Reputation Definitions

As reputation and trust has recently received considerable attention in different domains such as distributed artificial intelligence, computational economics, evolutionary biology, psychology, sociology, there are many diverse definitions of trust and reputation available in these domains [22]. In the following two subsections, we concentrate on some of the definitions of trust and reputation that have appeared mostly in e-commerce and computer science literature.

3.2.1 Trust Definitions

Unfortunately, the literature on trust and reputation modeling in the e-commerce domain is overwhelmed by many different definitions of trust. Even work presented as trust models may be considered as reputation models by other researchers and vice versa. Herein, we present 5 existing definitions of trust:

1. Mui et al. [51] define trust as "a subjective expectation an agent has about another's future behavior based on the history of their encounters."

2. Grandison and Sloman [28] state that trust is "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context."

3. According to Dasgupta [19],"Trust is a belief an agent has that the other party will do what it says it will or reciprocate, given an opportunity to defect to get higher payoffs."

4. Gambetta [27] defined trust to be "a particular level of subjective probability with which an agent assesses that another agent will perform a particular action, both before the assessing agent can monitor such an action and in a context in which it affects the assessing agents own action."

5. According to Olmedilla et al. [54], "Trust of party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)."

This thesis adopts the term *trust* as defined in 1 above.

3.2.2 Reputation Definitions

While trust definitions focus more on history of agents' encounters and their beliefs, reputation is based on the aggregated information from other individuals. These are 4 important definitions of reputation in the literature:

1. Sabater and Sierra [62] declared that "Reputation can be defined as the opinion or view of someone about something."

2. From the perspective of Abdul-Rahman and Hailes [1], "Reputation is the consequence of word-of-mouth recommendations."

3. According to Mui et al. [51], reputation is the "perception that an agent creates through past actions about its intentions and norms."

4. Castelfranchi et al [16] consider reputation as a component of trust.

Moreover, according to Mui et al. [51], reputation is the basis for the constructing of trust since trust between buyers and sellers can be inferred from the reputation that agents have in the system. Although the first and the second definitions of reputation are consistent, this thesis adopts the term *reputation* as defined in 1 above.

3.3 Trust Model Components

We here explain the components of trust and reputation model to help the understanding of the mechanisms taking place in trust and reputation models.

3.3.1 Roles

As mentioned earlier, trust and reputation models are designed to assist agents in deciding how, when and with whom to interact in a specific context [58]. In other words, an agent must be able to model the trustworthiness of potential interaction partners and make decisions based on that model. In these models, agents might have different roles in modeling of the trustworthiness of a given agent. According to Conte and Paolucci [18], there are four different roles in reputation (and trust) models:

- **Evaluator** is an agent who evaluates the trustworthiness of other agents based on social interactions.
- **Target** is the the agent that is judged.
- **Beneficiary** is the agent that reasons and decides to interact with an agent based on the calculated reputation and trust values.
- **Propagator** is the agent which transmits the reputation information about the target to another agent.

3.3.2 Information Sources

There are different sources of information that an evaluator might use to assess the trustworthiness of a given agent. In other words, agents can provide information to calculate the trust from different sources in their models; according to Sabater and Sierra [63], these sources are:

- **Direct Experiences**: This source, indubitably, is the most reliable information source for trust models. There are two types of direct experiences that an agent can use to calculate trust:
  1. **Direct interaction** which is the experience based on the direct interaction of an agent with a partner.
  2. **Observed interaction** that is the experience based on observed interactions of other members of community.
- **Witness Information**: It is the information that comes from other members of the community. This information can be the result of direct experiences or other gathered information.
- **Sociological Information**: This information is extracted from the social relations between agents in community and their roles in the community.
- **Prejudice**: Calculation of reputation and trust of an agent based on which groups the given agent belongs to. In this sense, prejudice is connected to identifying characteristics of individuals (e.g., skin color or religious beliefs).

All or a subset of these sources of information can be used in trust and reputation models based on the requirements of the system. In this regard, Barber and Kim [7] have compared reputation-based vs. experience-based trust modeling in order to confirm that experience is effective for long term interaction histories whereas reputation gives an accurate picture faster.

Trust and reputation models can be adaptive in choosing the source of information. For example, Fullam and Barber [25] proposed a technique for dynamically learning the best source of information (experience vs. reputation) given the parameters of frequency of transaction with trustee, trustworthiness of trustee and accuracy of provided reputation.

Direct experiences and witness information are in the scope of this thesis while the two others are out of scope.

3.3.3 Interactions

Agents can be in two states in trust and reputation models: *active* (alive) or *passive*. If an agent is passive, it does not participate in any interactions but still has an identity, and exists in the system. In contrast, an active agent interacts with all other active agents or the specific set of them that are the neighbors of the given agent. The neighbors of an agent might be subject to changes over the agent's lifetime. Agents can have various types of interactions with their neighbors to assess the trustworthiness of a given target.

An evaluator agent can assess trustworthiness of a target by using the history of direct interactions of itself with a specific target. When an agent does not have access to the history of direct interactions with the target agent (has never interacted with the target agent), it uses the other information sources for this assessment. The agent can ask about the trustworthiness of the target agent from the set of neighbors (evaluators) which already have assessed the target agent. Those evaluators can provide the asker agent (beneficiary agent) with their own ratings regarding the target. These ratings are called witness information and related interactions are called witness interactions.

The other source of information which can be used in assessment of a given target

agent is the observed interactions. In this light, an evaluator observes the behavior of a target agent in its direct interactions with other agents and judges about the trustworthiness of the given target agent based on those observations. In multi-agent systems, the observation of direct interactions is not straight forward for agents. Consequently, it is not supported by most of the models in the literature. In this thesis, the observation of direct interactions is simulated by letting agents report their direct interactions to each other (see Section 4.2.1). We called this mechanism the Reporting Interaction.

Trust and reputation models might employ incentive mechanisms in order to encourage trustworthy behaviors. In this sense, a trustworthy agent can reward the trustworthy behavior of another one by engaging in a specific interaction. For example, agents can introduce trustworthy agents to each other as a consequence of their trustworthy behavior. We called this mechanism the Introduction Interaction (see Section 4.2.3).

3.3.4 Characteristics

Fullam et al. [26] has brought together the following set of criteria to evaluate trust and reputation models:

- **Multi-dimensional and Multi-faceted:** Trust and reputation models must be able to discriminate between another agent's dynamic trustworthiness characteristics across multiple categories and sources of information.

- **Accurate:** Trust and reputation models should precisely model the agent's behavior.

- **Quickly Converging:** Trust and reputation models must be able to quickly create new models for unknown agents recently entering the system.

- **Adaptive:** The trust value should be adapted if the target's behavior changes.

- **Efficient:** Trust and Reputation models should calculate trust values with minimal computational cost.

3.4 Trust and Reputation Models Classifications

Trust and reputation models have been classified and categorized. This section presents two well-known trust and reputation classification and a functional ontology of reputation in order to provide better insight into trust and reputation models and related works.

3.4.1 Sabater and Sierra Classification

Sabater and Sierra [63] categorize computational trust and reputation models based on the following intrinsic features:

- *Conceptual Models*:
  1. **Cognitive**: These trust and reputation models are created based on underlying beliefs and the mental states of agents.
  2. **Game-theoretical**: These models rely on the result of pragmatic games and numerical aggregation of past interactions.
- *Information Sources*: agents can provide information to calculate the trust from different sources in their models. These sources are: direct experiences, witness information, sociological information and prejudice.
- *Visibility*:
  1. **Centralized:** Trust and reputation of an individual can be seen as a global property available for the entire society.
  2. **Decentralized:** Trust and reputation of an individual can be seen as a subjective property assessed by each individual.
- *Agent Behavior Assumptions*: Models take into account different levels of cheating behavior for agents. In some models, cheating behaviors and malicious individuals are not considered at all whereas in the others possible cheating behaviors are taken into account.

- *Type of Exchanged Information:* Trust models, in terms of the type of information exchanged between individuals for witnesses, can fall into two categories:

    1. **Boolean:** Models working based on probability usually use boolean information.
    2. **Continuous:** Models working based on aggregation mechanism use continuous information.

According to this classification, the scope of this thesis is decentralized game-theoretic trust models incorporating both direct experiences and witness information while considering different possible agent behavior.

3.4.2 Ramchurn et al. Categorization

In the other categorization of trust models presented by Ramchurn et al. [58], trust models are classified into two main groups:

- *Individual-level trust:* An agent has beliefs and understanding about the honesty of its interaction partners and the agent based upon these beliefs will act. There are three sub-categories:

    1. **Evolving and learning strategies:** The agents will learn about the other agents over a number of encounters and interactions.
    2. **Reputation Models:** The agent can reason about the other agent based on the information gathered from the environment, especially by asking for information from other agents.
    3. **Socio-cognitive models of trust:** The agent can characterize the known motivations of other agents.

- *System-level trust:* The agents in the system have to be trustworthy by the rules defined in the system. In other words, the system has been designed such that any interactions of an individual in the system will be reliable. There are three sub-categories in this category:

1. **Trustworthy Interaction Mechanisms:** Design the protocol of interaction such that the participating agents find no benefit by lying or betraying.
2. **Reputation Mechanism:** Developing reputation mechanisms that foster trustworthy behavior. For instance, an agent's reputation as being a liar can be spread by the system.
3. **Distributed Security Mechanisms:** Developing security mechanisms that ensure new entrants can be trusted.

The scope of this thesis is individual-level trust models including both evolving and learning strategies, and reputation models.

3.4.3 Casare and Sichman 's Ontology of Reputation

Sara Casare and Jaime Sichman [14] proposed a Functional Ontology of Reputation (FORe) for agents in order to reach two goals: (1) putting together the broad knowledge about reputation, and (2) representing that knowledge in a common and structured way. FORe declares that reputation is a social product as well as social process. It is a process because it consists of opinion agreement at some levels and a product in the sense that there is a flow of information and influence in the social network. Based on this definition, FORe does not distinguish between trust and reputation and in some parts of the proposed ontology, reputation can be considered as trust in terms of other researchers' definitions of trust.

FORe introduced Reputation Property, Reputation Role and Reputation Process as the main concepts of reputation. The Reputation Property represents two reputation dimensions of Reputation Nature and Reputation Type.

Reputation Nature discriminates reputation according to the nature of the entity whose reputation will be assessed and can have different types of individual, group, product, location, event and activity. On the other hand, **Reputation Type**, which distinguishes a reputation in regards to information source used in its calculation, is classified into *primary reputation* and *secondary reputation.* Primary reputation addresses direct reputation and observed reputation which are calculated based on direct interactions among agents and observations of interactions respectively. Secondary reputation has the categories of propagated reputation, collective reputation

(which is associated with a social group) and stereotyped reputation (based on social prejudice).

The Reputation Process represents the three processes: Reputation Evaluation Process, Reputation Maintenance Process and Reputation Propagation Process. The Reputation Role concept represents those roles played by entities involved in reputation processes such as reputation evaluation and propagation.

Given our chosen definitions of its trust and reputation (referring to Section 3.2), we will not refer to this functional ontology because of incompatibility with those definitions.

3.5 Exploiting Trust and Reputation Models

As highlighted in Section 1.3, we believe that exploitation resistance is a crucial feature of trust models. Exploitation resistance reflects the ability of a trust model to be impervious to agents who try to manipulate the trust model and who aim to abuse the presumption of trust. Exploitation resistance implies that the agents attempting to exploit another agent's trust model know both the details of the model and some or all of the model parameters. If partial knowledge of the parameters is assumed we refer to a system as being p-exploitation resistant. If complete knowledge of the parameters is assumed, we refer to a system as being strongly exploitation resistant, or simply exploitation resistant. In this section, we present the abstract exploitation models that an attacker (attackers) might utilize to mount an attack(s) on trust and reputation models. We put these exploitation models into two categories: *individual attacks* and *collusion attacks.*

We discuss some related works which consider the existence of cheating behavior, inaccurate information and exploitation in subsection 3.5.1. Then, we briefly describe two classes of exploitation models (individual attacks and collusion attacks) in subsections 3.5.2 and 3.5.3 respectively.

3.5.1 Cheaters, Inaccurate Witnesses and Exploitation

Most recently, researchers are attracted to the existence of cheaters (exploitation) in the artificial societies employing trust and reputation models [40], and the existence

of inaccurate witnesses [75, 20, 80]. We will review these works later in this section. Moreover, recently, several trust models have been introduced for distributed infrastructures, especially ad hoc networks, and tested against a small number of attacks. Although these models are out of the scope of this thesis, their examined threat and attack models are relevant to the reported work in this thesis and will be reviewed later in this section.

Smart Cheaters in Marketplaces

Kerr and Cohen [40] examined the security of several e-commerce marketplaces each of which employs a specific trust and reputation system. To this end, they first proposed Proliferation, Reputation Lag, Re-entry, and Value Imbalance attacks and then evaluated them on each marketplace.

In the proliferation attack, the seller simply open a number of accounts, and tries to sell the same product through each of them. As a consequence, the attacker will have more opportunities to sell her product. The Reputation Lag attacker is the seller who behaves honestly for 45 days and cheats for 15 days (the lag before an act of cheating impacts reputation) and then leaves the marketplace. A Re-entry attacker simply opens an account to cheat other agents, then leaves the account to open another. A Value Imbalance attacker is a trustworthy seller on small transactions to gain reputation, but a cheater on the large ones to gain extra profit.

This work measures the vulnerability of a specific marketplace against each of the proposed attacks based on the percentage of monetary profit that strategic cheaters make when compared to an honest sellers' profit. Apparently, the higher this percentage is, the lower is the security of that specific marketplace. However, it is not straightforward to conclude that these vulnerabilities are connected to the environment model and marketplace or the trust and reputation model used. This is mainly because the attacks were not mounted directly against the trust and reputation model but rather mounted against the marketplace embracing the agent policies (behaviors), trust models and rules for participation in that marketplace. Moreover, some proposed attacks (e.g., Proliferation and Re-entry) can not be classified as attacks against trust and reputation models.

Despite the success of the Reputation Lag attack in their experiments, we hypothesize that the following environment model assumption strongly effects this result: "After entering into a sale, a buyer will not know whether or not he has been cheated until after some number of days (14) has passed"[40]. In this sense, the marketplace models are built in a way that the honest buyers do not have enough time to develop their local knowledge regarding the sellers. Unfortunately, Kerr and Cohen assume that buyers are honest in the witness information provided to one another and consequently do not consider collusion attacks.

Coping with Inaccurate Witness information

The general solution to dealing with inaccurate witness information is to ignore or reduce the effect of unreliable opinions. There are two basic approaches to judging the accuracy of opinions. These are referred to as endogenous and exogenous methods by Josang et al. [37]. The former tries to detect unreliable witness information (opinions) by using the statistical properties of the reported opinions; for example, [75, 20]. The latter rely on other information such as the reputation of the source or the relationship with the trustee such as described in the work of Yu and Singh (2003) [80].

Threat/Attack Models in Distributed Infrastructures

Braconnot Velloso et al. [10] present a trust model for ad hoc networks which is robust to Slander attacks. In a Slander attack, a malicious node may collude to lie about the reputation of a particular neighbor. This can cause serious damage to the overall trust evaluation system. Their experiments show that their trust model can tolerate almost 40% of population being liars.

Liu et al. [45] have introduced a trust model for a distributed infrastructure and examined it using the following attacks: independent bad mouthing attack, collaborative bad mounting attack, and the conflict behavior attack. In an independent bad mouthing attack, the bad node attempts to attack the trust system by giving negative trust values to the good nodes whereas in a collaborative bad mounting attack, the bad nodes cooperate by rating each other highly positive and giving low ratings for

the good nodes. In the conflict behavior attack, the malicious node tries to develop opposite opinions between two subsets of nodes by behaving well to a subset and behaving badly to the other subset.

3.5.2 Individual Attacks

In individual attacks, an attacker usually takes the advantage of the existing vulnerability in the trust models to cheat other agents without the system preventing it. This type of attacks is mounted by only one attacker against another agent or a set of other agents and usually takes place in direct interactions.

A con-man attack presented in this thesis (see Section 5.3.1) is an example of the individual attacks. What the con-man attacker does is to build up trust from the victims view point by being honest with him/her in several direct interactions. Then, when it comes to a high risk interaction, the con-man will cheat on the victim. The con-man, by regaining the victims trust, can again cheat on the victim. Consequently, behaviour in which cycles of positive feedback followed by a single negative feedback results in untrustworthy agents remaining undetected in vulnerable trust models. In contrast, if the trust and reputation model is **con-resistant** (see Section 5.4.1 for a formal definition), this type of behavior is detected as untrustworthy behavior and the con-man will be penalized.

3.5.3 Collusion Attacks

Generally speaking, collusion can be defined as collaborative activity that gives to members of a colluding group benefits they would not be able to gain as individuals.

In contrast to individual attacks discussed above, collusion attacks are where a group of agents (at least two agents) conspire together to take advantage of breaches in trust models to defraud a specific agent or a set of agents. Even one or some of the agents can sacrifice themselves in collusion attacks in order to maximize the utility of the colluding group.

Collusion attacks usually work based on the basic idea that one or more agents show themselves as trustworthy agents in one type of interaction (usually direct interaction). Afterward, they will be untrustworthy in other type of interaction (e.g.,

witness interaction) by providing false information in favor of other members of the colluding group. This false information usually encourages a victim to interact with members of the colluding group. The members of the colluding group will cheat the victim, if victim interacts with them.

Malicious witnesses (Witness-based Collusion Attack) is an example of a collusion attack. Malicious witnesses aim to trick agents into believing they are trustworthy while providing high ratings for malicious agents (other members of the colluding group) in order to encourage the asker agent to interact with them, and consequently it will be exploited by them.

3.6 Trust and Reputation Testbeds

Open distributed systems can be modeled in open multi-agent systems that are composed of autonomous agents that interact with one another using defined policies. We here describe some existing testbed environments in which agents' interactions with their peers take place. It is worth mentioning that despite the fact that many trust models have been recently proposed, a general trust evaluation testbed does not exist. In this sense, we are interested in the testbeds which do not have any barriers for real world implementation and still are generic.

3.6.1 Iterated Prisoner's Dilemma

Iterated Prisoner's Dilemma (IPD) [5] (referring to Section 2.3.4) has been used as a testbed for evaluation of trust and reputation models and strategies [66, 52, 51, 5]. Utility is a metric used for the comparative assessment of trust and reputation models.

Although IPD is suitable for direct interaction modeling, it has several shortcomings for trust and reputation modeling: First, as agents evaluate one aspect of opponents' behavior, multidimensional trust modeling is not encouraged. Second, agents cannot separate untrustworthy agents because they have to interact with all other agents. Third, it suffers from the lack of system-level metrics and only focuses on total utility of the agent. In Chapter 4, we will discuss how our proposed game-theoretic testbed overcomes these drawbacks while maintaining the simplicity of the Iterated Prisoner's dilemma.

3.6.2 SPORAS Testbed

The SPORAS experiments [81] have been widely used for the evaluation of trust and reputation models. For instance, Regret [62], AFRAS [13, 60] have used this set of experiments. The SPORAS experiments evaluate reputation models by measuring the time taken for them in electronic marketplaces to converge to true reputations. However, these experiments suffer from the following shortcoming. First, the experiments only employ one single-agent metric and ignore the system-level metrics. Second, they do not consider multi-dimensional trust models. Third, while this experiment set focuses on trust model accuracy, and adaptivity, it ignores the agent's capability in making decisions based on trust, such as determining whether or not to disconnect from untrustworthy agents. The consideration of disconnection/connection for agents is important as human society possesses this capability.

3.6.3 ART

Fullum et al. [26] introduced the Agent Reputation and Trust (ART) Testbed which serves two roles: (1) as a competition platform in which researchers can compare their trust and reputation models against objective metrics, and (2) as a suite of flexible tools, allowing researchers to perform experiments.

In ART, agents use trust strategies to exchange expertise with others to appraise paintings. Agents make money by appraising the paintings while more accurate appraisal results in better income. Furthermore, those agents that appraise a painting more accurately will receive more paintings to appraise in the future. However, agents' expertise is limited to appraising a subset of the paintings. They are required to exchange expertise with other trustworthy agents. Moreover, agents can also exchange trust values with others (Witness Interaction). The agent who has the most money at the end wins the game.

To evaluate our proposed trust and reputation model in ART, we had to consider many variables that are not covered in ART. Our trust variables are built from a number of positive and negative outcomes and unfortunately, in ART, agents can exchange only a probability. The goodness of a trust model according to ART depends on the agent's bank balance, which involves many variables not covered in our model.

Most importantly, one of our main contributions is that our model provides Reporting Interaction and Introducing Interaction and the corresponding dimensions of trust for agents. Unfortunately, ART does not support any of them. Moreover, modeling certain exploitations (e.g., colluding) is hard in ART since each trust strategy controls a single agent, which works in competition against every other agent in the system. Unfortunately, ART suffers from a lack of tools and metrics for demonstrating the structure of the social network. There is no limitation in ART regarding agent communication (lack of privacy consideration) as opposed to our model in which only neighbor agents are allowed to interact and communicate with each other. Therefore, in order to evaluate our model, we design our own testbed as described in Chapter 4.

3.7 Centralized Trust and Reputation Models

The body of research on trust and reputation models is large; a review of which can be found in [58, 4, 63, 37]. In this section, we limit our discussion to some popular centralized models, while in Section 3.9 we will concentrate on the decentralized trust models incorporating multiple information sources which are the main focus of this thesis.

3.7.1 State of Technology: Practical Implementation

Amazon [3] and eBay [21] are important practical examples of centralized reputation management systems. In these systems, the sellers list their items for sale and buyers bid for items. Users are allowed to rate and submit textual comments. The overall reputation of a seller is the average of the ratings obtained from his customers.

For instance, eBay is an online auction and shopping website in which people and businesses buy and sell goods and services worldwide. In eBay, sellers receive feedback (+1, 0, -1) in each auction and their reputation is calculated as the sum of those ratings over the last six months [59]. Certain research has postulated that seller reputation has significant influence on prices, especially for high-valued products in eBay market [30, 59].

Page et al. [12] proposed PageRank which represents a way of ranking the best search results based on a page's reputation. Generally speaking, PageRank ranks a

page by considering how many other pages have links to it (point at it). This can be seen as a reputation system where the collection of hyperlinks to a given page can be seen as positive feedbacks yielding a reputation score. Google's search engine is based on the PageRank algorithm and the rapidly growing popularity of Google was obviously the consequence of the superior search results that the PageRank algorithm delivered.

3.7.2 Sporas and Histos

Zacharia, et al., have suggested *Sporas* and *Histos* systems for reputation management [82, 83]. Reputation in Sporas extends the reputation management systems used in eBay and Amazon by introducing a new method for rating aggregation. Briefly, once a rating is received it updates the reputation of the involved party with a special algorithm instead of storing all ratings and calculating average. However, Sporas is not suitable for open distributed systems because of its centralized design. Moreover, the Sporas experiments neither account for multidimensional trust, nor do they measure an agent's ability to make trust-based decisions, leading to isolation of untrustworthy agents.

Histos was developed to compensate for the lack of personalization that Sporas reputation values dealing with. This model covers direct interaction and witness information where a value for reputation is assigned by each individual. The main weakness of this model is the simultaneous use of the reputation value of an individual also as reliability of the provided information by that agent. If the agent is reliable in direct interaction, it does not mean that it has to be also a trustworthy witness. As a result, Sporas is vulnerable to collusion attacks.

3.7.3 Beta Reputation System

The Beta Reputation System (BRS) [36] is a probabilistic trust model and works based on the beta distribution. The system is centralized and designed to meet the requirements of online communities. In BRS, users rate the performance of other users by providing either negative or positive feedback. The feedback values are then used to calculate shape parameters of the user's reputation. In other words, the beta

distribution takes two parameters: a count of past honest (positive) interactions (feedback) and a number of past dishonest (negative) interactions (feedback). However, BRS does not show how it is able to cope with misleading (inaccurate) information. Whitby et al. [75] extend BRS and show how it can be used to filter out unfair or inaccurate ratings by using an endogenous method, where the agent's ratings are discarded if they are statistical outliers. However, their approach is only effective when a significant majority of available reputation sources are fair and accurate.

3.8 Decentralized Trust Models Using One Information Source

In this section, we limit our discussion to decentralized models that incorporate one information source (usually direct experience information). These models are not the main concern of this thesis as the main scope of this thesis is decentralized trust models incorporating multiple information sources (see Section 3.9).

3.8.1 Marsh

Steve Marsh was among the first to introduce a computational trust model for a distributed artificial intelligent society. His model attempted to integrate aspects of trust taken from sociology and psychology. His model takes into account only an agent's own experiences (direct interaction) while differentiating three types of trust: Basic Trust, General Trust and Situational Trust [47]. Since Marsh's model is based on sociological foundations, the model is too complex to be easily used in today's multi-agent systems. Moreover, the model only considers an agent's own experiences and does not involve any social mechanisms. Hence, a group of agents can not collectively build up a reputation for others. All agents can interact with each other without any constraint. We consider this lack of a neighborhood to be significant limitation of this work.

3.8.2 Tran and Cohen

Tran and Cohen [74] proposed a marketplace model and learning algorithms for buying and selling agents in electronic marketplaces. By considering the possible existence

of dishonest selling agents in the market, learning agents employed a trust model to distinguish untrustworthy agents and prevent from interacting with them. This work is representative of a direct experience model: agents make use only of their own experience in evaluating the trustworthiness of others. Each agent maintains pairs of expected outcomes and possible actions and then selects an action among the possible actions in order to maximize the expected value. After an action is taken, the real outcome is used to update the expected outcome for that action. Gradually, the buyer will learn which agents are trustworthy to interact with for a given product.

3.8.3 Mui et al.

Mui et al. [52] discuss the strength of the various notions of reputation using a simple simulation working based on evolutionary game theory. This work focuses on the strategies of each agent only for direct interaction, and do not consider gathering reputation information from other parties in the network or any other social mechanism. Moreover, they review existing works on reputation among diverse domains such as distributed artificial intelligence, economics, and evolutionary biology.

3.9 Decentralized Trust Models Using Multiple Information Sources

As mentioned before, the body of research on trust and reputation models is large; a review of which can be found in [58, 4, 63, 37]. Here we limit our discussion to decentralized models that incorporate multiple information sources or express the importance of doing so. Moreover, we will discuss the cons and pros of them.

3.9.1 Yu and Singh

Bin Yu and Munidar P. Singh developed an approach for social reputation management in which they represented an agent's ratings regarding another agent as a scalar and combined them with testimonies using combination schemes similar to certainty factors [78] (for our convenience, we refer to this model as YS2000). As we have used the direct trust component of this model, we here explore this component further.

YS2000's trust variable is defined by $T_{i,j}(t)$ indicating the trust rating assigned

by agent i to agent j after t interactions between agent i and agent j, while $T_{i,j}(t) \in [-1, +1]$ and $T_{i,j}(0) = 0$. One agent in the view of the other agent can have one of the following levels of trustworthiness: *Trustworthy*, *Not Yet Known*, or *Untrustworthy*.

For agent i, an upper threshold ($-1 \leq \omega_i \leq 1$) and a lower threshold ($-1 \leq \Omega_i \leq 1$) are defined to model different levels of trustworthiness. Agent j is trustworthy from the viewpoint of agent i after t times of interactions if and only if $T_{i,j}(t) \geq \omega_i$. Agent i sees agent j as an untrustworthy agent if $T_{i,j}(t) \leq \Omega_i$ and if $\Omega_i < T_{i,j}(t) < \omega_i$ then the agent j is neither considered trustworthy nor untrustworthy (Not Yet Known) in agent i 's view. An agent will update this variable based on the perception of cooperation/defection. Cooperation by the other agents generates positive evidence of $\alpha > 0$ and defection generates negative evidence of $\beta < 0$. The following trust updating scheme is proposed by [78]:

$T_{i,j}(t+1) =$

$$\begin{cases} T_{i,j}(t) + \alpha(1 - T_{i,j}(t)) & T_{i,j}(t) > 0 \text{ , } Cooperation \\ (T_{i,j}(t) + \alpha)/(1 - min(|T_{i,j}(t))|, |\alpha|) & T_{i,j}(t) < 0 \text{ , } Cooperation \\ (T_{i,j}(t) + \beta)/(1 - min(|T_{i,j}(t))|, |\beta|) & T_{i,j}(t) > 0 \text{ , } Defection \\ T_{i,j}(t) + \beta(1 + T_{i,j}(t)) & T_{i,j}(t) < 0 \text{ , } Defection \end{cases} \tag{3.1}$$

The drawbacks of the combination model of YS2000 led Yu and Singh to consider an alternative approach [79] (for our convenience, we refer to this work as YS2002); specifically, an evidential model of reputation management based on the Dempster-Shafer theory [1]. This model represents the agent's belief (probability) that a partner will cheat, and the probability that it will not cheat. Moreover, the model also explicitly represents the agent's lack of belief in those outcomes. In this model, an agent relies on its own experience if it is sufficient. If not, it asks for the opinions of others using a "TrustNet". An agent can solicit information from its neighbors when needed. If the neighbor cannot provide information, it may refer the agent to one of its own neighbors. Actually, they use direct information and witness information while not combining these two types together. In this model, there are two kinds of information that a witness can provide when it is asked about another agent: 1)

[1] Dempster-Shafer Theory [42] is founded on the fact that there is no causal relationship between a hypothesis and its negation. In this light, lack of belief does not mean disbelief and reflects a state of uncertainty.

rating about the queried agent if it is the neighbor of the given agent, or 2) referral to another agent. However, malicious witnesses and collusion attacks are not considered in either of these two proposed models. Moreover, both are vulnerable to the con-man attack.

In another work, Yu and Singh studied the problem of deception in reputation management [80] (for our convenience, we refer to this work as YS2003). Through the introduction of models of deception, the work proposed an approach to detecting the deceptions which follow those models. The approach involves an application of the weighted majority algorithm (WMA)[2] to the belief function and their aggregation. In the proposed model, the agents exchange witness information in the form of belief functions. Unfortunately, it is unclear how agent's exchange the belief functions. They consider three kinds of deception for witness providers: complementary, exaggerated positive and exaggerated negative. This work assumes that witness providers behave in a consistent manner. This consistency is in the terms of strategy and interacting partners. For example, a witness provider with a complementary strategy always returns the complement of ratings to all other agents.

Yu et al. have proposed the trust model in large-scale peer-to-peer systems in which each peer has its own a set of acquaintances [77] (for our convenience, we refer to this work as YSS2004). A subset of these acquaintances is identified as its neighbors. A peer maintains a model of each acquaintance. The acquaintance's reliability and credibility are included in this model. Reliability is used for providing high quality services while credibility is used for providing trustworthy ratings to other peers. The weighted majority algorithm (WMA) is adapted to predict the trustworthiness of an agent based on the set of testimonies from the witnesses. The focus of this work is more in peer-to-peer systems and it does not model the reporting interaction and introduction interaction.

[2]The weighted majority algorithm (WMA) [44] is designed to improve the predictions based on a set of advisers. It assigns weights to the advisers and makes a prediction based on the weighted sum of the ratings provided by them. The weights are tuned after each successful prediction such that the relative weights assigned to the successful advisers are increased and the relative weights assigned to the unsuccessful advisers are decreased.

3.9.2 Mui et. al

Mui et al. [51] have proposed probabilistic models for reputation which use Bayesian statistics. Reputation for an agent is inferred based on propagated ratings from an agent's neighbors (for our convenience, we refer to this work as Mui2002). In their probabilistic trust model, they show that if the number of interactions is too low then trust cannot be built. They calculate the probability of an agent being trustworthy on the next interaction by considering the frequency of positive and negative direct impressions gathered from the social network. This work does not take into account witness-based collusion and is vulnerable to the con-man attack.

3.9.3 Jurca and Faltings

Jurca and Faltings [38] introduce a reputation mechanism in which agents report truthfully about their interactions' results to the set of broker agents called R-agents (for our convenience, we refer to this work as JF2002). R-agents specialize in buying and aggregating reports from other agents and selling back reputation information to them when they need it. The reputation for a specific agent is simply calculated by averaging the reports related to that agent.

In spite of a distribution of R-agents in the system, the reputation mechanisms should not be regarded as completely decentralized mechanisms because regular agents are still dependent on R-agents for acquisition of a specific agent's reputation. Although a payment scheme for reputation reports is proposed, motivating agents to share their reports truthfully, this method does not work if most agents lie about the reports or if they collude in giving false reports. In these scenarios, the reputation score will be incorrect since the trustworthiness (reliability) of the reporter is not taken into account where the reputation is calculated by simple averaging of reports in this model. As a result, the model is vulnerable to collusion attacks. Moreover, newcomers are not modeled when there is an assumption that information agents already store some reputation information. This assumption is the result of one rule of the system, allowing agents to sell a report for an agent when they have previously bought reputation information for that agent. Direct interaction and witness interaction are also not addressed in this model.

3.9.4 Social Interaction Framework (SIF)

In the Social Interaction Framework (SIF) [66], agents are playing a Prisoner's Dilemma set of games with a partner selection phase. Each agent receives the results of the game it has played plus the information about the games played by a subset of all players (its neighbors). An agent evaluates the reputation of another agent based on observations as well through other witnesses. However, the reporting component of this work is completely centralized as opposed to our requirement for a decentralized reporting component. Moreover, there is an assumption that reported interactions are not manipulated or spurious. As a result, it is vulnerable to collusion attacks. The SIF does not describe how to find witnesses, whereas in electronic communities deals are broken among people who often would never have met each other. Moreover, it is also vulnerable to con-man attack.

3.9.5 Sen and Sajja

Sen and Sajja [68] model reputation using both direct interaction and observed interaction (for our convenience, we refer to this work as SS2002). Observations are noisy with noise modeled using a Gaussian distribution and may differ from the actual performance. One trust variable is considered for both sources of information and reinforcement learning is used to update the value of that variable. Due to the noise in observations, the rule used to update the reputation value for direct interaction has a greater effect than the rule used to update the value for an observation.

Agents can query other agents about the performance of a given agent and the response is a boolean value that says if the partner is good or not. The subset of agents to be queried is selected randomly from the set of possible witnesses. In this model, although the existence of liars is assumed, the liar should lie consistently and the number of them should be less than half of the population of agents. Agents only use witness information to make decisions while direct experiences are only used as pieces of information to be communicated to others. However, this model is vulnerable against collusion attacks since it uses the same trust variable for observations and direct interactions. The reporting mechanism is centralized and the amount of

observational noise is not defined clearly. Moreover, each agent should have a priori knowledge of the percentage of liars in the population to calculate the necessary numbers of witness queries. This work focuses more on calculation of the number of witnesses in order to get rid of liars' information instead of aggregation of witness information with the existence of liars.

3.9.6 Regret

Regret [62] is a decentralized trust and reputation system designed for e-commerce environments. The system takes into account three different sources of information: direct experiences, information from third party agents and social structures. The direct trust, witness reputation, neighborhood reputation and system reputation are introduced in Regret where each trust and reputation value can has an associated reliability measure. This measure tells the agent how confident the system is regarding that value according to how it has been calculated. The reliability value is calculated from the number of ratings taken into account in producing the trust values and the deviation of these ratings. However, this model still suffers from the malicious witness providing false reputation and as a result is vulnerable to collusion attacks. Moreover, except the direct trust component, the rest of the model is not readily applicable because it is not obvious how each agent can build the social network on which Regret depends. Unfortunately, this model is vulnerable to con-man attack as well.

Since we have used the direct trust component of Regret in our experiments, we will explain it herein with more detail. Regret uses the term subjective reputation (direct trust) to talk about the trust calculated directly from an agent's impressions. Regret defines an impression as the subjective evaluation made by an agent on a certain aspect of an outcome. $w_{i,j}(t) \in [-1, 1]$ is the rating associated with the impression of agent i about agent j as a consequence of specific outcome at time t. $W_{i,j}$ is the set of all $w_{i,j}(t)$ for all possible t. A subjective reputation at time t from agent i's point of view regarding agent j is noted as $T_{i,j}(t)$[3]. To calculate $T_{i,j}(t)$, Regret uses a weighted mean of the impressions' rating factors, giving more

[3]For the purpose of simplification, we have changed the original notation from [62].

importance to recent impressions. Intuitively, a more recent rating is weighted more than those that are less recent. The formula to calculate $T_{i,j}(t)$ is:

$$T_{i,j}(t) = \sum_{r_k \in W_{i,j}} \rho(t, t_k).r_k \tag{3.2}$$

where t_k is the time that w_k is recorded, t is the current time, $\rho(t, t_k) = \frac{f(t_k,t)}{\sum_{r_l \in W_{i,j}} f(t_l,t)}$, and $f(t_k, t) = \frac{t_k}{t}$ which is called the rating recency function.

3.9.7 FIRE

Huynh et al. proposed a trust and reputation model called FIRE that integrates a number of information sources to estimate the trustworthiness of an agent [31, 32]. Specifically, FIRE incorporates interaction trust, role-based trust, witness reputation, and certified reputation to provide a trust metric. The interaction trust and witness reputation are the result of the past experience of direct interaction and reports of witnesses about an agent's behavior respectively. Role-based trust is defined by different role-based relationships between agents whereas certified reputation is built from the third-party references which are provided by agents themselves. There are two assumptions in FIRE which makes it vulnerable to collusion attacks: (1) Agents have a tendency to share their experiences with one another, and (2) Agents are honest in exchanging information. In other words, FIRE does not consider the existence of malicious witnesses or reporter in its environments and consequently colluding is not considered. Moreover, the interactions between agents are not confined to an agent's neighborhood as any agent can interact with any other. Unfortunately, FIRE is vulnerable to con-man attack.

We herein explain in-detail direct trust components of FIRE, as we have used it in the experiments reported in this thesis (see Section 6.3.1). FIRE utilizes the direct trust component of Regret but does not use the rating recency function of Regret, the method used to calculate the weights for each rating. The rating recency function of Regret has a shortcoming regarding time granularity control and does not actually reflect a rating's recency. Consequently, FIRE introduced a new rating recency function based on the time difference between current time and the rating

time. The parameter λ is introduced in that rating recency function to scale time values. As a result, this parameter makes rating recency function adjustable to suit the time granularity in different applications. FIRE's rating recency function is given by the following formula:

$$f(t_k, t) = e^{-\frac{t-t_k}{\lambda}} \tag{3.3}$$

3.9.8 TRAVOS

TRAVOS [71, 72] is a probabilistic trust model that is built based on observations of past interactions between agents. Trust is calculated using probability theory and takes into account the past interactions and reputation information gathered from third parties while coping with inaccurate reputations. There are two assumptions in TRAVOS which are included in our proposed model as well: (1) agents may be self-interested and may provide false accounts of experiences with other agents, and (2) agents will need to interact with unvisited agents. TRAVOS simplifies the outcome of an interaction by providing a binary rating, where 1 and 0 represent successful and unsuccessful interactions respectively.

TRAVOS utilizes the beta family of probability density functions (PDF) to model the probability of having a successful interaction with a particular given agent. This probability represents the agent's trust value. Moreover, using PDFs, TRAVOS estimates the confidence of its trust values. If the confidence level of a trust value is low, TRAVOS will ask for witness information about the target agent's past performance from all other agents. A witness agent provides the witness information in the form of a pair consisting of the numbers of its successful and unsuccessful interactions with the target agent.

After the beneficiary agent (asker agent) interacts with the target agent, it will compare received witness information with its own observations. Afterward, the agent calculates the probability that the witness's information is compatible with the true behavior of the target agent within a reasonable margin of error. The calculated probability is used to weight the impact of the witness' opinions on future decisions.

TRAVOS, similar to our proposed model, filters out unfair opinions but there are a number of assumptions that make our proposed model different with it. First,

TRAVOS does not assess trustworthiness of other agents in terms of providing witness information while only decreasing the effect of unfair opinions. Second, TRAVOS assumes that the behavior of agent does not change over time which is, in many cases, an unsafe assumption. For example, a con-man changes its behavior several times. Third, in TRAVOS, every agent can communicate with, interact with and rate other agents; therefore, there is no social network of agents. This assumption is not compatible with the nature of open distributed systems in which it is not necessarily true that every entity with every other entity. Fourth, TRAVOS does not clarify what agent strategies are and how they use TRAVOS to decrease the risk of the interaction with untrustworthy agents. Fifth, TRAVOS does not employ any reporting mechanism and consequently does not consider any dimension of trust for it.

3.10 Summary

This chapter begins by reviewing trust and reputation definitions. Trust and reputation components including roles, information sources, interaction, characteristics are then explained thoroughly. Two different classifications of trust and reputation models along with a functional ontology of reputation are discussed. Afterward, two categories of exploitation models: individual attacks and collusion attacks are described. The shortcomings of existing trust and reputation testbeds are then addressed. We reviewed the state of the art for trust and reputation models while emphasizing decentralized models incorporating diverse sources of information. We discuss the pros and cons of each of them and a summary of this review is presented in Table 3.1.

In the next chapter, we will present our proposed environmental model (testbed) which addresses many of the limitations of the models reviewed in this chapter.

Witness Information	WINFO
Observed Interaction	OI
Direct Interaction	DI
Witness Interaction	WI
Reporting Interaction	RI
Introduction Interaction	II
Vulnerabilities	VUL
Multi-dimensional Trust Model	MTM
Collusion Attacks	COAT
Con-man Attack	CA
Social Network	SO-NET
Yes	✓
No	×
With Some Constraints	WCS
Not Applicable	NA

Name	Information Sources	Interactions	VUL	SO-NET	MTM
YS2000	DI+WINFO	DI+WI	COAT,CA	×	×
YS2002	DI+WINFO	DI+WI	COAT,CA	✓	×
YS2003	DI+WINFO	DI+WI	CA	✓	✓
MUI2002	WINFO	DI+WINFO	COAT,CA	✓	×
JF2002	WINFO	WI+RI†	COAT,CA	×	×
SIF	DI+OI+WINFO	DI+WI+RI†	COAT,CA	×	×
SS2002	DI+OI+WINFO	DI+WI	COAT,CA	×	×
Regret	DI+WINFO	DI+WI	COAT,CA	×	NA
FIRE	DI+WINFO	DI+WI	COAT,CA	×	×
TRAVOS	DI+WINFO	DI+WI	CA	×	×

Table 3.1: Summary of reviewed trust and reputation models
† This report interaction is not distributed and applicable for distributed systems

Chapter 4

The Environment Model of ERT

4.1 Introduction

As stated in Chapter 1, the majority of open distributed computer systems can be modeled as multi-agent systems (MAS) in which each component acts autonomously to achieve its objectives [34]. An important characteristic of many of these systems is that they are *open* in terms of agents joining and leaving the system. Huynh et al. [32] pointed out three interesting features of these systems: (1) the agents are likely to be self-interested and may be unreliable and (2) no agent can know everything about its environment. In other words, there is no global perspective and (3) no central authority can control all the agents due to different ownership. A key component of these open MAS is the interactions that certainly have to take place between agents. This chapter explains the main idea and principles behind our proposed multi-agent environment (the environment model of ERT) in which agents' interactions with their peers take place.

The environment model of ERT is designed to be consistent with the nature and characteristics of open distributed systems. The proposed environment model follows three features of open distributed systems as described by Huynh et al. [32]. In the proposed model, heterogeneous agents with various perceptive capabilities and decision making interact in a game theoretic manner. This environment model can be viewed of as an undirected dynamic graph with nodes of agents. An edge between two nodes (agents) in this graph indicates that these agents have interactions together and that they can communicate with each other.

The remainder of this chapter proceeds as follows. The different interaction types of agents in ERT are explained in Section 4.2. The extensions of the Prisoners Dilemma which are used in interaction modeling are discussed in Section 4.3. Different kinds of cooperation/defection are described in Section 4.4. Protocols for possible

interaction and connection/disconnection of agents are detailed in Sections 4.5 and 4.6 respectively. We describe the registry list, agent type specification and newcomer modeling in Sections 4.7, 4.8 and 4.9 respectively. Before presenting the summary of the chapter in Section 4.11, we present the metrics that our proposed environment is equipped with in Section 4.10.

4.2 Interactions

An agent interacts with the specific set of other agents that are the neighbors of the given agent. Two agents are *neighbors* if they interact with one another continuously. An agent maintains the *neighborhood* set which contains the name (unique ID) of its neighbors. The *neigborhood* set is a dynamic set, subject to changes over the agent's lifetime (i.e., the agent drops or adds connections). These changes are based on the results of the agent's interactions. Agents can have four types of interactions with their neighbors: Direct Interaction, Observed Interaction, Witness Interaction and Introduction Interaction.

4.2.1 Direct Experience Interactions

Direct experience, incontrovertibly, is the most popular source of information for trust and reputation models [63, 58]. There are two types of direct experiences that an agent can infer agents' trustworthiness from: *Direct Interaction* and *Observed Interaction.*

Direct Interaction. Different fields have their own interpretation and understanding of direct interaction. In the context of e-commerce, direct interaction might be considered as buying or selling a product whereas in peer-to-peer systems (e.g., file sharing systems) direct interaction is uploading or downloading files. Providing a service and consuming a service can be regarded as a direct interaction in the context of web services while asking a question (sending a query) and answering that question (receiving the result of that query) is a direct interaction from the perspective of information retrieval.

Observed Interaction. Agents can judge the trustworthiness of another agent by relying on the observation of the given agent's interactions with other agents in

the community. This source of information is not common in decentralized trust and reputation models because of the existing limitation for observing interaction in open distributed systems. For example, the observation of interactions in a peer-to-peer system, which is an open distributed system, for each peer is not straightforward.

Social learning theory leads us to hypothesize that observation of interactions can be considered one of the main information sources for learning of trustworthiness of other agents. Observational or social learning is based primarily on the work of Albert Bandur and his colleagues who showed that learning could occur through the simple process of observing someone else's activity. Consequently, people learn through observing others' behavior [56].

To provide the observation of interactions for agents and resolve the aforementioned limitation in open distributed systems, the neighbors of an agent report their direct interactions with their own neighbors to their immediate neighbors. In this sense, ERT has a decentralized system of news broadcasting that is consistent with its decentralized nature and provides the facility for agents to have social (observational) learning. From now on, we call this type of interaction by which agents report their direct interactions to their neighbors the ***Reporting Interaction***.

4.2.2 Witness Interaction

Witness information is information that comes from other members of the community regarding another agent. This information is provided in the form of a rating that can be based on observed interaction or direct interaction. An agent can ask for an assessment of the trustworthiness of a specific agent from its neighbors and then the neighbors send their ratings of that agent to the asking agent. We call this asking for an opinion and receiving a rating, a ***Witness Interaction***.

4.2.3 Introduction Interaction

We are proposing a novel type of interaction for agents in ERT model called an *Introduction Interaction*. Agents can introduce or recommend one of their neighbors to the other one by using an Introduction Interaction. This introduction/recommendation can be *request-driven* or *asynchronous*.

In the request-driven scenario, an agent will make a request for a connection recommendation to one of its neighbors, and then, in the response, the neighbor introduces a new agent to the requester. In contrast, an asynchronous introduction is solely based on the decision of the recommender. For instance, after an agent is known as a trustworthy agent from the perspective of a neighbor as a consequence of direct interactions, the neighbor might introduce the agent to one of the other trustworthy agents. In this way, the trustworthy agent can extend its neighborhood by adding new trustworthy agents. This introduction can be an incentive for agents to be trustworthy to their peers in order to be introduced to more trustworthy agents in the community. Our motivation for proposing this interaction is the existence of systems such as PGP.

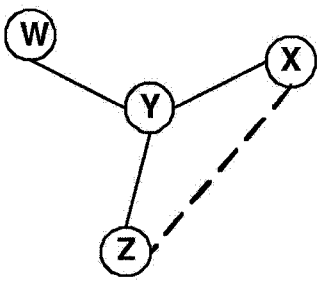


Figure 4.1: An example for introduction of an agent to another

For instance, as shown in Figure 4.1, after several direct interactions of agent X with agent Y, if agent X is known to be trustworthy from the viewpoint of agent Y, agent Y might introduce agent X to one of its other trustworthy neighbors; let us say agent Z.

4.3 Games: IPD and GPD

We have modeled interactions in the ERT environment using two extensions of the Prisoners Dilemma: Iterated Prisoner's Dilemma (IPD) and Generalized Prisoner's Dilemma (GPD).

As explained in Section 2.3.3, the Prisoner's Dilemma forms a non-zero-sum, non-cooperative and simultaneous game in which two players may each cooperate with or defect from the other player. Similar to other games in game theory, the goal of

each individual is maximizing his/her payoff, without any concern for other player's payoff. In this game, cooperating is strictly dominated by defecting since the game will only be played once between individuals.

In contrast, since the game is played repeatedly in the Iterated Prisoner's Dilemma [5], each player has an opportunity to "punish" the other player for previous uncooperative play. As a result, cooperation might emerge as an equilibrium outcome. The Iterated Prisoner's Dilemma is closely related to the evolution of trust because if both players trust each other they can both cooperate and prevent mutual defection. Moreover, this trust can only build up in the environment where individuals have to interact with each other repeatedly.

We have modeled direct interactions using an iterated prisoner's dilemma game. Each agent plays one game with each of its neighbors in each cycle of simulation.

The Generalized Prisoner's Dilemma (GPD) is a two-person game which specifies the general forms for an asymmetric payoff matrix that preserves the social dilemma. GPD is compatible with client/server structure where one player is the client and the other one is the server in each game. It is only the decision of the server which determines the ultimate outcome of the interaction. Note that, a player can be a server in one game and a client in another [23].

We used GPD to model witness, reporting, and introduction interactions because these interactions are compatible with the nature of client/server structure. For example, in a witness interaction, the asker agent is a client while the witness information provider is a server for that request and in a reporting interaction the reporter is the server whereas the listener is a client.

4.4 Cooperation and Defection

We define different kinds of **Cooperation** and **Defection** in the ERT model. There are 4 types of cooperation and defection:

- Cooperation/Defection in Direct Interaction (CDI/DDI)
- Cooperation/Defection in Reporting Interaction (CRI/DRI)
- Cooperation/Defection in Witness Interaction (CWI/DWI)

- Cooperation/Defection in Introduction Interaction (CII/DII)

4.4.1 CDI/DDI

Cooperation/Defection in Direct Interaction (CDI/DDI) have different interpretations depending on the context. In the context of e-commerce, defection in an interaction can be interpreted as that the agent does not satisfy the terms of a contract, sells poor quality goods, delivers late or does not pay the requested amount of money to a seller depending on the role of the agent [58]. Therefore, defection could get higher payoffs for the agent defecting and cause some utility loss for the other agent. In contrast, if both interaction participants cooperate, they will get higher payoff in the long term [5].

Cooperation in peer-to-peer systems (e.g., file sharing) might mean allocating high bandwidth for uploading files while defection might be considered as low bandwidth allocation for uploading. In the context of information retrieval, defection in an interaction can be interpreted as that the queried agent returns irrelevant documents to the asking agent as the consequence of its query. In contrast, cooperation means that a proper answer is provided according to the query for the questioner.

Cooperation and defection may have their own interpretation in the domain of the web services. Generally, the cooperative service provider prepares the desirable service for a consumer, subject to the set of consumer constraints. By contrast, defection is the outcome of preparing a low quality and undesirable service.

4.4.2 CWI/DWI

As explained in Section 4.2, an agent can ask for an assessment of the trustworthiness of the specific agent from the perspective of other agents. In this sense, the witness agent can provide honest ratings of the agent or a false rating of the agent. Even a witness agent can hide its rating from an asking agent and might pretend not to have any relevant information. Therefore, the asking agent may encounter two types of response behavior from a witness agent: (1) cooperation, or (2) defection. We define Cooperation/Defection in the context of Witness Interaction (CWI/DWI) as follows:

Definition: Cooperation in a witness interaction (CWI) means that the witness

agent will provide a reliable and honest rating for the asker agent regarding the queried agent. In contrast, defection in a witness interaction means that the witness agent does not provide a reliable and honest rating for the asker agent regarding the queried agent.

It is interesting to note that the defection in providing witness information can be based on malicious incentive, incompetence or even noise. A witness agent might have an incentive to misrepresent its trust view of the trustee, which might result in a positive or a negative effect on a trustee's reputation. The witness agent may choose to overestimate the trust value of a trustee in the case of having a strong cooperative relationship with the trustee, whereas a competitive relationship may lead the rating agent to underestimate the trustee.

4.4.3 CRI/DRI

Agents might cooperate and defect in terms of reporting their news to their neighbors. We define cooperation and defection in a reporting interaction as follows:

Definition: Cooperation in a reporting interaction (CRI) means that the agent will report important results of its interactions to the other party and it will not hide, lie about, or bias them. Similarly, defection in a reporting interaction (DRI) means that the agent will hide, bias or lie about the result of its own interactions with its other neighbors.

4.4.4 CII/DII

An agent with regard to introduction of one agent (a neighbor) to another one (another neighbor) can cooperate or defect. There are four cases regarding this cooperation/defection:

1. An agent introduces two trustworthy agents to each other, which is considered cooperation of the agent with both other agents.

2. An agent introduces one trustworthy agent to one untrustworthy agent, which is considered cooperation of the agent with the untrustworthy agent and defection for the trustworthy one.

3. An agent prevents the introduction of two trustworthy agents to each other, while they are known as trustworthy agents from the perspective of the given agent. This is considered as defection for both of the trustworthy agents.

4. An agent introduces two untrustworthy agents to each other, which is considered defection of the agent with both other agents.

Definition: Cooperation in introducing agents to each other (CII) means that the cooperative agent will introduce trustworthy agents to each other. In contrast, defection in introducing agents to each other (DII) means that the agent will not introduce trustworthy agents to each other or introduce an untrustworthy agent to the trustworthy one.

CII/DII can be perceived indirectly and directly. For indirect perception, when agent k is introduced to agent i by agent j, agent i based on the CDI/DDI of agent k can understand that this introduction was cooperation or defection. In other words, if the introduced agent k cooperates with agent i in the context of direct interactions, those cooperations also take into account for agent j's introduction interaction. Likewise, if agent k defects, this defection also will count for the agent j's introduction interaction. In this light, someone who introduces two agents to each other is responsible for the behavior of them, and will be punished or rewarded for this introduction.

4.5 Protocols

We here explain protocols of the ERT model which agents use for their different interactions and for connecting with each other. Corresponding to four types of interactions explained in Section 4.2, there are four protocols: Direct Interaction Protocol, Reporting Interaction Protocol, Witness Interaction Protocol, and Introduction Interaction Protocol. All protocols use messages and each message is defined by the tuple of <*Name, Content, SenderID, DestinationID, TargetID*>, where *Name* shows the type (name) of the message. The value of the *Content* variable differs in each message type (i.e., it is type-dependent). $SenderID$ and $DestinationID$ represent the sender's id (name) of the message and the destination's id (name) of message

respectively. *TargetID*, which is used by some message types (not all), provides meta-data information.

4.5.1 Direct Interaction Protocol

To model playing the prisoner's dilemma game for a direct interaction, each agent sends a Direct Interaction Message (DIM) with the value of either cooperation or defection to each of its neighbors. As the neighbors will do the same, the agent will receive direct interaction messages from them as well. We denote a Direct Interaction Message as *<DIM,CDI/DDI, SenderID, DestinationID, nil>*, where nil means that no information is provided.

4.5.2 Witness Interaction Protocol

To simulate the witness interaction in our model, the agent looking for witness information about a target agent will send an *investigation* message, denoted by *<Inv, nil, SenderID, DestinationID, TargetID>*, to one or all of its neighbors. $TargetID$ includes the ID of the target agent and $SenderID$ is the id of the asker agent. The receiver agent will send its opinion in response to the investigation message using an *opinion* message denoted by *<Op, rating, SenderID, DestinationID, TargetID>*. The witness agent who sends out the *opinion* message will send a Witness Interaction Message (WIM) to the asker agent after T_w cycles of simulation. The witness interaction message denoted by *<WIM, CWI/DWI, SenderID, DestinationID, nil>* indicates that the previous *opinion* message was cooperation or defection. The WI message will be sent out after T_w cycles to simulate this fact that it takes some time to understand whether the witness agent was cooperative or non-cooperative in the provided witness information. The intention behind the WI message is to simulate the perception of whether the corresponding opinion message was cooperation/defection, and it should not be mistaken as an agent's confession about its cooperation or defection.

In order to clarify the above explanation, consider the example illustrated in Figure 4.2, where *investigation* and *opinion* messages are depicted by boxes with the labels of I and O respectively. Agent X intends to know about the reputation of the target

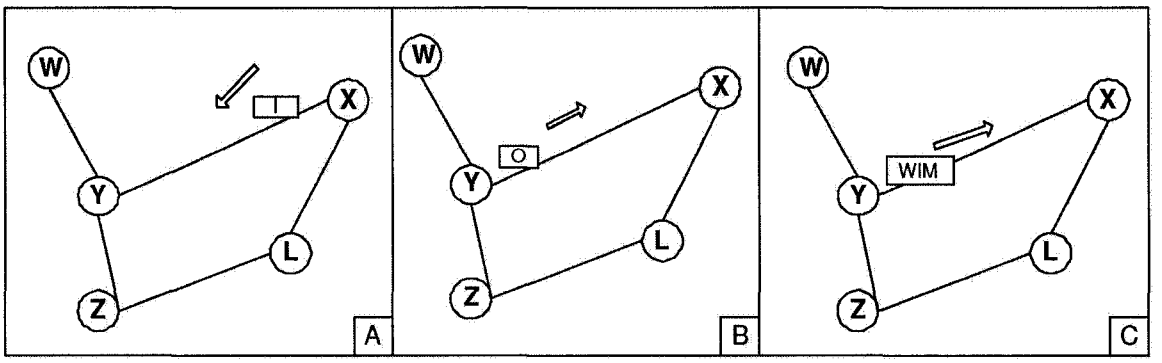


Figure 4.2: A Scenario for Demonstration of Witness Interaction Protocol

agent Z while never having interacted with it. Agent X thus sends an *investigation* message to agent Y asking about agent Z as shown in Figure 4.2A. Upon receiving the *investigation* message, agent Y sends out its rating of agent Z to agent X as illustrated in Figure 4.2B. After T_w cycles of simulation, agent Y will send *WIM* to agent X as shown in Figure 4.2C.

It is worth mentioning that the perception of whether provided witness information is a cooperation or defection is not a hard problem. This problem is solved by other researchers and is beyond the scope of this thesis. There are two basic approaches to achieving this perception that are proposed in the literature; these are referred to as endogenous and exogenous methods by Josang et al. [37]. The former tries to detect unreliable witness information (opinions) by using the statistical properties of the reported opinions; for example, [75, 20]. The latter rely on other information such as the reputation of the source or the relationship with the trustee such as used in the work of Yu and Singh (2003) [80].

4.5.3 Reporting Interaction Protocol

In every cycle, or after specific number of cycles, each agent can send out all or a part of the results of its direct interactions with its neighbors in the format of a Report message, denoted by *<RM, Reports, ReporterID, DestinationID, nil>*. A report message includes the results of several direct interactions of the reporter with its neighbors stored in the *Reports* array. The result of each interaction is an element of the *Reports* array and is a tuple $<ID_1, ID_2, A_1, A_2>$, where A_1 and A_2 are the

actions of the agent ID_1 and ID_2 in the reported interaction respectively. Note that the value of A_1 and A_2 are either cooperation or defection (more precisely, either CDI or DDI).

According to the simulation of the perception of cooperation/defection in a reporting interaction, we first concentrate on how an agent can understand whether one neighbor is cooperative in reporting interactions or not. There are several scenarios, shown in Figure 4.3, which are:

- As shown in Figure 4.3A, agent X is receiving the reports of interactions from Y. To understand whether Y is cooperating or not, agent X needs to hear the same report from another trustworthy source. Suppose W also reports its interactions to X and it is known as a cooperative peer in terms of its reporting interactions. Since Y and W are interacting with one another, so their reports on interactions with each other should be compatible. If incompatible, Y is not cooperative in reporting interactions given that W is already known as a trustworthy agent in terms of reporting interactions.

- As shown in Figure 4.3B, agent X is receiving reports of interactions of Y with L and Z from Y. To understand whether Y cooperates in reporting, agent X needs to hear the same report from another trustworthy source. But X cannot hear about these interactions from other parties, so it will consider that Y is cooperating in reporting unless proven otherwise. Agent X might connect to L, and Z in future and hear about their previous interactions with Y. Then X will determine whether Y was cooperative previously or not.

To model this perception simply, the reporter agent (agent Y in the above example) will send a Reporting Interaction Message (RIM) to the listener agent (agent X in the above example) indicating whether it was cooperative in reporting the interactions or not. A RIM is denoted by *<RIM, CRI/DRI, ReporterID, DestinationID, nil>*. The RIM will be sent out T_r cycles after the corresponding report was sent out. This simulates the fact that it takes time to understand whether a reporter was cooperative or non-cooperative in the provided reports. The intention behind the RIM is to simulate the perception of whether the corresponding report message

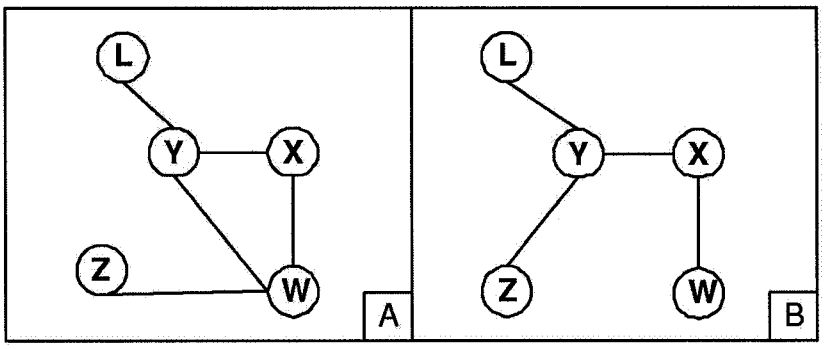


Figure 4.3: Scenarios for Reporting Interactions

was cooperation/defection, and it should not be interpreted as an agent's statement about its cooperation or defection.

4.5.4 Introduction Interaction Protocol

As explained in Section 4.2.3, an introduction interaction can be either request-driven or asynchronous. For request-driven interactions, the agent which is looking for a recommendation sends a *AskingForRecommendation* Message (AFRM), denoted by *<AFRM, nil, SenderID, DestinationID, nil>*, to a neighbor. The neighbor has two response choices upon receiving this message:

- Recommending an agent by sending a *Recommendation* Message (REM) denoted by *<REM, nil, RecommenderID, DestinationID,TargetID>*, where *TargetID* maintains the ID of the introduced (recommended) agent.
- Not recommending any agents to the requester.

In the case of introducing an agent, the perception whether this introduction was cooperation or defection is indirect which relies on the cooperation and defection of the introduced agent with the requester (recall section 4.2.3). When there is no recommendation, the perception is direct which we simulate by sending an Introduction Interaction Message (IIM) to the requester. IIM is denoted by *<IIM,CII/DII, IntroducerID, DestinationID, nil>* and can have the value of cooperation or defection (CII/DII). If the recommender agent has no unknown-to-the-requester trustworthy neighbors then it will send IIM with the value of cooperation (CII) to the requester.

But if the recommender agent was reluctant to introduce a trustworthy agent to the requester, it will send an introduction interaction message with the value of defection (DII) to the requester. It is important to note that introduction interaction messages simulate the perception of cooperation/defection while in a live system the cooperation/defection of an introduction interaction would be something that the agent itself would determine based on the received information.

All the above explanations are valid for asynchronous introduction but the difference is that there is no AskingForRecommendation message in an asynchronous introduction. The recommender agent will send the Recommendation message to a neighbor at any time based on its introduction policy. Depending on the situation, the introduction interaction message will be forwarded to the neighbors.

4.6 Connection and Disconnection

Agents are interacting solely with their neighbors. The agent's neighborhood is dynamic and subject to change over the course of the simulation. An agent can disconnect from a neighbor and exclude it from the neighborhood set or it might make a new connection to other agents and include them in the neighborhood set. In Section 5.8, we will explain that these decisions are all made by the connection policy of agents.

Two agents can become connected to each other if and only if both agents agree to this relationship. Usually, one agent requests a connection to another agent by sending a Connection Request Message (CRM) denoted by *<CRM, nil, SenderID, DestinationID, nil>* and the other one processes the request and based on its connection policy and perception variables decides whether to accept or reject this connection request. In the case of the acceptance of a connection request, the agent will send a Connection Acknowledge Message (CAM) denoted by *<CAM, nil, SenderID, DestinationID, nil>* to the requester. Agents require the unique ID of the agent with which they intend to connect in order to make a request for connection. These IDs might be acquired either by referring to the registry list(see Section 4.7) or by introduction of another agents. Identity is unique and reported reliably by all agents. Authentication is out of scope of this thesis as indicated in Chapter 1.

For disconnection, the decision of one agent is enough. One agent can disconnect

from a neighbor, and then the neighbor will be notified about this disconnection by a Disconnection Acknowledgment Message (DCAM). DCAM is denoted by <*DCAM, nil, SenderID, DestinationID, nil*>, where senderID represents the agent which has decided to disconnect from DestinationID agent. An agent decides when to disconnect from a neighbor based upon its policies. For example, one agent can disconnect from a neighbor if the neighbor is known as untrustworthy, thus resulting in punishment of the untrustworthy agent by not interacting with it.

4.7 Registry List

There are situations in which agents have a tendency to connect to unknown and unvisited agents in order to interact with them. Two scenarios are modeled in this thesis. When an agent is isolated because of either the consequence of its previous interactions or its recent entrance to the system, it needs to make a connection request to some existing unknown agents. The IDs of those agents are necessary to make this connection request. In the ERT model, we introduce a component called the *registry list* in which the IDs of all existing agents are registered.

This registry list plays a role similar to a white page service in a distributed system. One of the roles of the registry list would be to authenticate agents when they register the system; however authentication was not considered in the research reported in this thesis. Those agents who are in need of a connection can acquire an agent ID by referring to this registry list. It is worth mentioning that knowing the ID of an agent cannot guarantee the successful connection to the given agent (recall Section 4.6). Moreover, IDs appear in random order and the IDs are shuffled by each access to the registry list. This shuffling prevents the attack in which malicious agents try to register themselves at the top of the list to attract more isolated agents to themselves and consequently to have more opportunities for fraudulent interactions.

4.8 Agent Type and Initialization

The ERT model provides the facility to define and to specify heterogeneous agents in terms of their perceptions and behaviors. Each agent type is defined as a tuple <*id,*

Po_{id}, PV_{id}> where id is a unique identifier for that specific type of agent and Po_{id} is a set of policies for different interaction types and connection/disconnection policies. These policies make decisions based on the set of perception variables PV_{id} consisting of trust and reputation variables (see Chapter 5).

ERT offers a facility to define different types of agents varying in their Po and PV sets. After definition and specification of Po and PV for each agent type, agents in the simulation environment will be initialized using one of these types. We define the $PT = \{p_1, p_2, ..., p_n\}$ vector which consists of the percentage of each type of agent in the simulation environment. The p_i in PT vector is the percentage of agents which will be initialized by Po_i and PV_i of agent type i. Note that, $\sum_{i=1}^{n} p_i = 1$. In other words, each agent is initialized by the Po_i and PV_i with the probability of p_i.

4.9 Newcomers

Newcomers play crucial roles and have their own concerns in open distributed systems given that distributed entities might enter into the system at any time. To model this characteristic, some agents can uniformly be inserted as isolated agents (nodes) in the environment over the course of the simulation. Suppose that the simulation period is 300 cycles and 50 agents are intended to be inserted over the simulation period. In this case, every 6 cycles (time step) one agent, an isolated node, will be inserted into the system.

A newcomer adopts its type based on the same p_i probability explained in Section 4.8. The newcomer is not able to interact until it gets connected to at least one agent. To make a connection with an existing agent, the newcomer should acquire an agent ID and make a connection request. This ID acquisition can be accomplished by accessing the registry list as explained in Section 4.7.

4.10 Metrics

ERT provides a collection of tools and metrics to researchers in order to experimentally analyze the agent types on both microscopic and macroscopic levels.

On the macro level, the structure of agent society will be depicted in the form

of undirected graph over the course of the simulation. This visualization assists a researcher study how society structure will be changed over interactions.

On the micro level, we were interested in examining the internal properties of each agent type, such as utility of agents and the number of unsuccessful connections made by agents which is an indicator of the encounter risk of agents.

ERT offers the following metrics for micro level analysis:

$\overline{U_{AT}(i)}$, the average of utilities for agents with the type of AT at time step i, is calculated by:

$$\overline{U_{AT}(i)} = \frac{\sum_{a \in AT} U_{Avg}(a, i)}{N_{AT}} \tag{4.1}$$

where $U_{Avg}(a, i)$ is the average of utility of agent a over its interactions at time step i and N_{AT} is the total number of agents of society whose type is AT. The utility of each interaction is calculated based on the following payoff matrix (well-known payoff matrix of the Iterated Prisoner's Dilemma [5]):

P_1/P_2	**Cooperate**	**Defect**
Cooperate	**3,3**	**0,5**
Defect	**5,0**	**1,1**

Table 4.1: Payoff Matrix of Iterated Prisoner's Dilemma

According to Table 4.1, if agent P_1 defects and agent P_2 cooperates, agent P_1 gets the Temptation to Defect payoff of 5 points while agent P_2 receives the Suckers payoff of 0. If both cooperate each gets the Reward for Mutual Cooperation payoff of 3 points, while if both defect each gets the Punishment for Mutual Defection payoff of 1 point.

$\overline{D_{AT}(i)}$, the average of dropped connections for agents with the type of AT at time step i, is calculated by:

$$\overline{D_{AT}(i)} = \frac{\sum_{a \in AT} D_{total}(a, i)}{N_{AT}} \tag{4.2}$$

where $D_{total}(a, i)$ is the total number of connections broken for agent a from the start time to time step i and N_{AT} is the total number of agents of society whose type is AT.

4.11 Summary

In this chapter, we proposed our testbed which provides researchers with facilities to model a general experiment environment. Individuals in open distributed systems can be modeled as agents in the environment model of ERT. The proposed testbed provides the desirable features and properties for an effective trust and reputation environment which are presented by Fullam et al. [26]:

- **Modularity:** The proposed testbed provides a wide range of capabilities through adjustable environment and agent parameters. Parameterization allows the researcher flexibility while conducting a wide variety of experimental scenarios.
- **Multipurpose Design:** Our testbed can be used in different modes such as experiments and competitions. In competition mode, the average utility metric presented in Section 4.10 can be used to rank the agent types.
- **Accessibility:** Various types of trust and reputation can be tested in the ERT. The proposed environment model is completely independent of an agent's architecture and model. This model provides the opportunity to define different agents types as explained in Section 4.8.
- **Objective Metrics:** The metrics in ERT, as explained in Section 4.10, captures single-agent (microscopic) and system-wide (macroscopic) perspectives.
- **Problem Focus:** By employing game theoretical concepts and notions (recall Section 4.3), our testbed is not restricted to domains such as e-commerce. It is an abstract model.

The environment model of ERT does not suffer any of the shortcomings of the iterated prisoner's dilemma (mentioned in Section 3.6) in spite of maintaining the simplicity of game theoretic models. First, agents can evaluate different aspects of opponents behavior and consequently multi-dimensional trust is encouraged. Second, agents can separate untrustworthy agents because they do not have to interact with all other agents and only have to interact with their neighbors. Third, it is equipped with system-level (macroscopic) metrics.

While existing testbeds such as the Iterated Prisoner's Dilemma and ART (recall subsection 3.6) have focused on either or both of direct interactions and witness interactions, agents can have four types of interactions in the proposed model: Direct Interaction, Witness Interaction, Reporting Interaction, and Introduction Interaction. Reporting interactions are the localized decentralized reporting mechanism which let an agent inform its neighbors regarding the result of its current interactions. The introduction interaction, which can be request-driven or asynchronous, provides an incentive for agents to be trustworthy in order to extend their trustworthy neighborhood.

We modeled the perception of cooperation and defection (recall Section 4.5) and explained the interpretations of them (recall Section 4.4) for each interaction type. The openness of distributed systems is modeled and explained in Section 4.9. We present the registry list which plays the role of white page service while the isolated agents and newcomers can look up the IDs of other agents using it.

In the next chapter, we will explain the agent model of ERT, which assists agents in deciding with whom, when and how to interact in the environment model proposed in this chapter.

Chapter 5

The Agent Model of ERT

5.1 Introduction

As noted previously, the majority of open distributed computer systems can be modeled as multi-agent systems (MAS) in which each component acts autonomously to achieve its objectives [34]. This chapter explains our proposed agent model (the agent model of ERT) which provides mechanisms for deciding with whom, when and how an agent will interact. This agent model is designed to perceive the behavior of other agents and consequently predict the trustworthiness of them in order to help an agent in making low-risk decisions.

The proposed agent model consists of a set of perception variables and a set of policies. The perception variables, including trust variables and reputation variables, help agents in modeling the trustworthiness and reliability of other agents. On the other hand, the policies assist them in how they should behave with others considering the perceived trustworthiness of the other (possibly adversarial) agents. The perception variables are designed so as to be exploitation resistant while the policies are intended to weed out untrustworthy agents within the agent society.

This chapter is organized as follows. The goal on which our agent model is based is explained in Section 5.2. The design challenges and requirements are detailed in Section 5.3 and Section 5.4 respectively. The trust variables, the con-resistance component and reputation variables of ERT are described in Section 5.5, Section 5.6, and Section 5.7 respectively. Agent policies are described in Section 5.8. Finally, we summarize the chapter in Section 5.9.

5.2 Goal

The main utility of trust and reputation models is minimizing the the risk of interacting with others. To reach this goal, an agent must be able to model the trustworthiness of potential interaction partners and make decisions based on those models. In other words, to reach its goals, an agent usually requires resources that only other agents can provide. The agent benefits from choosing the agents with which it interacts and which are most likely to provide those resources. In this light, the agent can minimize the risk of unsuccessful interactions and failure by predicting the outcome of interactions, and avoiding risky (unreliable) agents. Modeling the trustworthiness of potential interaction partners enables the agent to make these predictions. Broadly speaking, the aim of each trust and reputation model is to guide an agent's decision making in deciding how, when and with whom to interact in an uncertain environment.

Although trust and reputation models have a strong foundation with the assumption that agents may attempt to exploit each other, there is little consideration of the possibility that agents may attempt to exploit the trust and reputation model itself. This lack of consideration leads models to be vulnerable. This vulnerability in trust and reputation models may allow an agent (or group of agents) to cheat other agents without the model recognizing the cheaters (malicious agents).

Therefore, research objectives and goals firstly include building exploitation-resistant trust models with desirable characteristics as explained in Section 3.3.4 and secondly include an agent's ability to make decisions and take actions based on its trust models. Moreover, to empirically analyze the exploitation-resistance feature of each trust model, we need to accommodate various types of agent behavior. The agent model presented in this chapter enables researchers to achieve this task and model heterogeneous agents such as malicious, naive (see Section 5.3.2), learning agents, etc.

5.3 Challenges

There are several challenges in designing exploitation-resistance trust and reputation models. We present these challenges through several attacks which are not referred to

as conventional attacks on the system implementation itself. In contrast, these attacks are referred only to attacks composed of legitimate actions within the system itself. We put these attacks into two categories: individual attacks and collusion attacks. From the former category, we formally model the Con-man Attack as explained in Section 5.3.1. From the latter category, we model the Witness-based Collusion Attack and Report-based Collusion Attack as described in Section 5.3.2.

5.3.1 The Con-man Attack

A con-man, also known as a "confidence man", is someone who takes advantage of someone else – usually for financial gain – using what is known as a confidence trick, where a confidence trick or confidence game is an attempt to defraud a person or group by gaining their confidence.

To model the con-man attack, we use the terms *cooperation* and *defection* from the language of game theory. The level of trust of an agent towards another agent can be changed based on the evaluation of an interaction. If an agent perceives the other agent was cooperative during the specific interaction, its trust in the other agent will be increased. In contrast, if the agent perceives that the other agent has defected for a specific interaction, it will decrease its trust in that agent.

What the con-man does is to build up trust from the victim's view point by cooperating with him/her several times. Then, when it comes to a high risk interaction, the con-man will defect. After the con-man has defrauded the victim, he/she has two choices: never interact again with the victim or regain the lost trust with some subsequent cooperative behavior. The con-man, by regaining the victim's trust, can again con (defect) the victim.

In our view, it is hard to understand the intention of a cooperative person and to make sure he/she will continue cooperating forever and will never be tempted to con. Therefore, this thesis does not plan to identify the con-man before the con happens. Our work is aimed at identifying the repetition of the confidence trick and not let the con-man regain a high trust value easily.

We model the repetition of a confidence trick by introducing the parameter θ. The con-man will defect after θ times of cooperation. After each defection, the con-man

will again cooperate θ times. The con-man will repeat this interaction pattern several times (maybe, forever). The formal language (natural language) L over the alphabet $\Sigma = \{C, D\}$ demonstrates the interaction pattern of the con-man:

$$L = \{(C^{\theta}D)^{+} | \theta \geq 1\} \tag{5.1}$$

where C and D stand for cooperation and defection respectively.

In Section 6.3.1, we will demonstrate how three well-known trust models fail to identify the repetition of a confidence trick and the con-man still will have/can gain a high trust value. We have observed this type of attack in reputation management systems used by eBay, for example. In eBay, sellers with good reputations can take advantage of their good reputations to sell a few faulty and low-quality products among plenty of high-quality products that they are selling. For instance, a microphone seller with a good reputation might sell 980 high-quality microphones and 20 faulty microphones every month. Despite his/her defection for selling 20 damaged microphones, he/she can still have a high reputation value (above 90%) since the reputation value is calculated as the sum of all ratings over the last six months.

It should be observed that agents with time-varying behavior have been previously studied in other works to test the adaptability of trust models. For instance, Hang et al. [29] introduced damping and capricious agents to analyze the adaptability of its trust scheme. Capricious agents change their behavior between cooperation and defection every two cycles and damping agents have cooperative behavior for several cycles before defecting for the remainder. In this thesis, the detection of the con-man by the trust model is of interest instead of analyzing the adaptability of the trust model.

5.3.2 Witness-based and Report-based Collusion Attacks

In a witness-based collusion attack, an unreliable witness provider – in spite of being cooperative in its direct interactions – is unreliable in witness interactions by providing high ratings for other malicious agents (other members of the colluding group). These high ratings, and the fact that the given witness provider is cooperative in direct interaction, encourage the asker agent to interact with other members of the colluding

group. Consequently, the asker agent will be exploited by them.

An unreliable witness provider can have a malicious or non-malicious intent. When it cooperates with malicious agents by providing high ratings in the favor of them, the unreliable witness provider has malicious intent. The unreliable witness provider which is naive in terms of the assessment of other agents and employed by the other malicious agents has non-malicious intent. As intention and belief modeling of agents are not within the scope of this thesis, we introduce the concept of a naive agent which is consistent with both mentioned models of intention. Our proposed naive agent can be considered either as a part of colluding group or as a naive agent deployed by malicious agents.

We define a naive agent as follows: a naive agent is incapable of properly deciding how, when and with whom to interact. In this sense, it fails to detect and stop interacting with untrustworthy agents due to the lack of proper assessment of other agents. They are optimistic such that they consider all other agents completely trustworthy and always cooperate with every member of the society. Naive agents provide high ratings for every member of the agent society including malicious agents.

Examples of naive agents can be seen in many places. On eBay, sellers receive feedback (+1, 0, -1) in each auction and their reputation is calculated as the sum of those ratings over the last six months. It can be observed that there are many users (buyers) who do not receive satisfactory goods or services but they rate the sellers highly and even continue interacting with them. We see these users as naive users. In peer-to-peer file sharing systems free riding is a well-documented problem (e.g., BitTorrent). Free-riders do not share enough or appropriate files while benefiting from the society by downloading files from peers. It can be observed that there are some users in these systems who are incapable of detecting free-riders and share all of their files to everyone in the society. These peers follow our definition of naive agents.

In a Report-based collusion attack, a naive reporter – in spite of being cooperative in its direct interactions – is unreliable in reporting interactions by reporting positively regarding the interactions of malicious agents (other members of the colluding group). Similarly to the Witness-based Collusion Attack, we have modeled Report-based Collusion by using the concept of a naive agent introduced above.

5.4 Requirements

We herein explore the necessary requirements for preventing the attack scenarios discussed in Section 5.3. We have declared the desirable exploitation-resistance characteristics for each of the con-man attack and witness-based collusion attacks in the two subsections of 5.4.1 and 5.4.2 respectively. These characteristics are proposed based on the attacks reported in Section 5.3. By modeling more complicated attacks, other requirements might be added.

5.4.1 Characteristics of Con-resistant Models

Formally, a trust and reputation model is con-resistant if it capable of detecting a con-man and labeling him as untrustworthy. A con-man is an agent employing the con-man strategy defined in equation 5.1. To explore the features of con-resistant trust models, we provide a hypothetical example. Alice and Carol are the owners of two separate bakeries. Alice can identify con-men but Carol can not. Bob is the manager of a flour mill. Bob offers to provide high quality flour to each bakery; both accept.

Carol initially accepts daily shipments of 50kg. After 10 satisfactory shipments (cooperations), Carol increases her trust in Bob by doubling her daily order to 100kg. The next day Bob sends low-quality flour at an unchanged price to Carol (a defection). Carol understands the defection and reduces her order to its initial size (50kg). Bob realizes that Carol detected the defection and so cooperates by providing high-quality flour. Bob and Carol continue this cyclical interaction pattern (10 days cooperation then one day defection) for a long time; Carol never understands that Bob is playing a confidence trick on her.

Alice also accepts 50kg daily from Bob who attempts the same confidence trick. Alice doubles her order after 10 satisfactory shipments. However, when Bob defects, Alice realizing the defection reduces her order to 40kg (less than its initial size) and doubles the number of shipments required before increasing her order to 20 shipments. When Bob repeats this cycle, Alice remembers the previous defections and reduces her order by 10kg when compared to the starting cycle order. After 5 cycles, Alice cancels her contract with Bob.

Alice detects the confidence trick by doing two things: reducing trust more severely than the previous reduction and decreasing the rate at which trust accumulates with each cooperation. She does this by remembering defections, which Carol does not.

We propose the following heuristics for con-resistant trust models:

- **Cautiously increment trust after defection:** The more the agent perceives defection, the corresponding trust value should be increased more slowly by perceiving the consecutive cooperations.

- **Larger punishment after each defection:** The more the agent perceives defection, the corresponding trust value should be decreased more sharply by perceiving each defection.

The above heuristics will not remove forgiveness from trust models, which is a frequently noted aspect of trust and reputation theory [62, 5]. The above characteristics are mainly motivated by the facts that forgiveness is slower when several defections have happened, and punishments are bigger for those who defect more.

In general, to prevent individual attacks, we encourage researchers to assume changing agent behavior in their designs and to design adaptive trust update mechanisms which adapts their parameters based on the perceived behavioral changes.

5.4.2 Characteristics of Collusion-resistant Models

As explained in Section 5.3.2, in a witness-based collusion attack, an unreliable witness provider in spite of being cooperative in its direct interactions is unreliable in witness interactions. This unreliability is because high ratings are provided for other malicious agents (other members of the colluding group). An asker agent might consider these unreliable witness informations (rating) as reliable information because the witness provider is cooperative and trustworthy in direct interactions. As a consequence, the witness-based collusion attack will take place.

It can be observed that when the asker agent bases its assessment of witness information on the cooperations (trustworthiness) in direct interactions, this attack will be successful. In particular, the success of this attack is the result of the inappropriate

assumption that whoever is cooperative (trustworthy) in direct interactions will be cooperative (trustworthy) in providing witness information regarding other agents.

The witness-based collusion attack is prevented if the asker agent utilized an independent multi-dimensional trust model. In this sense, the asker agent will assess the witness providers based on their cooperations in witness interactions. Independent multi-dimensional trust has a strong foundation given the simple fact that if a person is trustworthy regarding service provision (direct interaction), he may not be trustworthy in assessing (rating) the trustworthiness of others (witness interaction). This hypothesis simply prevents the witness-based collusion attack, where one untrustworthy agent masquerades itself as a trustworthy agent in order to rate falsely untrustworthy agents. Similar observations can be made regarding report-based collusion attacks.

As stated in Section 3.5.3, collusion attacks usually work based on the basic idea that one or more agents show themselves as trustworthy agents in one interaction type (usually direct interaction). Afterward, they will be untrustworthy in another type of interaction (e.g., witness interaction) by providing false information in the favor of other members of colluding groups. It is the hypothesis of this thesis that independent multi-dimensional trust can provide a solution for many collusion attacks. The key point is to have an independent dimension of trust per type of interaction taking place in the environment.

5.5 Trust Variables

To have multi-dimensional trust models, ERT equips an agent (the truster) with four independent dimensions of trust (trust variables) while each trust variable corresponds to an interaction type. The motivation for having four trust variables is that trust in information received should be independently assessed. Furthermore, it is the position of this thesis that such independent trust assessment is crucial for collusion resistant trust models. As stated above, an agent which is trustworthy in direct interactions is not necessarily trustworthy in reporting interactions or witness interactions. This section provides descriptions for these trust variables.

Each trust variable is defined by $T_{i,j}(t)$ indicating the trust rating assigned by

agent i to agent j after t interactions between agent i and agent j, while $T_{i,j}(t) \in [-1, +1]$ and $T_{i,j}(0) = 0$. In ERT, one agent in the view of the other agent can have one of the following levels of trustworthiness:

- *Trustworthy*
- *Not Yet Known*
- *Untrustworthy*

Following Marsh [47], we define for each agent an upper and a lower threshold to model different levels of trustworthiness. The agent i has its own upper threshold $-1 \leq \omega_i \leq 1$ and lower threshold $-1 \leq \Omega_i \leq \omega_i$. Agent j is trustworthy from the viewpoint of agent i after t times of interactions if and only if $T_{i,j}(t) \geq \omega_i$. Agent i sees agent j as a untrustworthy agent if $T_{i,j}(t) \leq \Omega_i$ and if $\Omega_i < T_{i,j}(t) < \omega_i$ then the agent j is in the state *Not Yet Known* in agent i 's view.

5.5.1 Direct Interaction Trust (DIT)

Direct Interaction Trust (DIT) is the result of the cooperation/defection that agents have in their direct interactions (CDI/DDI). Each agent maintains $DIT_{i,j}(t)$ variables for the agents having had direct interactions with them (its neighbors or ex-neighbors). The agent will update this variable based on the perception of cooperation/defection in direct interaction (CDI/DDI). We used the following trust updating scheme motivated by that proposed in [78]:

$$DIT_{i,j}(t+1) = \begin{cases} DIT_{i,j}(t) + \alpha_D(i)(1 - DIT_{i,j}(t)) & DIT_{i,j}(t) > 0\ ,\ CDI \\ (DIT_{i,j}(t) + \alpha_D(i))/(1 - min(|DIT_{i,j}(t))|\ , |\alpha_D(i)|) & DIT_{i,j}(t) < 0\ ,\ CDI \\ (DIT_{i,j}(t) + \beta_D(i))/(1 - min(|DIT_{i,j}(t))|\ , |\beta_D(i)|) & DIT_{i,j}(t) > 0\ ,\ DDI \\ DIT_{i,j}(t) + \beta_D(i)(1 + DIT_{i,j}(t)) & DIT_{i,j}(t) < 0\ ,\ DDI \end{cases} \tag{5.2}$$

Where $1 > \alpha_D(i) > 0$ and $-1 < \beta_D(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively for updating of the direct interaction trust variable of agent i. The value of $DIT_{i,j}(t)$, ω_i^{DIT} and Ω_i^{DIT} determine that the agent j is either *Trustworthy*, *Not Yet Known* or *Untrustworthy* in terms of direct interaction from the perspective of agent i.

5.5.2 Witness Interaction Trust (WIT)

Witness Interaction Trust (WIT) is the result of the cooperation/defection that agents have regarding their witness interactions (CWI/DWI). Agent i maintains a $WIT_{i,j}(t)$ variable for the agent j from whom it has received witness information. Agent i will update this variable based on the perception of CWI/DWI from agent j.

$WIT_{i,j}(t+1) =$

$$\begin{cases} WIT_{i,j}(t) + \alpha_W(i)(1 - WIT_{i,j}(t)) & WIT_{i,j}(t) > 0\ ,\ CWI \\ (WIT_{i,j}(t) + \alpha_W(i))/(1 - min(|WIT_{i,j}(t))|\ , |\alpha_W(i)|) & WIT_{i,j}(t) < 0\ ,\ CWI \\ (WIT_{i,j}(t) + \beta_W(i))/(1 - min(|WIT_{i,j}(t))|\ , |\beta_W(i)|) & WIT_{i,j}(t) > 0\ ,\ DWI \\ WIT_{i,j}(t) + \beta_W(i)(1 + WIT_{i,j}(t)) & WIT_{i,j}(t) < 0\ ,\ DWI \end{cases} \tag{5.3}$$

Where $1 > \alpha_W(i) > 0$ and $-1 < \beta_W(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively for updating of the WIT variable of agent i. The value of $WIT_{i,j}(t)$, ω_i^{WIT} and Ω_i^{WIT} determine that the agent j is either *Trustworthy*, *Not Yet Known* or *Untrustworthy* in terms of witness interactions from the perspective of agent i.

5.5.3 Reporting Interaction Trust (RIT)

Reporting Interaction Trust (RIT) is the result of the cooperation/defection that agents have in reporting their interactions to their neighbors (CRI/DRI). Agent i maintains a $RIT_{i,j}(t)$ variable for the agent j from whom it has received report messages. Agent i will update this variable based on the perception of CRI/DRI from agent j.

$RIT_{i,j}(t+1) =$

$$\begin{cases} RIT_{i,j}(t) + \alpha_R(i)(1 - RIT_{i,j}(t)) & RIT_{i,j}(t) > 0\ ,\ CRI \\ (RIT_{i,j}(t) + \alpha_R(i))/(1 - min(|RIT_{i,j}(t))|\ , |\alpha_R(i)|) & RIT_{i,j}(t) < 0\ ,\ CRI \\ (RIT_{i,j}(t) + \beta_R(i))/(1 - min(|RIT_{i,j}(t))|\ , |\beta_R(i)|) & RIT_{i,j}(t) > 0\ ,\ DRI \\ RIT_{i,j}(t) + \beta_R(i)(1 + RIT_{i,j}(t)) & RIT_{i,j}(t) < 0\ ,\ DRI \end{cases} \tag{5.4}$$

Where $1 > \alpha_R(i) > 0$ and $-1 < \beta_R(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively for updating of the RIT Trust variable.

The value of $RIT_{i,j}(t)$, ω_i^{RIT} and Ω_i^{RIT} demonstrate that the agent j is either *Trustworthy, Not yet Known* or *Untrustworthy* in terms of reporting interactions from the perspective of agent i.

5.5.4 Introduction Interaction Trust (IIT)

Introduction Interaction Trust (IIT) is the result of the cooperation/defection of the agents in introduction interactions (CII/DII). Agent i maintains an $IIT_{i,j}(t)$ variable for the agent j which has introduced agent k. Agent i will update this variable based on the perception of CII/DII of introducer agent j and CDI/DDI of the introduced agent k.

$IIT_{i,j}(t+1) =$

$$\begin{cases} IIT_{i,j}(t) + \alpha_I(i)(1 - IIT_{i,j}(t)) & IIT_{i,j}(t) > 0\ ,\ CII(j) \\ (IIT_{i,j}(t) + \alpha_I(i))/(1 - min(|IIT_{i,j}(t))|\ , |\alpha_I(i)|) & IIT_{i,j}(t) < 0\ ,\ CII(j) \\ (IIT_{i,j}(t) + \beta_I(i))/(1 - min(|IIT_{i,j}(t))|\ , |\beta_I(i)|) & IIT_{i,j}(t) > 0\ ,\ DII(j) \\ IIT_{i,j}(t) + \beta_I(i)(1 + IIT_{i,j}(t)) & IIT_{i,j}(t) < 0\ ,\ DII(j) \\ IIT_{i,j}(t) + \alpha_I(i)(1 - IIT_{i,j}(t)) & IIT_{i,j}(t) > 0\ ,\ CDI(k) \\ (IIT_{i,j}(t) + \alpha_I(i))/(1 - min(|IIT_{i,j}(t))|\ , |\alpha_I(i)|) & IIT_{i,j}(t) < 0\ ,\ CDI(k) \\ (IIT_{i,j}(t) + \beta_I(i))/(1 - min(|IIT_{i,j}(t))|\ , |\beta_I(i)|) & IIT_{i,j}(t) > 0\ ,\ DDI(k) \\ IIT_{i,j}(t) + \beta_I(i)(1 + IIT_{i,j}(t)) & IIT_{i,j}(t) < 0\ ,\ DDI(k) \end{cases} \tag{5.5}$$

where $CDI(k)/DDI(k)$ is the cooperation/defection in direct interaction received by agent i from introduced agent k. Therefore, a cooperation/defection of the introduced agent k takes into account to update introducer agent j's IIT, $IIT_{i,j}$. This mechanism is the indirect perception of DII/CII which is explained in Section 4.4.4.

Moreover, $1 > \alpha_I(i) > 0$ and $-1 > \beta_I(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively for updating of the IIT variable. The value of $IIT_{i,j}(t)$, ω_i^{IIT} and Ω_i^{IIT} demonstrate that the agent j is either *Trustworthy, Not Yet Known* or *Untrustworthy* in terms of introduction interactions from the perspective of agent i.

5.6 The Con-resistance Component (CRC)

The required trust variables in ERT model have now been introduced. Although the trust variables presented in Section 5.5 are multi-dimensional, which is a requirement for collusion-resistance trust models, they still do not satisfy the con-resistance requirements as explained in Section 5.4.1. We herein explain about our proposed con-resistance component which can easily be integrated with each trust variable such as DIT, RIT, WIT, and IIT.

As explained in Section 5.5, all four dimensions of trust utilize a similar update formula in which $1 > \alpha > 0$ is the rate of trust increment and $-1 < \beta < 0$ is the rate of trust decrement. Furthermore, in Section 5.4.1, we introduced two characteristics of con-resistant trust models: first, cautiously increment trust after having seen any defection and second, larger punishments after each defection. Therefore, defection should decrease α but increase the absolute value of β based on the above characteristics.

We introduce the following update schema for a positive evidence weighting coefficient of $1 > \alpha > 0$ and a negative evidence weighting coefficient $-1 < \beta < 0$ when the agent perceives defection:

$$\alpha = \alpha \times (1 - |\beta|) \tag{5.6}$$

$$\beta = \beta - \gamma_d \times (1 + \beta) \tag{5.7}$$

Where γ_d is the discounting factor, and can be calculated based on following formula:

$$\gamma_d = C \times |T_{i,j}| \tag{5.8}$$

Based on the presented formulae[1], α is decreased with the rate of $1 - |\beta|$ which results in a large decrement of α for a high value of $|\beta|$ and a small decrement of α for a low value of $|\beta|$. We have chosen this rate of decrement because in our view after several defections (when $|\beta|$ is high), making up for a defection should be harder and require more cooperation. As presented in the Formula 5.8, the discounting factor γ_d for the β update is proportional to the absolute value of trust of agent i in agent

[1]Technically, α, β and γ_d should have a subscript to represent interaction type and also be a function of agent index but they are omitted in the interest of clarity.

j, $|T_{i,j}|$. We hypothesize that the discounting factor should be high when the target agent is either trustworthy ($T_{i,j}$ is close to 1) or untrustworthy ($T_{i,j}$ is close to -1). This hypothesis is motivated by the well-known fact that "Trust is hard to earn but easy to lose". $0 < C \leq 1$ is a constant in the above formula and is set to $\frac{1}{e}$ in our experiments.

Furthermore, we introduce the following update formula for α when the agents observe cooperation from other agents:

$$\alpha = \alpha + \gamma_c \times (\alpha_0 - \alpha) \tag{5.9}$$

$$\alpha = Min(\alpha_0, \alpha) \tag{5.10}$$

This update results in an increment of α while α will never exceed its initial value, α_0. Therefore, an agent which previously had a decrement in α as a consequence of defection can compensate for it and gradually increase α to the initial value of α_0 by cooperating for some time. γ_c is the learning rate (discounting factor). We believe that if an agent has a high value of β because of its previous defections, its α should be increased more slowly when it is cooperating. Therefore, γ_c should decrease as the magnitude of β increases and we propose the following formula:

$$\gamma_c = 1 - |\beta| \tag{5.11}$$

From now on, we have this convention that whenever any of the trust dimensions (trust variables) is integrated with a con-resistant component, a "-CRC" suffix will be appended to its name. In this sense, if DIT, WIT, RIT, and IIT use a con-resistant component, they are called DIT-CRC, WTI-CRC, RIT-CRC, and IIT-CRC respectively.

5.7 Reputation Variables

The four trust variables explained in section 5.5 are the result of cooperation/defection of the neighbors of the agent in different aspects of direct, witness, reporting and introduction interactions. These variables are used by the agent to model the trustworthiness of their neighbors in order to understand whether the given agent should maintain its connection with them or how much of the information received by the

agent is reliable. On the other hand, agents need to predict the trustworthiness of those agents with whom they have never interacted. Therefore, we use reputation variables for predicting the trustworthiness of these agents. These reputations are calculated based on the information (report or witness information) received from an agent's neighbors and the related trust variable.

A reputation variable is defined by $R_{i,j}(t)$ indicating the trust rating assigned by agent i to agent j after receiving t pieces of information (report or witness information), while $R_{i,j}(t) \in [-1, +1]$ and $R_{i,j}(0) = 0$.

We have defined two kinds of reputation: (1) Report-based Reputation and (2) Witness-based Reputation. The former is calculated based on the reports received by an agent from its neighbors and the latter is computed relying on witness information received by the agent from its neighbors.

5.7.1 Report-based Reputation (RR)

Report-based Reputation (RR) is calculated based on report information. As explained in Section 4.2, a report conveys the result of an interaction of a neighbor with one of its own neighbors. An agent will store the information of a report in $rep_{i,j}(t)$, showing the result of the t^{th} reported direct interaction of agent i with agent j while the result can be cooperation or defection. For each report, two values of $rep_{i,j}(t)$ and $rep_{j,i}(t)$ will be stored by the report listener. Suppose that agent i has received a report message from agent j regarding the interactions of agent j and k, the agent i (recipient of the report) will calculate the Estimated Direct Interaction Trust, $EDIT_{j,k}(t)$, of agent k from the perspective of agent j based on $rep_{k,j}(t)$.
$EDIT_{j,k}(0) = 0$ and
$EDIT_{j,k}(t+1) =$

$$\begin{cases} EDIT_{j,k}(t) + \alpha_E(i)(1 - EDIT_{j,k}(t)) & EDIT_{j,k}(t) > 0\ ,\ rep_{k,j}(t+1) = CDI \\ (EDIT_{j,k}(t) + \alpha_E(i))/(1 - min(|EDIT_{j,k}(t))|, |\alpha_E(i)|) & EDIT_{j,k}(t) < 0\ ,\ rep_{k,j}(t+1) = CDI \\ (EDIT_{j,k}(t) + \beta_E(i))/(1 - min(|EDIT_{j,k}(t))|, |\beta_E(i)|) & EDIT_{j,k}(t) > 0\ ,\ rep_{k,j}(t+1) = DDI \\ EDIT_{j,k}(t) + \beta_E(i)(1 + EDIT_{j,k}(t)) & EDIT_{j,k}(t) < 0\ ,\ rep_{k,j}(t+1) = DDI \end{cases} \tag{5.12}$$

where $1 > \alpha_E(i) > 0$ and $-1 < \beta_E(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively. The Report-based Reputation of agent k

from the perspective of agent i, $RR_{i,k}(t)$, is calculated by:

$$RR_{i,k}(t) = \frac{\sum_{j \in neighbor(i)} (\phi(RIT_{i,j}(t)) \times EDIT_{j,k}(t))}{\sum_{j \in neighbor(i)} (\phi(RIT_{i,j}(t))} \tag{5.13}$$

Where $neighbor(i)$ includes the neighbors of agent i and $RIT_{i,j}(t)$ is the reporting interaction trust of agent j from the perspective of agent i after receiving t reports. $\phi(r)$ is the converter function that maps the values of a trust variable to the weights in the range of $[0, 1]$ with regard to the related ω and Ω. We present two variants of converter functions, *Linear* and *Logarithmic* denoted by $\phi_{Li}(r)$ and $\phi_{Lo}(r)$ respectively. Figure 5.1 shows these two different types of converter function.

$$\phi_{Li}(r) = \begin{cases} 0 & -1 \leq r < \Omega \\ \frac{r-\Omega}{\omega-\Omega} & \Omega \leq r \leq \omega \\ 1 & \omega < r \leq 1 \end{cases} \qquad \phi_{Lo}(r) = \begin{cases} 0 & -1 \leq r < \Omega \\ \frac{\log(r-\Omega+1)}{\log(\omega-\Omega+1)} & \Omega \leq r \leq \omega \\ 1 & \omega < r \leq 1 \end{cases} \tag{5.14}$$

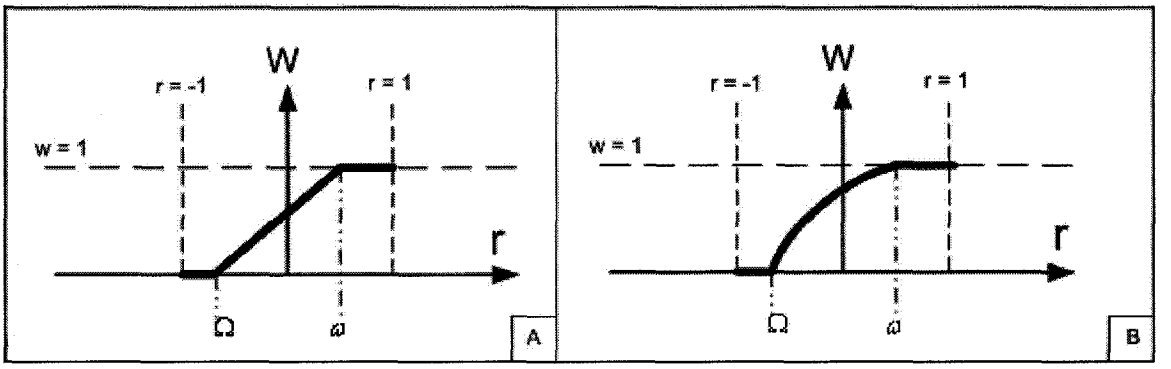


Figure 5.1: Demonstration of A) $\phi_{Li}(t)$ and B) $\phi_{Lo}(t)$ converter functions

5.7.2 Witness-based Reputation (WR)

Witness-based Reputation (WR) for a specific agent is calculated based on the ratings of other agents. As explained in Section 4.2.2, this rating can be based on the observed interactions or direct interactions. The asking agent stores the ratings of other agents in an *Opinion* variable. $Opinion(j, k)$ shows the rating issued by agent j regarding agent k. The value of this variable is in the range of $[-1, 1]$. WR of agent k from the perspective of agent i after reception of t opinions (ratings) is denoted by $WR_{i,k}(t)$ and calculated by:

$$WR_{i,k}(t) = \frac{\sum_{j \in OpinionSenders} (\phi(WIT_{i,j}) \times Opinion(j,k))}{\sum_{j \in OpinionSenders} \phi(WIT_{i,j})} \quad (5.15)$$

Where the *OpinionSenders* variable includes indices of the neighbors of agent i who sent their ratings about agent k and $WIT_{i,j}$ is the current value of WIT variable of agent j from the perspective of agent i. Note that, $\phi(r)$ is a converter function as previously explained in Section 5.7.1.

5.8 Policies and Strategies

The required perception variables (trust and reputation variables) in the ERT model have now been introduced; these variables help agents perceive the cooperation/defection of other agents situated in their environment in four dimensions and consequently to determine the trustworthiness of other agents. Incontrovertibly, this perception provides agents with the foundation for making decisions as with whom they should interact. This perception is absolutely necessary but not sufficient for a trust model since agents need to decide how and when they should interact with other agents. In this sense, each agent requires policies to help them in making decisions for their interactions with the other agents. Different types of policies are introduced and explained in the following subsections.

5.8.1 Direct Interaction Policy (DIP)

This type of policy assists an agent in making decisions regarding its direct interactions while this decision might be made based on the trust perception of the agent. For example, *malicious* agents might have an *unconditional defection* policy which means they defect in interactions with any other agents regardless of its trustworthiness level. They might have a more complicated policy while cooperating with a group of agents and defecting in interactions with the remainder of agent society (i.e., colluding).

5.8.2 Witness Interaction Policy (WIP)

This type of policy exists to aid an agent in making three categories of decisions related to its witness interactions. First, agents should decide how to provide the witness information for another agent on receiving a witness request. Should they manipulate the real information and forward false witness information to the requester (an example of defection) or should they tell the truth? The second decision made by the Witness Interaction Policy is related to when and from whom the agent should ask for witness information. Should the agents ask for the witness information when it has a connection request from an unknown party? Should the agents ask for witness information from a subset or all of its neighbors? The third decision is on how agents should aggregate the received ratings. For example, should the agent calculate the simple average of ratings or a weighted average of ratings?

We defined three sub witness interaction policies: Answering Policy (AP), Querying Policy (QP), and Information-Gathering policy (IGP). Answering Policy intends to cover the the first category of decisions mentioned above while Querying Policy and Information-Gathering policy apply to the second and third categories respectively.

5.8.3 Reporting Interaction Policy (RIP)

This type of policy controls the flow of reports to neighbors. An agent, based on this policy, decides how to select important news and broadcast it to their neighbors. Moreover, the agents decide whether or not to report the interactions honestly or not or whether to hide the news from its neighbors. As with other policies, each agent can cooperate or defect in reporting. We defined two sub reporting interaction policies: Reporting Policy (RP), and Report-Gathering Policy (RGP). Reporting Policy intends to cover how the agent should select the news to report and how the agent should report the past interactions (e.g., being honest or lying). On the other hand, the Report-Gathering Policy focuses on how to aggregate the reports and how to make decisions based on the reports.

5.8.4 Introduction Interaction Policy (IIP)

Agents can have different behaviors for Introduction Interaction; for example, an agent might introduce two trustworthy agents or try to introduce trustworthy agents to untrustworthy ones. Furthermore, agents need to decide if they should connect to the agents recently introduced. These types of decisions are all made by an Introduction Interaction Policy. Note that, for making this type of decision different trust or reputation variables might be taken into account.

Algorithm 1 An Example for Introduction Interaction Policy

```
{Suppose that the agent i is executing this code}
for all j, k ∈ Neighborhood do
  if DIT_{i,j}(t) ≥ ω_i^{DIT} and DIT_{i,k}(t) ≥ ω_i^{DIT} and shouldBeIntroduced(j, k) then
    Introduce k to j
    Introduce j to k
  end if
end for
if i receives introduction of agent k from agent j then
  if IIT_{i,j}(t) ≤ Ω_i^{IIT} then
    Reject the introduction
  else
    ConnectTo(k)
  end if
end if
```

A simple introduction interaction policy is demonstrated in Algorithm 1, for example. This policy attempts to connect trustworthy agents (in direct interactions) together if the primitive $ShouldBeIntroduced(j,k)$ returns a *true* value.

$ShouldBeIntroduced(j,k)$ can be easily implemented relying on whether agent j and agent k have been previously introduced to each other or not; if yes, the primitive returns false otherwise a true value will be returned. The second if statement of this policy deals with the introductions received by the agent. If the introducer agent is *Untrustworthy* in terms of introduction interactions ($IIT_{i,j}(t) \leq \Omega_i^{IIT}$) then the

introduction will be rejected; otherwise the agent will connect to the introduced agent.

5.8.5 Connection Policy (CP)

This type of policy assists an agent in making decisions regarding whether it should make a request for a connection to other agents and whether the agents should accept/reject a request for a connection.

5.8.6 Disconnection Policy (DP)

DP aids an agent in deciding whether or not it should drop a connection to a neighbor. For example, this policy can decide to whether or not to disconnect from a given agent based on the agent's trustworthiness in direct interactions.

5.9 Summary

This chapter has described our proposed agent model (the agent model of ERT) which assists agents in deciding with whom, when and how to interact. This agent model is designed to perceive the behavior of other agents and consequently predict the trustworthiness of them in order to help an agent in making low-risk decisions. After discussing the challenges in the design of our proposed exploitation-resistant trust and reputation model, we described the requirements of exploitation-resistant trust models. In this regard, the trust model should be multi-dimensional and should use an adaptive trust update mechanism. These two characteristics reinforce two of five criteria for evaluation of trust and reputation models as presented by Fullam et al. [26]: **Multi-dimensional**, accurate, quickly converging, **adaptive**, and efficient.

Taking into account all of these criteria, we presented the perception variables of the ERT model which employs four dimensions of trust and two reputation variables to assess the trustworthiness of other agents. Table 5.1 summarizes the features of proposed model, when compared to the existing trust models (see Section 3.10 for comparison and abbreviation definitions).

Name	Information Sources	Interactions	VUL	SO-NET	MTM
ERT	DI+OI+WINFO	DI+WI+RI+II	-	✓	✓

Table 5.1: Summary of ERT

Finally, we presented a set of policies which are required for each agent in making decisions for their interactions with the other agents. Six types of policies are introduced where four of them deal with the behavior of the agent regarding four interaction types and the remaining two deal with connection/disconnection of the given agent to other agents.

As the ERT model has been fully defined, it can be evaluated empirically. In the next chapter, we will explain our empirical experiments using the ERT model.

Chapter 6

Empirical Experiments

6.1 Introduction

This chapter provides an experimental evaluation of the agent model proposed in the previous chapter. Several experiments are proposed. These experiments are designed and conducted for three reasons: (1) demonstrating the vulnerability of existing trust and reputation models against individual-level attacks, and collusion attacks as explained in Section 5.3; (2) empirically showing the necessity of the requirements proposed in Section 5.4 for trust and reputation models and (3) demonstrating how the proposed agent model is resistant against the con-man attack, the witness-based collusion attack, and the report-based collusion attack. All the experiments have run on the environment model presented in Chapter 4.

The remainder of this chapter proceeds as follows. Section 6.2 describes the experimentally evaluated policies. We explain the experiments regarding the con-man attack (i.e., the vulnerability of the existing models, and the robustness of our proposed model) in Section 6.3. The collusion attacks experiments and the corresponding results are presented in Section 6.4. Finally, we summarize this chapter in Section 6.5.

6.2 Experimentally Evaluated Policies

This section describes policies used by experimentally evaluated agent types.

6.2.1 Direct Interaction Policies

Three kinds of DIPs used in our experiments are: Always Cooperate (AC), Always-Defect (AD)[1], Trust-based Tit-For-Tat (TTFT) and Con-Man (COM).

[1]Always Cooperate and Always Defect have been called unconditional cooperation and unconditional defection respectively in game theory literature.

Algorithm 2 TTFT Policy for Direct Interaction

```
{Suppose that agent i is executing this code }
for all j ∈ Neighborhood do
  if Ω_i^DIT < DIT_{i,j}(t) < ω_i^DIT then
    Tit-For-Tat(j)
  else
    if DIT_{i,j}(t) ≤ Ω_i^DIT then
      Defect(j)
    else
      if DIT_{i,j}(t) ≥ ω_i^DIT then
        Cooperate(j)
      end if
    end if
  end if
end for
```

Agents using the AC policy for their direct interactions will cooperate with their neighbors in direct interactions regardless of the action of their neighbor. In contrast, agents using the AD policy will defect in all neighbor interactions. Agents employing TTFT will start with cooperation and then imitate the neighbors' last move as long as the neighbors are neither trustworthy nor untrustworthy. Algorithm 2 shows the TTFT policy.

The COM policy follows the formal language presented in Section 5.3.1 which is solely dependent on the parameter θ. The Algorithm 3 demonstrates the COM(θ) policy, where θ is the input parameter. Using the COM(θ) policy, the con-man agent will defect after θ times of cooperation. After each defection, the con-man will again cooperate θ times.

6.2.2 Witness Interaction Policies

As explained in Section 5.8, there are three sub WIPs: Answering policy (AP), Querying Policy (QP) and Information-Gathering Policy (IGP).

Algorithm 3 COM(θ) Policy for Direct Interaction

{Suppose that agent i is executing this code.}
{θ is the input parameter.}
{$NI_{i,j}$ is the number of direct interactions of agent i with agent j}
for all $j \in Neighborhood$ **do**
  if $NI_{i,j}$ % $(\theta + 1) < \theta$ **then**
    Cooperate(j)
  else
    Defect(j)
  end if
end for

Answering policy (AP)

We have specified three kinds of answering policies in our experiments: Honest (Ho), Liar (Li), and Simpleton (Si). All of these sub-policies use the pseudo-code presented in Algorithm 4 while differentiating in the assignment of opinion variable (refer to * in Algorithm 4). The asterisk should be replaced by $DIT_{i,j}(t)$, "$-1 * DIT_{i,j}(t)$", or 1 for Honest, Liar, or Simpleton policy respectively.

Algorithm 4 Answering Policy

{Suppose that agent i is executing this code }
if receiving a witness request about j from k **then**
  $opinion = *$
  send $< OP, opinion, i, k, j >$ to k
  if $|opinion - DIT_{i,j}(t)| < DT$ **then**
    Send $< WIM, CWI, i, k, nil >$ to k after T_w time steps
  else
    Send $< WIM, DWI, i, k, nil >$ to k after T_w time steps
  end if
end if

The Honest policy always tells the truth to everyone. An agent employing the Liar

policy gives manipulated ratings to other agents by giving high ratings for untrustworthy agents and low ratings for trustworthy ones. The Simpleton policy always ranks all other agents as trustworthy. CWI/DWI will be sent based on whether the forwarding opinion is in contradiction with the internal trust value of an agent or not. If the difference between them is less than the Discrimination Threshold (DT), an agent will send CWI otherwise DWI is sent. In this sense, Liar always defects, Honest always cooperates, and Simpleton sometimes defects (by providing a high rating for untrustworthy agents) and sometimes cooperates (by providing a low rating for trustworthy agents) in providing the witness information. Note that DT is set to the value of 0.25 in our experiments.

Querying Policy (QP)

We have specified two kinds of Querying policies in our experiments: Regular (Reg), and Examiner (Ex).

Using the Regular policy presented in Algorithm 5, the agents ask for witness information from their neighbors regarding agents which are in the SIQ queue. SIQ contains a list of agents whose reputations should be investigated. After asking for witness information regarding a specific agent, the ID of that agent is inserted in the WIFQ queue. WIFQ contains the list of agents waiting for the result of an investigation. If an agent in the WIFQ is known as trustworthy in the context of WR, then the ID of that agent will be added to CAQ which contains the list of confirmed agents. The agents known as untrustworthy in terms of WR will be removed from WIFQ. If an agent in WIFQ is known neither as trustworthy nor as untrustworthy and the primitive $ShouldBeReInvestigated(k)$ returns a *true* value, then again the agent will request witness information from their neighbors regarding the given agent (the agent k). This primitive can be easily implemented relying on whether agent k remains in WIFQ for more than a specific amount of time.

The agent employing the Examiner policy assesses all of its neighbors in witness interaction by asking for witness information regarding one of the untrustworthy agents which has already interacted with the given agent. The result of this assessment might allow the agent to understand which neighbors are capable of detecting

Algorithm 5 Regular Querying Policy

```
{Suppose that agent i is executing this code}
{SIQ: a queue of should-be-investigated agents}
{CAQ: a queue of confirmed agents}
{WFIQ: Waiting-For-Investigation Queue}
if SIQ is not empty then
  k = dequeue(SIQ)
  {Ask for witness information about k from all neighbors}
  for all j ∈ Neighborhood do
    send < Inv, nil, i, j, k > to j
  end for
  enque(WFIQ, k)
end if
for all k ∈ WFIQ do
  if WR_{i,k} > ω_i^{WR} then
    enque(CAQ, k)
    remove(WFIQ, k)
  else
    if WR_{i,k} < Ω_i^{WR} then
      remove(WFIQ, k)
    else
      if ShouldBeReInvestigated(k) then
        {Ask again ratings of k from all neighbors}
        for all j ∈ Neighborhood do
          send < Inv, nil, i, j, k > to j
        end for
      end if
    end if
  end if
end for
```

untrustworthy agents and which are not. Algorithm 6 presents the Examiner policy.

Algorithm 6 Examiner Querying Policy

```
{Suppose that agent i is executing this code }
{BlackList: a list of known untrustworthy agents in terms of direct interactions}
if BlackList is not empty then
  Select randomly k from BlackList
  {Ask for witness information about k from all neighbors}
  for all j ∈ Neighborhood do
    send < Inv, nil, i, j, k > to j
  end for
end if
```

Information-Gathering Policy (IGP)

We have specified two information-gathering policies: DIT-based Weighted (DTW), and WIT-based Weighted (WTW). Both use Algorithm 7 while differentiating in the calculation of $WR_{i,k}$ (refer to * in Algorithm 7). DTW calculate it by using the formula presented in the Equation 6.1:

$$WR_{i,k}(t) = \frac{\sum_{j \in OpinionSenders} (\phi(DIT_{i,j}) \times Opinion(j,k))}{\sum_{j \in OpinionSenders} \phi(DIT_{i,j})} \tag{6.1}$$

whereas WTW use the formula presented in the Formula 5.15. In the Formula 6.1, the *OpinionSenders* variable includes indices of the neighbors of agent i who sent their ratings about agent k and $WIT_{i,j}$ is the current value of WIT variable of agent j from the perspective of agent i. Note that, $\phi(r)$ is a converter function as previously explained in Section 5.7.1.

6.2.3 Report Interaction Policies

As explained in Section 5.8, we have defined two sub RIPs: Reporting policy (RP), and Report-Gathering policy (RGP).

Algorithm 7 Information-Gathering Policy

{Suppose that agent i is executing this code}
if receiving $< Op, rating, j, i, k >$ **then**
  {if receiving opinion about k from j}
  Calculate $WR_{i,k}(t)$ based on *
end if

Reporting policy (RP)

We have specified three kinds of reporting policies in our experiments: Consistent (Co), Falsifier (Fa), and Blind Naive (BN). All of these sub-policies use the pseudo-code presented in Algorithm 8 while differentiating in how the $ManipulateReport^*()$ primitive is implemented for them. As illustrated in Algorithm 8, the agent i, at time step t, has access to the *Reports* variable which stores all the interactions of agent i that have taken place at time $t-1$. Then, the agents, based on its $ManipulateReport^*()$, will change (or not change) the interaction results and forward them to all of its neighbors. CRI/DRI will be sent after T_r time steps based on whether the forwarding report is in contradiction with the actual report or not. $ManipulateReport^*()$ for Consistent (Co) reporting policy does not modify the actual report while it changes cooperations to defections or defections to cooperations for Falsifier (Fa) reporting policy. In the Blind Naive (BN) reporting policy, $ManipulateReport^*()$ changes all defections to cooperations; in this way all interactions will be reported as successful interactions.

Report-Gathering Policy (RGP)

We have specified the Report-Gathering Policy presented in Algorithm 9 for our experiments. Using RGP, the agents confirm or reject the connection to the agents which are in the SIQ queue. SIQ contains a list of agents whose reputations should be investigated. As illustrated in Algorithm 9, after dequeuing the agent from SIQ queue, the ID of that agent is inserted in the WIFQ queue. WIFQ contains the list of agents waiting for the result of investigation. Upon receiving any reports from the neighbors the report-based reputation of the agent which participate in the reported

Algorithm 8 Reporting Policy

```
{Suppose that agent i is executing this code at time t}
{Reports: an array containing the reports of agent i's interactions at time t − 1}
for k = 1 to SizeOf(Reports) do
  sentReport = ManipulateReport*(Reports(k));
  for all j ∈ Neighborhood do
    send < RM, sentReport, i, j, nil > to j
    if sentReport = reports(k) then
      Send < RIM, CRI, i, j, nil > to j after T_r time steps
    else
      Send < RIM, DRI, i, j, nil > to j after T_r time steps
    end if
  end for
end for
```

interaction ($RR_{i,k}$) will be updated. If an agent in the WIFQ is known as trustworthy in the context of RR, then the ID of that agent will be added to CAQ which contains the list of confirmed agents. The agents known as untrustworthy in terms of RR will be removed from WIFQ.

6.2.4 Connection Policies

The three kinds of connection polices used in our experiments are: Conservative (C), Naive (N), and Greedy (G). There is an internal property for each of these policies called Socializing Tendency (ST) which dramatically affects decisions for making a connection request and the acceptance of the connection request. Both Naive and Greedy policies use the Algorithm 10 with different values for the ST variable.

According to Algorithm 10, any connection request from other agents will be accepted regardless of ST value but the agent will acquire unvisited agent IDs if its number of neighbors is less than ST.

Using the Conservative policy presented in Algorithm 11, the agents connect to confirmed agents regardless of the number of their neighbors. CAQ contains the list of agent IDs confirmed; this confirmation of an agent might be accomplished by a

Algorithm 9 Report-Gathering Policy

{Suppose that agent i is executing this code}
{SIQ: a queue of should-be-investigated agents}
{CAQ: a queue of confirmed agents}
{WFIQ: Waiting-For-Investigation Queue}
if SIQ is not empty **then**
  k = dequeue(SIQ)
  enque(WFIQ, k)
end if
if receiving $< RM, Reports, j, i, nil >$ **then**
  {suppose Reports contain the interaction between agent j and k}
  Update $RR_{i,k}(t)$ based on the Formula 5.13
end if
for all $k \in WFIQ$ **do**
  if $RR_{i,k} > \omega_i^{RR}$ **then**
    enque(CAQ, k)
    remove(WFIQ, k)
  else
    if $RR_{i,k} < \Omega_i^{RR}$ **then**
      remove(WFIQ, k)
    end if
  end if
end for

Algorithm 10 Naive and Greedy Policies

```
{Suppose that agent i is executing this code }
{CRQ is a queue containing the connection requests}
if CRQ is not empty then
  j = dequeue(CRQ)
  connectTo(j)
end if
if size(neighborhood) < ST then
  j = Get unvisited agent from Registry List
  if ∃j ≠ null then
    requestForConnectionTo(j)
  end if
end if
```

witness interaction policy. If the number of neighbors is less than ST, the agent connects to the agents requested or to an unvisited agent from the Registry List. Finally, if there are any agent IDs in CRQ (a queue of connection requests), the first agent ID will be inserted in SIQ (a list of agents whose reputations should be investigated). In this sense, the reputation of unknown agents will be investigated by witness interaction policies as explained in Section 6.2.2.

We set the value of ST to be 5, 15, and 100 for Conservative, Naive, and Greedy connection policies respectively.

6.2.5 Disconnection Policies

We have utilized three kinds of disconnection policies in our experiments: Lenient (Le), Moderate (Mo), and Strict (St). Using the Lenient policy, an agent will never drop any connections. As illustrated in Algorithm 12, an agent which uses the Moderate policy will disconnect from a neighbor known as an untrustworthy agent in terms of direct interaction. As shown in Algorithm 13, an agent with the Strict connection policy disconnects from a neighbor which is known to be untrustworthy either in direct interactions or in witness interactions.

Algorithm 11 Conservative Connection Policy

```
{Suppose that agent i is executing this code}
{CRQ: a queue of connection requests}
{CAQ: a queue of confirmed agents}
{SIQ: a queue of should-be-investigated agents}
if CAQ is not empty then
  j = dequeue(CAQ)
  connectTo(j)
end if
if size(neighborhood) < ST then
  if SIQ is not empty then
    j = dequeue(SIQ)
  else
    if CRQ is not empty then
      j = dequeue(CRQ)
    else
      j = Get unvisited agent from Registry List
    end if
  end if
  if ∃j ≠ null then
    connectTo(j)
  end if
end if
if CRQ is not empty then
  j = dequeue(CRQ)
  enque(SIQ,j)
end if
```

Algorithm 12 Moderate Disconnection Policy

{Suppose that agent i is executing this code }
if $DIT_{i,j}(t) \leq \Omega_i^{DIT}$ **then**
  disconnetFrom(j)
end if

Algorithm 13 Strict Disconnection Policy

{Suppose that agent i is executing this code }
if $DIT_{i,j}(t) \leq \Omega_i^{DIT}$ **then**
  disconnetFrom(j)
else
  if $WIT_{i,j}(t) \leq \Omega_i^{WIT}$ **then**
    disconnetFrom(j)
  end if
end if

6.3 Con-man Experiments

We here explain the experiments conducted and corresponding results of the con-man attack against three existing well-known trust models and our proposed con-resistant trust model. We have selected FIRE (recall Section 3.9.7), Regret (recall Section 3.9.6) and YS2000 (recall Section 3.9.1) as the representatives of the existing well-known trust models.

All simulations reported in this section were run with two agents, one trust-aware agent (TAA) which utilizes a specific trust model (e.g., Regret, FIRE, and YS2000) and a strategic con-man agent (SCA). Table 6.1 summarizes the specification of these two types of agent.

Name	Trust	DIP	WIP	CP	DP
TAA	FIRE/REGRET/YS2000/DIT-CRC	TTFT	-	N	Le
SCA(θ)	-	COM(θ)	-	N	G

Table 6.1: Agent Types and Specifications of Con-man Experiments

When the trust-aware agent uses Regret or FIRE as a trust model, the cooperation

and defection is mapped to 1 and -1 respectively and the value is used as an input of the trust model. In the case of using the Yu and Singh trust model (YS2000), cooperation and defection will be used directly for updating the of trust value. Each simulation is run for 400 time steps.

6.3.1 Con-man Attack Vulnerability Demonstration

Objective

The experiments reported here intend to demonstrate the vulnerability of three trust models (FIRE, Regret and YS2000) against the con-man attack presented in Section 5.3.1.

Settings

We ran three sets of 5 simulations in each of which a trust-aware agent using either FIRE, Regret, or YS2000 trust models interacts with one of the following strategic con-man agents: SCA(5), SCA(10), SCA(20), SCA(30), and SCA(40). The values of α and β for the Yu and Singh model were set to 0.05 and -0.5 respectively. These values are conservative, leading to trust being built up slowly and reduced quickly. We set $\lambda = \frac{-5}{Ln(0.5)}$ as proposed in the original research.

Results and Discussions

Figure 6.1 demonstrates the variation of the direct interaction trust value of TAA over the course of simulation for the Yu and Singh model. It is interesting that the con-man agent with $\theta > 10$ is eventually determined to be trustworthy from the perspective of the trust-aware agent. Although the magnitude of β is set at ten times that of α, which leads to a small improvement for a cooperation and a big drop for a defection, the con-man by choosing $\theta > 10$ is known as trustworthy in this trust model with this parameter setting. It can be shown that for each α and β, there is a θ_t that the con-man by choosing its SCA ($\theta > \theta_t$) will still be recognized as trustworthy despite being a con-man.

Figure 6.2 shows the direct interaction trust value variation of TAA over the 400

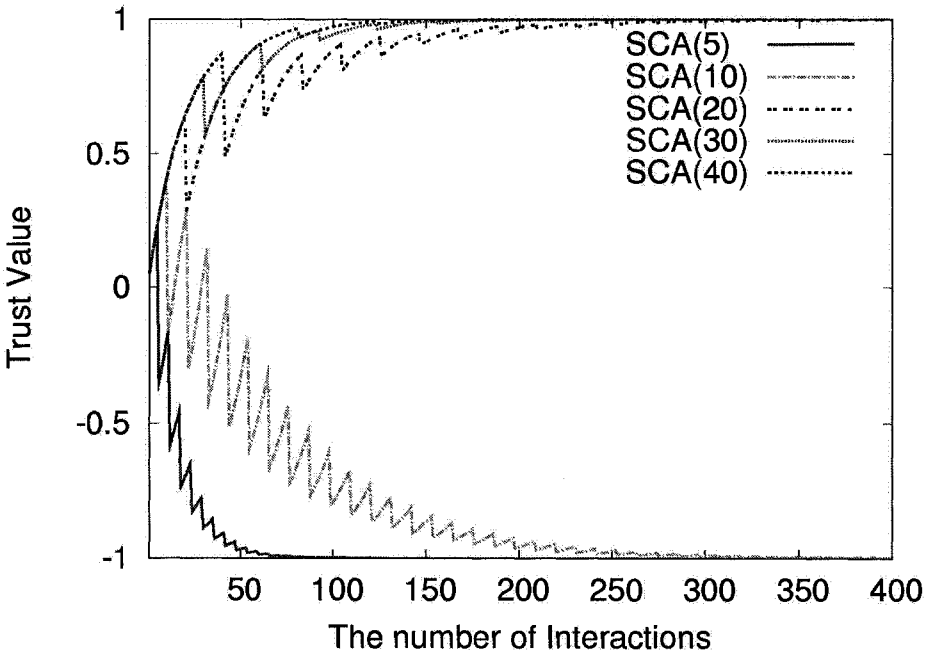


Figure 6.1: Exploitation of Yu & Singh model by a con-man.

interactions for the Regret model. It is clear that the con-man SCA(5) can stabilize its trust value at 0.66. Moreover, by increasing θ to 10, 20, 30 and 40, the con-man agent can reach a trust value of 0.81, 0.90, 0.93, and 0.95, which are high values of trust for a con-man; i.e., the agent is considered trustworthy.

Figure 6.3 depicts the variation of the trust value of TAA over the simulation for the FIRE model (the larger gray box magnifies part of the graph for clarity). Although FIRE is more sensitive to defection when compared to Regret as a result of its enhanced rating recency function (referring to Section 3.9.7), it is still vulnerable to the con-man attack. Obviously, the con-man SCA(5) can have trust value in the range of 0.56 to 0.73. Moreover, by increasing θ to 10, 20, 30 and 40, the con-man agent can ensure that its trust value will not fall below that of 0.67, 0.72, 0.73, and 0.74 respectively, while the maximum value is close to 1.

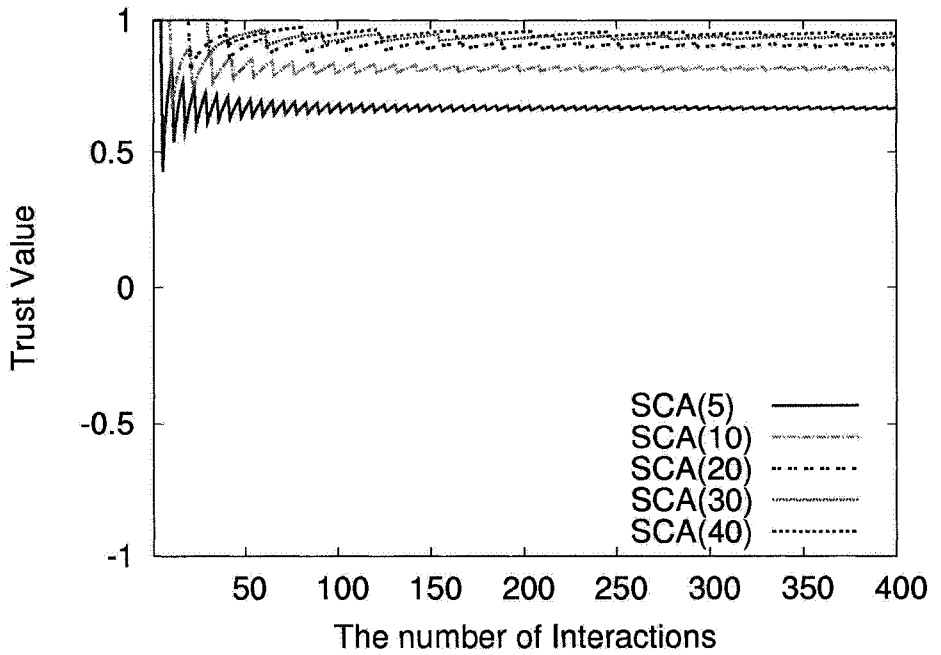


Figure 6.2: Exploitation of Regret model by a con-man.

6.3.2 Con-man Attack vs. DIT-CRC

Objective

The experiments reported here intend to check whether DIT-CRC (the combination of direct interaction trust and the proposed con-resistant component) is robust against the con-man attack presented in Section 5.3.1. In other words, we are interested in understanding whether the con-man will end up with a low trust value from the perspective of TAA agent when TAA uses DIT-CRC.

Settings

We ran 5 simulations in each of which a trust-aware agent using DIT-CRC variables interacts with one of the following strategic con-man agents: SCA(5), SCA(10), SCA(20), SCA(30), and SCA(40). The initial values of α and β (α_0 and β_0) were set to 0.05 and -0.5 respectively.

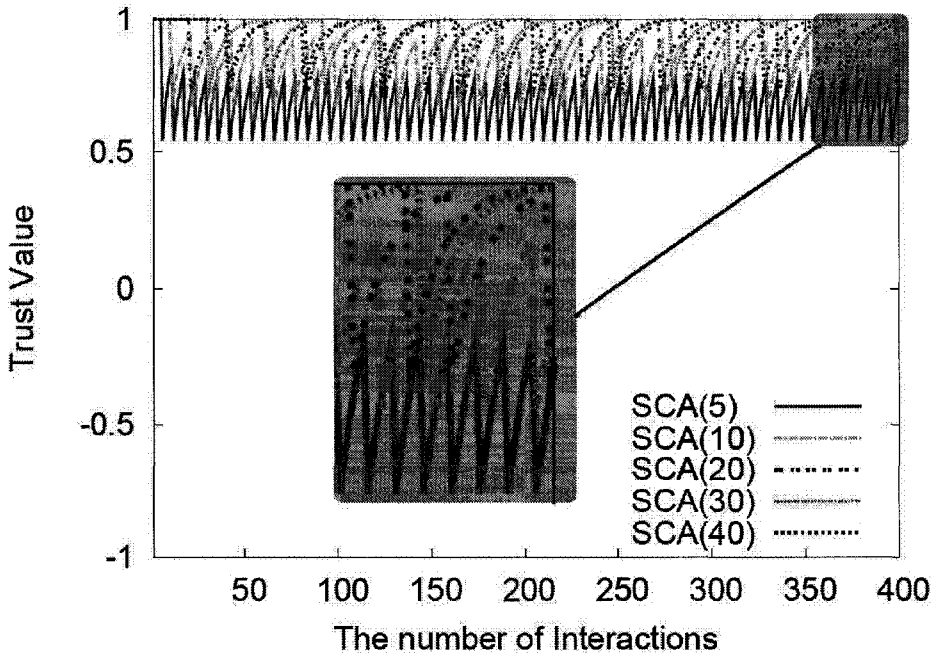


Figure 6.3: Exploitation of FIRE model by a con-man.

Results and Discussions

Figure 6.4 shows the trust value variation of the TAA over the 400 interactions. Interestingly, regardless of the value of θ for SCA(θ), the con-man was recognized by the trust model and achieved a low value of trust. These results show that our proposed con-resistant component is robust to the con-man attack. It is worth noting that the con-man still has a chance to be forgiven but with a very large number of cooperations and a change in its pattern of interaction. Figure 6.4 also shows that the speed of detection of the con-man is inversely proportional to θ.

6.3.3 α and β Updates

Objective

This experiment is designed to (1) determine whether or not the α and β update formulae presented in Section 5.6 are sensitive to the values of α_0 and β_0, and (2) provide better insight into whether there are some α_0 and β_0 which help the con-man remain undetected by the TAA agent using DIT-CRC.

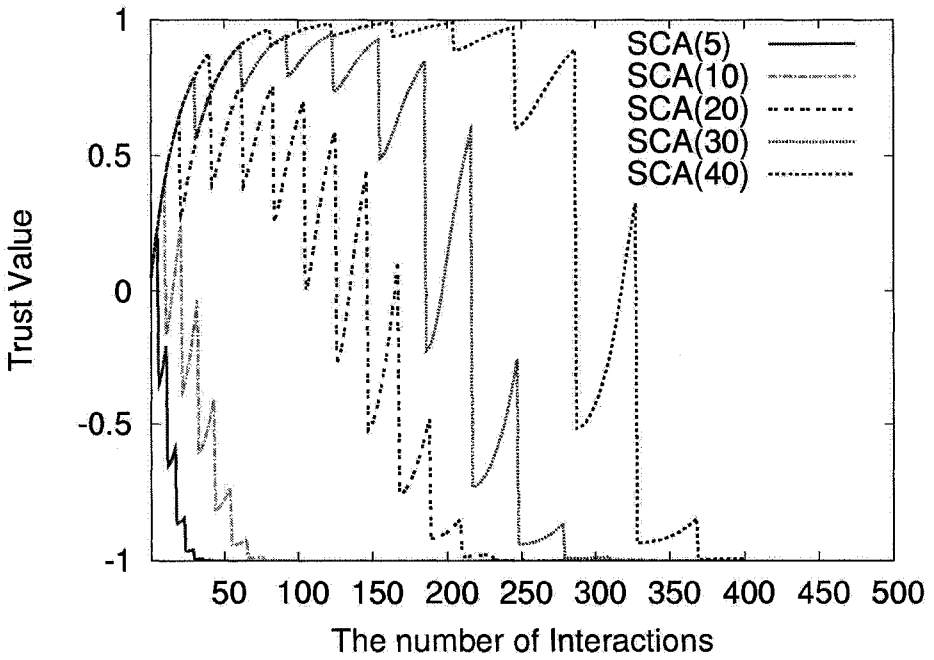


Figure 6.4: The reaction of DIT-CRC to the con-man attack.

Settings

We ran 5 simulations in each of which a SCA(20) agent interacts with the trust-aware agent using DIT-CRC with different initialization values of α_0 and β_0.

Results and Discussions

As presented in Table 6.2, not only did the trust-aware agent recognize the CA as an untrustworthy agent during the 400 interactions but also the final values of α and β for different initializations were close to each other. Similar final values for α and β support the hypothesis that α and β update formulae are insensitive to the values of α_0 and β_0.

Figure 6.5 Shows the variation of α and β parameters over the course of simulation in which $\alpha_0 = 0.05$ and $\beta_0 = -0.5$. By studying how the value of parameter β changes through time, we can observe that this value decreases with each defection and consecutive cooperations do not increase it. The value of α decreases after each defection and consecutive cooperations can increase it to the initial value. When the number of defections increases, a greater number of consecutive cooperations is

	$\alpha_0 = 0.20$ $\beta_0 = -0.2$	$\alpha_0 = 0.15$ $\beta_0 = -0.3$	$\alpha_0 = 0.10$ $\beta_0 = -0.4$	$\alpha_0 = 0.05$ $\beta_0 = -0.5$
α	0.00003	0.00002	0.00002	0.00005
β	-0.99983	-0.99984	-0.99981	-0.99893

Table 6.2: Final Values of α and β after 400 interactions of the trust-aware agent with SCA(20).

necessary to increase the value of α to its initial value, α_0.

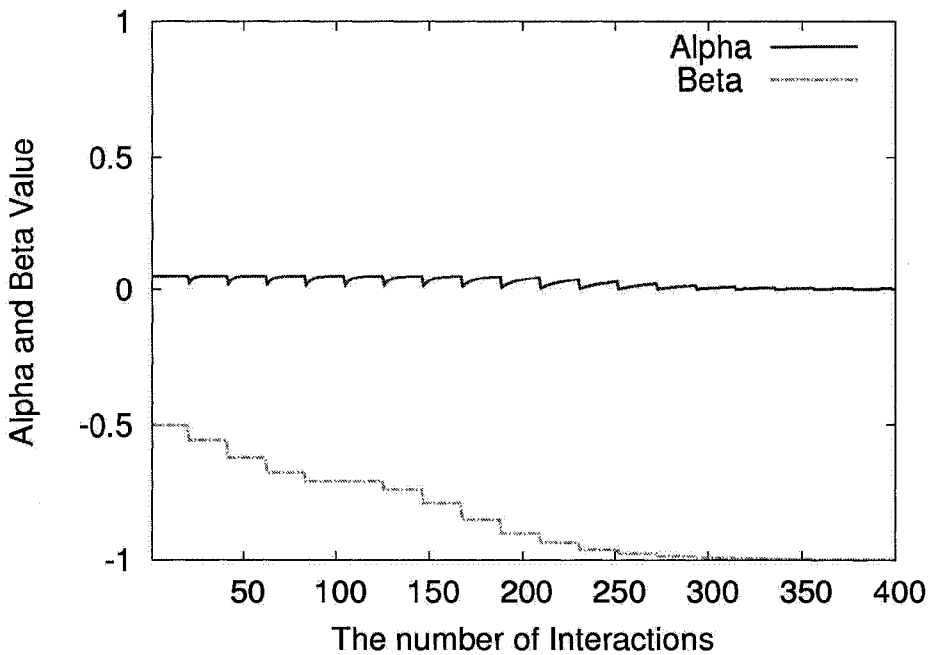


Figure 6.5: Alpha and beta values over the course of simulation

6.4 Collusion Experiments

This section describes the experiments conducted and results obtained for Witness-based collusion. We have modeled the witness-based collusion attacks by using Naive and Malicious agents as explained in Section 5.3.2. In addition to these two types of agents, the agent society includes Trust-Aware agents which are equipped with perception variables (trust and reputation variables) to assess trustworthiness of others and with policies to properly interact with others. We have defined two classes of

Trust-Aware agents for our experiments: Trust-Aware(TA) and Trust-Aware$^+$(TA^+) where TA uses a unidimensional trust model as opposed to TA^+ which uses a multi-dimensional trust model. Both of TA and TA^+ have two variations by using a reputation mechanism (the subscript of w will be used in this case) or not using a reputation mechanism. Therefore, we have defined the following variations of Trust-aware agents: TA, TA_w, TA^+ and TA^+_w. Table 6.3 presents all agent types used in the experiments reported in this section.

Name	Naive	Malicious	TA	TA_w	TA^+	TA^+_w	TA^+_r
Trust	-	-	DIT	DIT	DIT&WIT	DIT&WIT	DIT&RIT
DIP	AC	AD	TTFT	TTFT	TTFT	TTFT	TTFT
CP	N	G	N	C	C	C	C
DP	Le	Le	Mo	Mo	St	Mo	Mo
AP	Si	Li	Ho	Ho	Ho	Ho	-
QP	-	-	-	Reg	Ex	Reg	-
IGP	-	-	-	DTW	-	WTW	-
RP	BN	Fa	-	-	-	-	Co
RGP	-	-	-	-	-	-	RGP

Table 6.3: Agent Types and Specifications of Collusion Experiments

6.4.1 Unidimensional Trust and Non-collusive Agent Society

Objective

The objective of this experiment is to understand if the DIT variable and policies used by TA agents can help them identify each other, keep connections to each other and stay in equilibrium with each other (continue cooperating with each other) while they disconnect Malicious agents from the society where there is no collusion in the agent society. In other words, the intention is to understand whether cooperation emerges between TA agents while they isolate themselves from Malicious agents in a non-collusive agent society.

Settings

We run the simulation with the population size of 200 agents where TA agents cover 66% of population and the rest are Malicious agents. The simulation is run for 400 time steps.

Results and Discussions

Different stages of this simulation are depicted in Figure 6.6, where TA agents and Malicious agents are in green (light gray in white-black print) and in black respectively. Initially, there are no connections between agents so they make requests to connect with each other and accept/reject the connection requests of others based on their connection policy.

As shown in Figure 6.6b, the society of agents does not exhibit any special structure at time step 20 while agents are connected to each other without any discernible pattern.

At time step 60 (Figure 6.6c), two groups of TA and malicious agents are distinguishable. TA agents have dropped the connections to Malicious agents which are known as untrustworthy agents from their perspective. Malicious agents are still trying to connect to TA agents because of their Greedy connection policy.

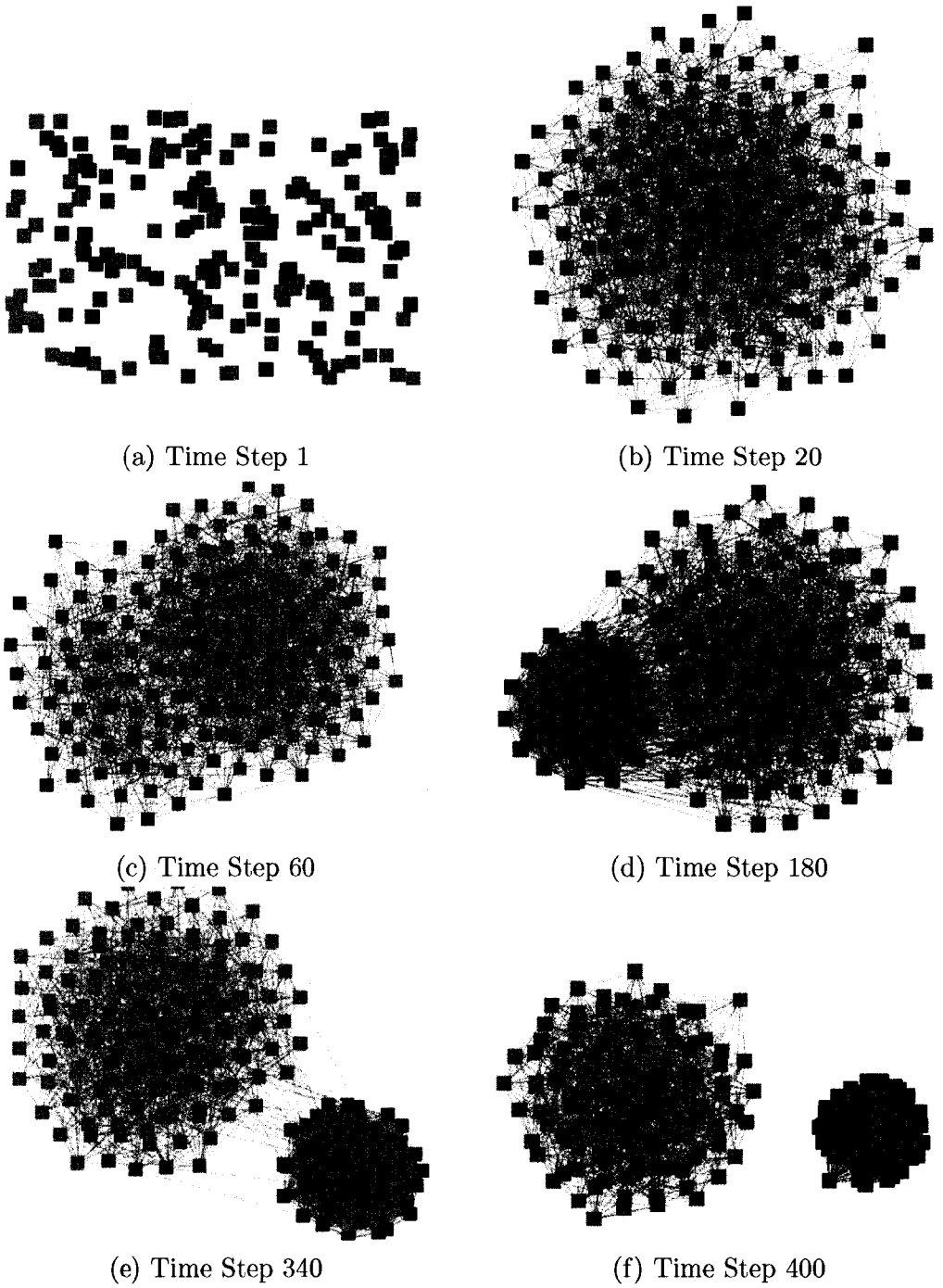
As the connections of Malicious agents with TA agents will be dropped, Malicious agents are more connected to each other (Figure 6.6d). The connections between Malicious agents and TA agents diminish when Malicious agents have tried all of the agents in the society and TA agents have dropped those connections (referring to Figure 6.6e).

Finally, we have two isolated groups of Trust-Aware and Malicious agents, where a member of each group is interacting only with peers in the same group as illustrated in Figure 6.6f.

As demonstrated in this experiment, cooperation emerges between TA agents using the specified trust variables and policies. Moreover, Malicious agents get isolated from the agent society in which there is no collusion.

6.4.2 Unidimensional Trust and Witness-based Collusion

We herein describe two experiments which are conducted to demonstrate the vulnerability of an unidimensional trust model against witness-based collusion attacks. We have analyzed the effect of this attack at the macro-level and micro-level for the two following experiments.



(a) Time Step 1

(b) Time Step 20

(c) Time Step 60

(d) Time Step 180

(e) Time Step 340

(f) Time Step 400

Figure 6.6: Structural changes in the Agent Society

Objective

The objective of this experiment is to show TA agents using unidimensional trust (only DIT) are unable to detect naive agents which are colluding with malicious agents. We will show that TA agents are incapable of separating themselves from colluding agents.

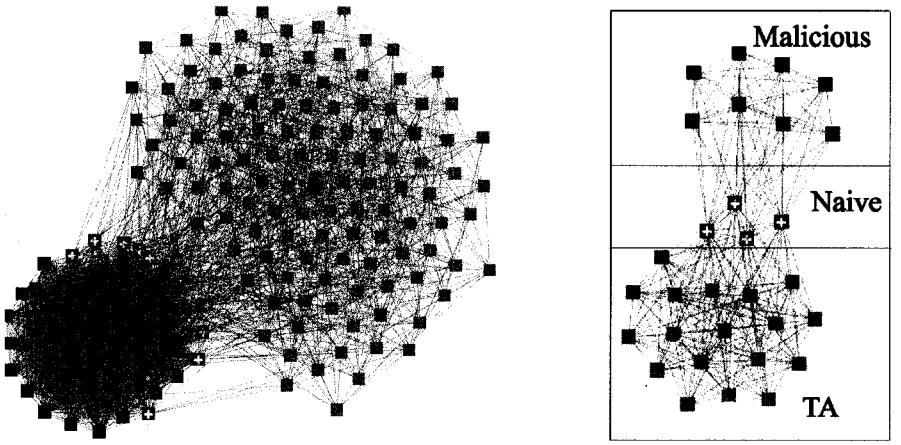
Settings

We run 200 agents where 55%, 11% and 34% of population are TA, Naive and Malicious agents respectively. The simulation is run for 400 time steps.

Results and Discussions for TA agents

The structure of the agent society after 400 time steps is presented in Figure 6.7a. Malicious and Trust-Aware agents are shown with the same colors as used in the previous experiment and blue squares with white "+" represent Naive agents. With the introduction of Naive agents, we could not achieve the separation of Malicious and TA agents reached in the previous experiment. Since TA agents perceived Naive agents as trustworthy agents in direct interaction so they maintain their connections with Naive agents. On the other hand, since Naive agents accept all connection requests and do not drop any connection, they will be exploited by Malicious agents. As illustrated in Figure 6.7a, TA agents are connected indirectly to Malicious agents by means of Naive agents. In order to show the transitive relationship between Malicious and TA agents, a simulation with only 30 agents was run. Figure 6.7b clearly shows the Naive agents acting as a buffer between the 2 other agent communities.

Figure 6.8 shows the $\overline{U}$ of each agent type over the course of the simulation. $\overline{U}_{TA}$ is increasing over the course of the simulation with small fluctuations. These fluctuations are the result of the connections of Malicious agents to TA agents, and especially their first interactions in which Malicious agents defect and the TA agents cooperate, thus resulting in a sucker payoff for TA agents. The closer $\overline{U}_{TA}$ gets to 3, the higher the proportion of interactions of TA agents are mutual cooperation. $\overline{U}_{Malicious}$ is increasing due to connecting to more Naive agents. The $\overline{U}_{Naive}$ drops over the course of simulation since the number of their connections with Malicious



(a) 200 Agents (b) 30 Agents

Figure 6.7: The Final Society Structure

agents increases. All three graphs stabilize before time step 350, which is the result of not establishing new connections by any agents. Not requesting any connections can be the result of reaching the ST threshold (e.g., Naive and Trust-Aware) or scanning all of the agents (e.g., Malicious agents).

Objective

This experiment intends to demonstrate that TA_w agents are incapable of decreasing the impact of naive agent's rating on their decisions. In this sense, the used reputation mechanism will not help TA_w decrease the encounter risk.

Settings

We have run two simulations of 200 agents for this experiment, in each of which Naive, Malicious agents are 44% and 34% of the populations respectively. The remainder of the population (22%) is either TA or TA_w. The TA_w agents benefit from using the Conservative connection policy and witness interaction policies for inquiring about the trustworthiness of the connection requester from neighbors. The simulation is run for 400 time steps.

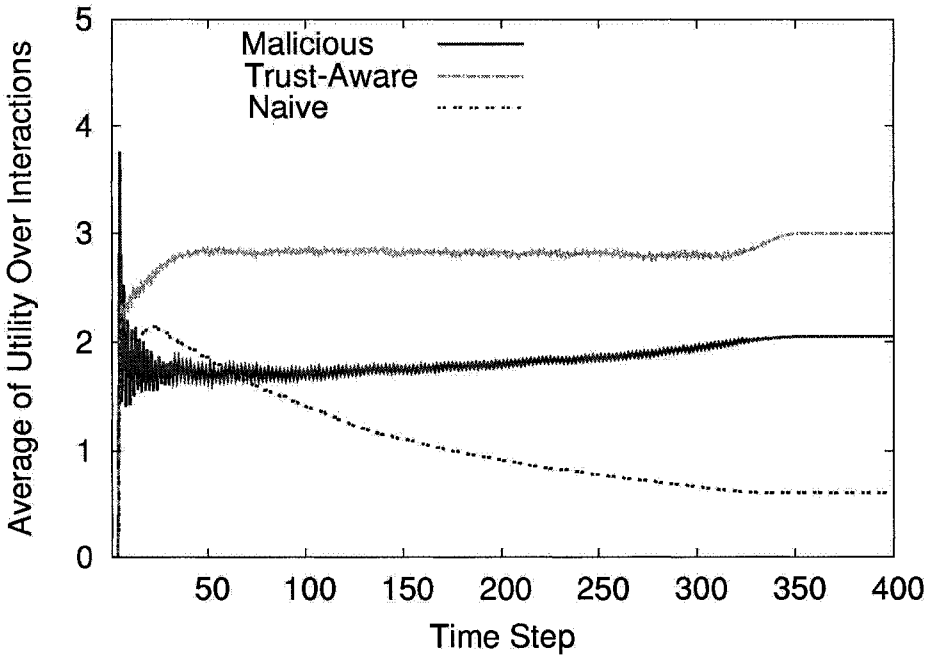


Figure 6.8: $\overline{U}$ of agent types over simulation

Results and Discussions

Figure 6.9 illustrates $\overline{D}$ of TA, and TA_w over the course of simulation. TA_w agents have almost the same average number of dropped connections when compared to the TA agents. In this sense, TA_w and TA agents expose themselves to almost the same level of risk of being exploited by malicious agents, although TA_w uses the Conservative connection and DTW information-gathering policies. This is due to the fact that each TA_w agent is surrounded mostly by Naive agents, resulting in receiving more inaccurate opinions about other malicious agents while the senders of all opinions are trustworthy in terms of direct interactions.

6.4.3 Population Proportion of Naive Agents

Objective

This experiment intends to show the effect of population proportion of Naive agents on the agent society (other types of agents such as Malicious and TA).

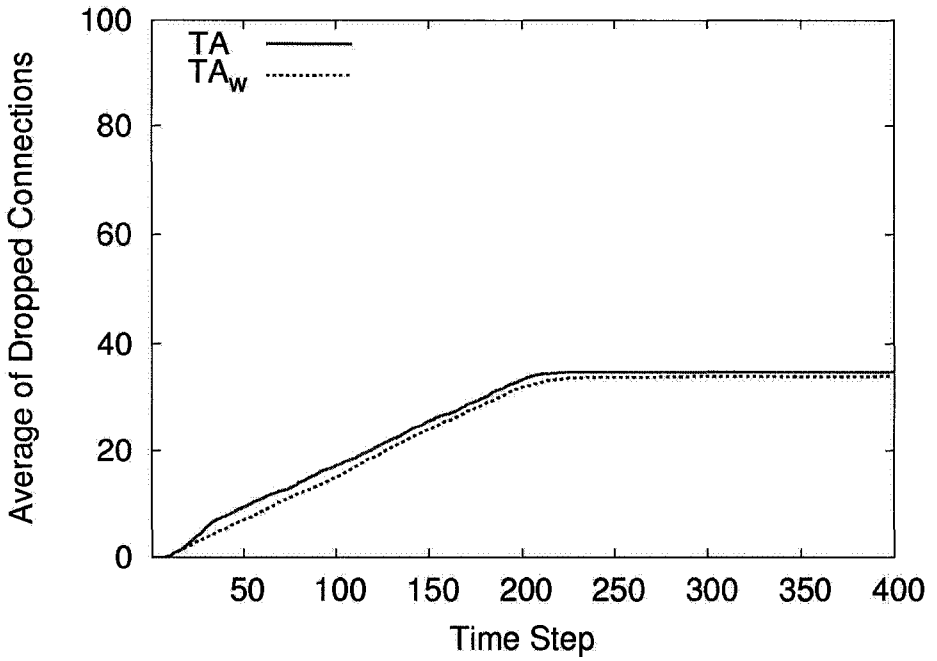


Figure 6.9: $\overline{D}$ of agent types over simulation

Settings

We have run five simulations of 200 agents with different proportions of Naive and Trust-Aware agents while keeping the proportion of Malicious agents unchanged as shown in Table 6.4. The simulations are run for 400 time steps.

Agent Type	Population				
	Pop1	Pop2	Pop3	Pop4	Pop5
Malicious	34%	34%	34%	34%	34%
Naive	0%	11%	22%	33%	44%
Trust-Aware	66%	55%	44%	33%	22%

Table 6.4: Population Distributions of Experiment 3

Results and Discussions

Figure 6.10 presents $\overline{U}$ of each agent type at time step 400 for each of the runs. By increasing the proportion of Naive agents, $\overline{U}_{Malicious}$ will be increased considerably although the proportion of Malicious agents remains unchanged. $\overline{U}_{TA}$ in all runs stays

at 3 indicating that the proportion of Naive agents does not influence $\overline{U}_{TA}$. $\overline{U}_{Naive}$ increases slightly. This is because Malicious agents have more choices to connect to Naive agents and to satisfy their ST threshold. For Pop5, the $\overline{U}_{Malicious}$ is higher than two other types. As a consequence, in such societies, there is no incentive to be a Trust-aware agent since Malicious agents have better utility; that is the outcome of having a high proportion of Naive agents in the society.

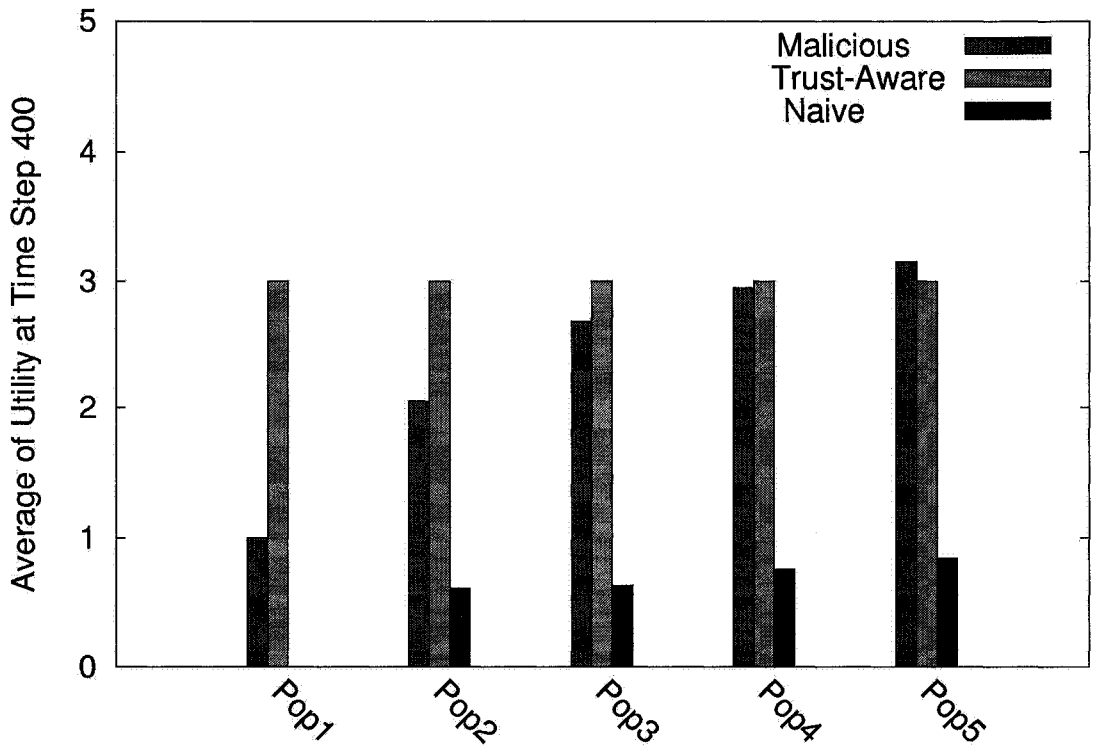


Figure 6.10: $\overline{U}$ for five runs.

6.4.4 Multi-dimensional Trust and Witness-based Collusion

We herein explain two experiments which are conducted to demonstrate the benefit of using multi-dimensional trust where there are the witness-based collusion attacks.

Objective

The intention behind this experiment is to show that the TA^{+} agent class, by using the multi-dimensional trust variables (DIT and WIT) and the appropriate set of

policies, can detect both naive and malicious agents (colluding group) and separate themselves from them.

Settings

We run 200 agents where 55%, 11% and 34% of the population are Trust-Aware$^+$ (TA$^+$), Naive and Malicious agents respectively. The simulation is run for 400 time steps.

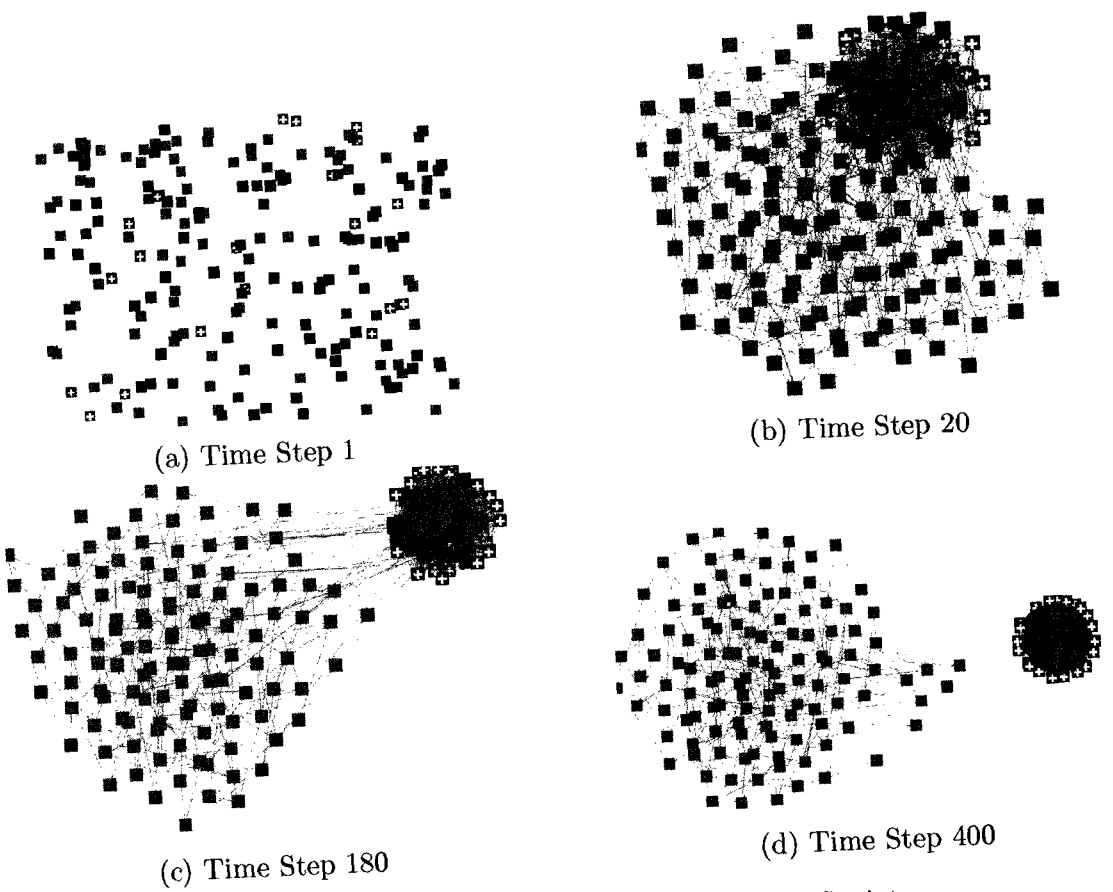
Results and Discussions

The structure of the agent society at three points in the simulation is presented in Figure 6.11. Malicious and Naive agents are shown with the same colors as used in previous experiments and TA$^+$ agents are presented in green. It is interesting to observe that Naive and Malicious agents are isolated from the TA$^+$ agents. By using multi-dimensional trust (DIT and WIT) and the Strict disconnection policy, TA$^+$ agents could identify both Malicious and Naive agents to isolate them from their community. Naive agents are detected based on their failure to provide the appropriate witness information while Malicious agents are recognized by their defections in direct interactions.

This experiment shows how by addition of another dimension of trust (i.e., WIT) to direct interaction trust, naive agents are distinguishable. In this sense, TA$^+$ agents assessed the ability of their neighbors in detecting malicious agents. Those agents which fail in this assessment turn out to be naive agents.

Objective

The intention behind this experiment is to show that TA_w^+ agents by using multi-dimensional trust and appropriate witness interaction policies (e.g., WTW) can decrease the impact of naive and malicious agents (colluding groups) on aggregating the ratings. As a result, the TA_w^+ agents can decide more reliably regarding the trustworthiness of other agents and expose themselves to a lower level of risk.



(a) Time Step 1

(b) Time Step 20

(c) Time Step 180

(d) Time Step 400

Figure 6.11: Structural Changes of Agent Society

Settings

We have run two simulations of 200 agents for this experiment, in each of which Naive, Malicious agents are 44% and 34% of the populations respectively. The remainder of the population (22%) is either TA_w, or TA_w^+. Both TA_w and TA_w^+ benefit from using the Conservative connection policy and witness interaction policies for inquiring about the trustworthiness of the connection requester from neighbors. Note that, these two types employ various witness information-gathering policies. The simulation is run for 400 time steps.

Results and Discussions

As shown in Figure 6.12, TA_w^+ agents have considerably fewer dropped connections when compared to TA_w. Policies used by this agent type result in successful acceptance/rejection of connection requests. In this sense, TA_w^+ agents expose themselves to smaller numbers of untrustworthy agents and consequently lower the level of risk of being exploited by these agents.

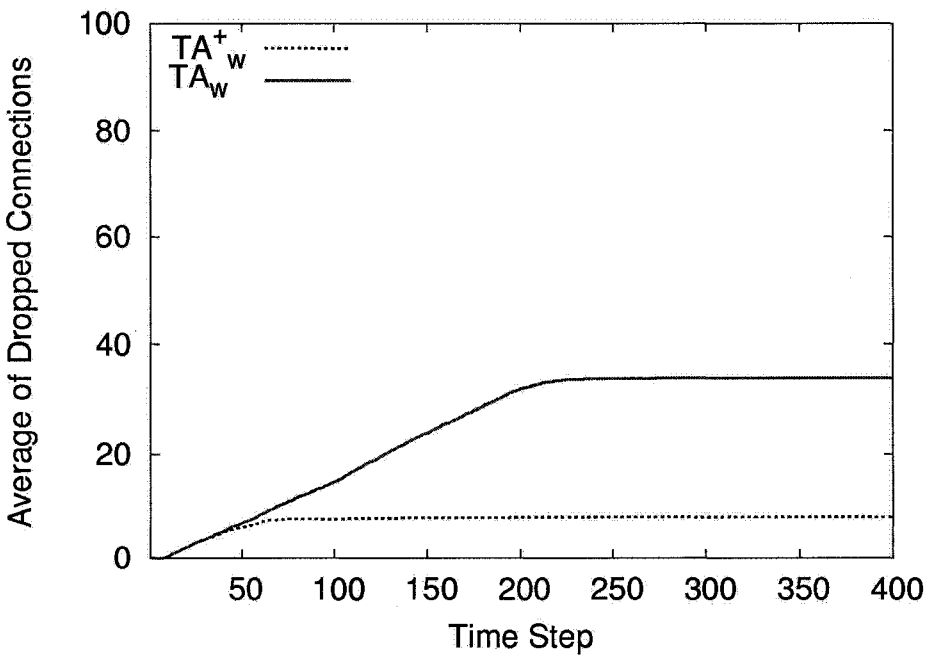


Figure 6.12: $\overline{D}$ of agent types over the simulation

Figure 6.13 illustrates $\overline{U}$ for TA_w and TA_w^+ types over the course of the simulations.

$\overline{U}_{TA_w^+}$ reaches the value of 3 faster for TA_w^+ than TA_w and will not fall below it later. This is evidence of the learning capability of TA_w^+ agents especially by using WIT for aggregating opinions in witness interaction policies. Each TA_w^+ agent, by updating WIT, will learn which of its neighbors are trustworthy in terms of witness information and then weight their opinions based on their WIT which is completely independent of DIT. As a result, false opinions of neighbors cannot mislead them several times whereas TA_w agents can be deceived several times by false opinions from the same neighbors (naive agents) because of the lack of this trust dimension.

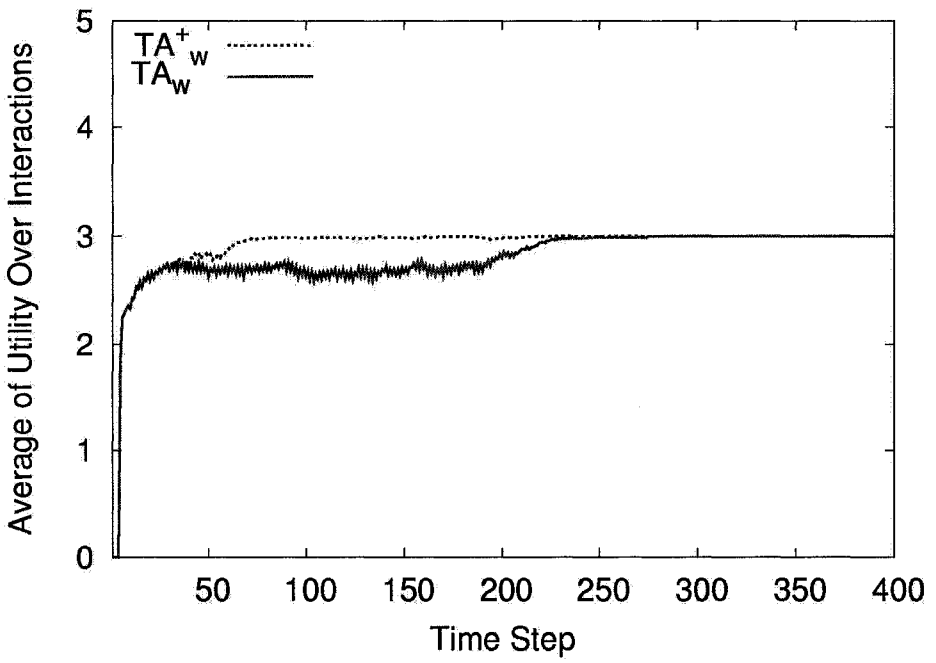


Figure 6.13: $\overline{U}$ of agent types over the simulation

6.4.5 Multi-dimensional Trust and Report-based Collusion

The intention behind this experiment is to show that TA_r^+ agents by using multi-dimensional trust and appropriate reporting interaction policies can decrease the impact of naive and malicious agents (colluding groups) on report-based reputation. We are interested in comparing the efficacy of witness-based reputation which works based on witness interaction and report-based reputation which works based on the reporting interaction presented in this thesis. For this purpose, we compare TA_r^+

agents with TA_w^+ agents. In this regard, those agents which decide more reliability regarding the trustworthiness of other agents and expose themselves to a lower level of risk have better efficacy.

Settings

We have run two simulations of 200 agents for this experiment, in each of which Naive, Malicious agents are 44% and 34% of the populations respectively. The remainder of the population (22%) is either TA_w^+, or TA_r^+. Both TA_r^+ and TA_w^+ benefit from using the Conservative connection policy while TA_r^+ uses report-based reputation and related report interaction policy for approval of connection to unknown agents as opposed to TA_w^+ which utilizes witness interaction policies for inquiring about the trustworthiness of the connection requester from neighbors. The simulation is run for 400 time steps.

Results and Discussions

As shown in Figure 6.14, TA_r^+ agents have fewer dropped connections when compared to TA_w^+. Policies used by this agent type result in more successful acceptance/rejection of connection requests. In this sense, TA_r^+ agents expose themselves to smaller numbers of untrustworthy agents and consequently lower the level of risk of being exploited by these agents.

Figure 6.15 illustrates $\overline{U}$ for TA_r^+, and TA_w^+ types over the course of the simulations. $\overline{U}_{TA_w^+}$ and $\overline{U}_{TA_r^+}$ are almost identical. It can be seen that TA_r^+ and TA_w^+ have the same utility over the course of simulation but have lower risk of exposure to untrustworthy agents. We hypothesize that this is the effect of having access to more evidence (reports) for judging the trustworthiness of information providers (reporters) in reporting interactions when compared to the witness interactions.

6.5 Summary

We empirically analyze the utility of our proposed model and the vulnerability of existing trust and reputation models through the experiments reported in this chapter.

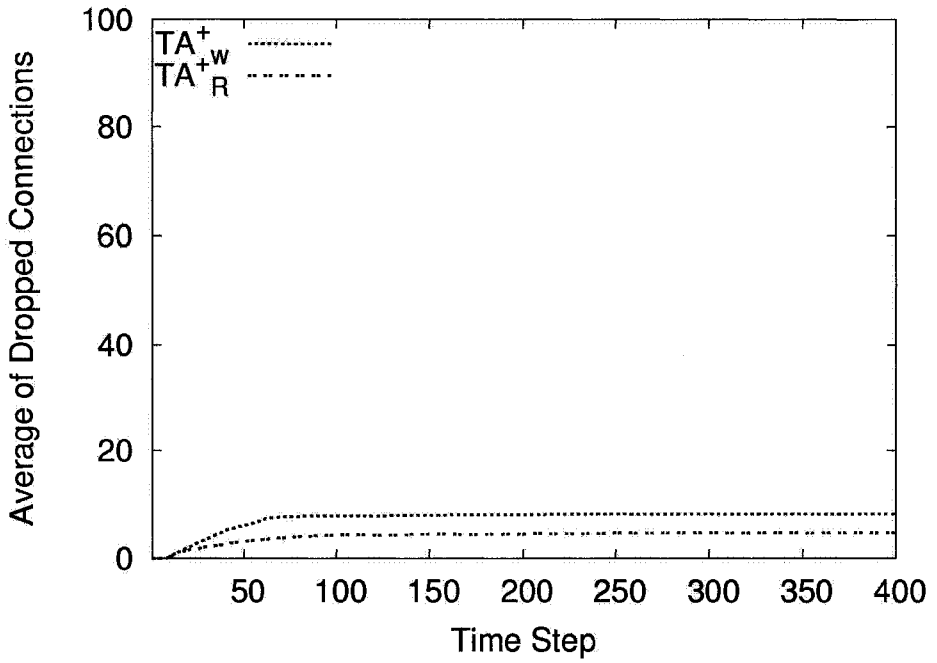


Figure 6.14: $\overline{D}$ of agent types over the simulation

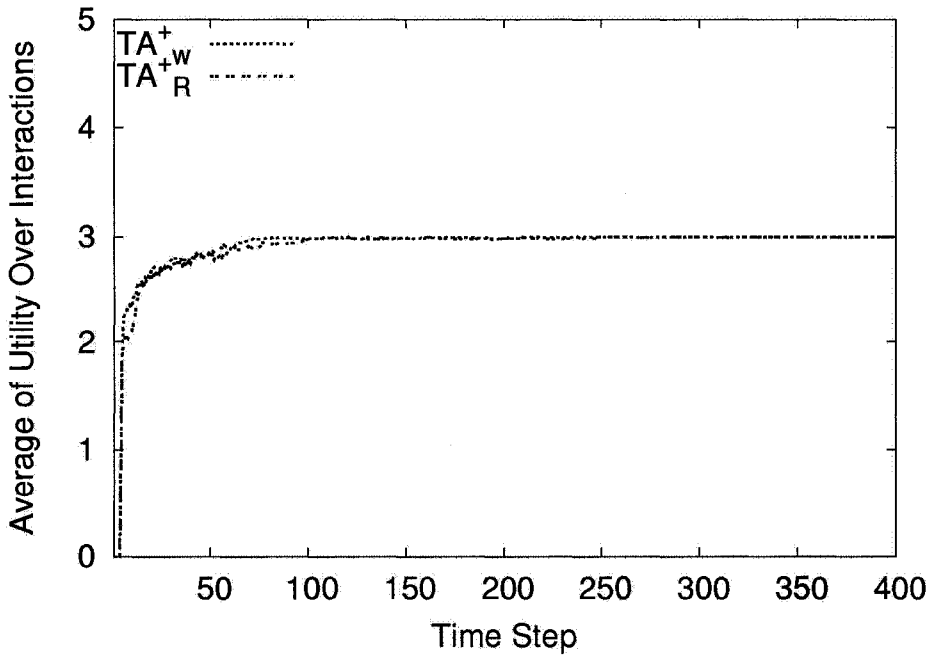


Figure 6.15: $\overline{U}$ of agent types over the simulation

We applied the con-man attack introduced and modeled in this thesis to direct trust components of trust models. We experimentally demonstrated how a con-man can exploit three well-known trust models YS2000 [78], Regret, and FIRE such that he/she is still known as trustworthy after repeated cycles of interaction while conning others. Moreover, our con-resistant component – as explained in Section 5.6 – shows robustness against the con-man attack. We demonstrated how the update formula of the con-resistant component is insensitive to the initial values of α and β.

The isolation of untrustworthy agents (malicious agents) from the society of agents is considered as one of the main research objective of trust models [78]. We demonstrate that the isolation of malicious agents from the agent society can easily be achieved based on the direct interaction trust of agents where the society of agents is solely composed of malicious and trust-aware agents. On the other hand, with the introduction of Naive agents, the policies and trust variables used for trust-aware agents (TA) did not work in isolating malicious agents from the society. We empirically provided an insight into how the proportion of naive agents affects the utility of malicious agents. When this proportion exceeds some threshold, malicious agents have the best utility in the society and consequently there is no incentive for trust-aware agents to stay trustworthy. In contrast, they are motivated to be malicious to exploit naive agents in a way similar to malicious agents.

We experimentally show how by addition of another dimension of trust (i.e., WIT) to direct interaction trust, naive agents (colluding agents) are distinguishable. In this sense, TA^+ agents assessed the ability of their neighbors in detecting malicious agents. Those agents which fail in this assessment turn out to be naive agents. Furthermore, TA_w^+ agents by using WIT could weight the rating of naive agents and decrease the impact of them in their final assessment. This results in exposing themselves to lower level of risk in their interactions. We show that TA_r^+, by using reporting interaction, multi-dimensional trust, and an appropriate set of policies have lower risk of exposure to untrustworthy agents than TA_w^+ agents.

In the next chapter, we will summarize the key messages of this thesis and discuss possible directions that future work might take.

Chapter 7

Conclusions and Future Work

This thesis is motivated by the dire need for trust and reputation models in artificial societies and open distributed environments, especially e-commerce. While reviewing important existing trust and reputation models from the literature as presented in Chapter 3, embracing centralized and decentralized models, we have noted a tendency to focus on exploitation of trust and reputation models. As a result, we have noted the exposure of such models to individual and collusion attacks (see Section 3.5). These vulnerabilities reinforce the need for new evaluation criteria for trust and reputation models called exploitation resistant which reflects the ability of a trust model to be unaffected by agents who try to manipulate the trust model.

This chapter is organized as follows. Section 7.1 summarizes the key messages of this thesis while highlighting the contributions claimed in Chapter 1 and presented in previous chapters. We suggest possible directions of future work that might be followed in Section 7.2. Finally, we briefly summarize the thesis in Section 7.3.

7.1 A summary of Key Messages

7.1.1 Environment Model of ERT

We introduced a decentralized game-theoretic trust and reputation environment model (testbed). The proposed environment model is compatible with the characteristics of open distributed systems as mentioned in Section 1.1. As explained in Chapter 4, agents can have different types of interactions and consequently access to different sources of information for assessment of other agents.

The proposed environment model provides the facility to define agents with various behaviors and is flexible enough to accommodate a variety of adversarial behaviors. The proposed environment model does not suffer from the shortcomings of existing

testbeds (see Section 3.6) in spite of maintaining the simplicity of game theoretic models.

7.1.2 Reporting Interaction

Besides direct, witness interaction and introduction interactions, agents in our environment model can have a novel type of interaction called the reporting interaction as explained in Section 4.2. This interaction type and its associated trust dimension – which are introduced in this thesis – facilitates the decentralized reporting mechanism in distributed environments.

7.1.3 Agent Model of ERT

We proposed the Exploitation-Resistant Trust (ERT) agent model in Chapter 5 which incorporates multiple sources of information to assess the trustworthiness of agents. This model, besides possessing the five characteristics of trust and reputation models presented by Fullam et. al [26] (i.e., multi-dimensional, quick convergence, precise modeling, adaptivity and efficiency), is exploitation resistant. By exploitation resistance we mean that adversaries cannot take advantage of the trust model and its associated systems parameters even when they are known or partially known to adversaries.s

7.1.4 The Con-man Attack and Con-resistant Trust Models

In Section 3.5, we categorize the possible exploitations into two groups: individual attacks and collusion attacks. Individual attacks are concerned with attacks that are mounted by only one agent while the collusion attacks are usually mounted by the collaboration of a group of agents.

We introduced a new type of individual attack called the con-man attack [65]. In the con-man attack, a con-man usually takes advantage of someone else and attempts to defraud that person by gaining their confidence. We formally model the con-man attack in Section 5.3.1 and empirically demonstrate the vulnerability of three well-known trust models (FIRE, Regret, and YS2000) against this exploitation in Section 6.3.1.

The desirable characteristics of con-resistant trust models are discussed in Section 5.4.1. We have introduced two heuristics of con-resistant trust models: first, cautiously increment trust after having seen any defection and second, larger punishments after each defection. Based on these characteristics, a con-resistant component for ERT is proposed in Section 5.6. We empirically show that the proposed con-resistant component can successfully detect a con-man and considers it as an untrustworthy agent by assigning a low trust value to it (see Section 6.3.2).

7.1.5 Witness-based Collusion Attacks and Multidimensional Models

In Section 5.3.2, we modeled a witness-based collusion attack as a representative of collusion attacks through the introduction of the concept of a naive agent. Moreover, we provided a real example of naive entities in e-commerce and other domains [64].

We show the vulnerability of unidimensional trust models against witness-based collusion attacks (naive agents) in Section 6.4.2. We empirically analyzed the impact of the naive agent class on agent society in Section 6.4.3. The experiments provide an insight into how the proportion of naive agents affects the utility of malicious agents. When this proportion exceeds some threshold, malicious agents have the best utility in the society and consequently there is no incentive for trust-aware agents to stay trustworthy.

We proposed strategies for dealing with witness-based collusion attacks in Section 6.4.4. We found that trust and reputation models need to be multi-dimensional in order to be resistant against collusion attacks (see Section 6.4.4). Moreover, we show that trust-aware agents needs multi-dimensional trust models to separate malicious and naive agents from the trustworthy community.

7.2 Future Work

Future work is planned in the following 4 areas: (1) the extension of environment and agent models of ERT by introducing new types of interactions and information sources, (2) further exploration of reporting interactions and introduction interaction, (3) exploration of newcomers impact on society, and (4) development of richer and more varied attacks that attempt to exploit one or more information sources and

proposing solutions for them.

We plan to extend the ERT agent and environment models by using other sources of information such as sociological information. In this sense, the agents possess different roles in society, and will be judged based on their roles and relationships with others. Note that, this extension should still be consistent with open distributed characteristics as explained in Chapter 1 which makes it a significant practical and research challenge. Indubitably, designing a decentralized mechanism for providing sociological information is the main challenge. Furthermore, in a practical system trust modification would include a consideration of the value of a transaction.

In the design of the environment and agent models of ERT proposed in this thesis, we ignore the existence of noise in the perception of interactions. For example, in our model, when an agent perceives a direct interaction of another agent (target agent) as a defection, the target agent is penalized by the receiver agent regardless of the fact that whether this defection was the result of noise. We are interested in considering the effect of noise in our model in future research.

We introduce the reporting and introduction interactions and their relevant dimensions of trust in this thesis but we did not explore them to any great depth. It would be interesting to further explore these interactions and their relevant problems.

The attacks in introduction interactions can be modeled in future work as follows. The attacker shows itself trustworthy in different dimensions in order to introduce several malicious agents to the trustworthy agents.

Although we consider introduction of the newcomer agents to agent community in our model (see Section 4.9), we did not explore this feature in our experiments. This should be undertaken as part of a comprehensive evaluation of all facets of the model proposed in this thesis.

We plan to design a con-resistant extension for Regret that can also be used for FIRE. With the advent of probabilistic trust models, future work will include the design of con-resistant probabilistic trust and reputation models.

The con-man attack can be extended to more complicated attacks in which the con-man observes the behavior of his/her opponents and changes his/her interaction patterns based on those observations (e.g., θ can be adaptive over the interactions of

a con-man). Design of these attacks might provide more comprehensive insight into characteristics of con-resistant trust models.

Furthermore, it would be interesting to observe the effect of con-man agents in a society of agents that employ social mechanisms (e.g., the use of witness information) which helps agents avoid encounters with con-man agents and thereby reduce exposure to confidence tricks.

We are interested in modeling spy-based attacks. In these attacks, the spy agent, by being trustworthy in all dimensions, and becoming a trusted member of society can gather information for malicious agents. Then, malicious agents will use the provided information for mounting their attacks.

7.3 Summary

Artificial societies and open distributed environments, especially e-commerce need trust and reputation models. Moreover, agents require trust and reputation concepts in order to identify communities of agents with which to interact reliably. Unfortunately, although trust and reputation models have a strong foundation on the assumption that agents may attempt to exploit each other, there is a little consideration regarding the possibility that agents may attempt to exploit the trust and reputation model itself. This observation has motivated this thesis to introduce a decentralized Exploitation-Resistant Trust (ERT) model which consists of environment and agent models as explained in Chapters 4 and 5 respectively. The agent model of ERT incorporates multiple sources of information to assess the trustworthiness of agents and is exploitation resistant in two dimensions of individual attacks (the con-man attack as explained in Section 5.3.1) and collusion attacks (the witness-based collusion attack as explained in Section 5.3.2). We empirically show utility and robustness of ERT in Chapter 6. We conclude that adaptivity and multi-dimensionality are the essential factors of exploitation resistant trust and reputation models. While this thesis has provided a useful starting point in the evaluation of the proposed model, it raises several new questions in areas related to model attacks, agent perceptions and learning mechanisms. We anticipate that the future work proposed here will uncover interesting new properties and requirements for distributed trust and reputation models

that will hopefully lead to practical implementation in future e-commerce systems.

Bibliography

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6007, Washington, DC, USA, 2000. IEEE Computer Society.

[2] G. A. Akerlof. The market for lemons: Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, 1970.

[3] Amazon. `http://www.amazon.com`, Feb 18 2009.

[4] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semant.*, 5(2):58–71, 2007.

[5] R. Axelrod. *The Evolution of Cooperation.* New York: Basic Books, 1984.

[6] R. Axelrod. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration.* Princeton University Press, Princeton, NJ, 1997.

[7] K. S. Barber and J. Kim. Soft security: Isolating unreliable agents from society. In *Trust, Reputation, and Security*, pages 224–233, 2002.

[8] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[9] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(3):7280–7287, May 2002.

[10] P. Braconnot Velloso, R. Pinaud Laufer, O.-C. Duarte, and G. Pujolle. A trust model robust to slander attacks in ad hoc networks. *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, pages 1–6, Aug. 2008.

[11] S. Brainov and T. Sandholm. Contracting with uncertain level of trust. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 15–21, New York, NY, USA, 1999. ACM.

[12] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998.

[13] J. Carbo, J. Molina, and J. Davila. Comparing predictions of sporas vs. a fuzzy reputation agent system. In *Proceedings of the International Conference on Fuzzy Sets and Fuzzy Systems*, pages 147–153, Interlaken, Switzerland, 2002.

[14] S. Casare and J. Sichman. Towards a functional ontology of reputation. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 505–511, New York, NY, USA, 2005. ACM.

[15] C. Castelfranchi, R. Conte, and M. Paolucci. Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation*, vol. 1, no. 3, 1998.

[16] C. Castelfranchi, R. Falcone, and G. Pezzulo. Trust in information sources as a source for trust: a fuzzy approach. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 89–96, New York, NY, USA, 2003. ACM.

[17] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, London, UK, 1996. Practical Application Company.

[18] R. Conte and M. Paolucci. *Reputation in Artificial Societies: Social Beliefs for Social Order (Multiagent Systems, Artificial Societies, and Simulated Organizations)*. Springer, October 2002.

[19] P. Dasgupta. *Trust: Making and Breaking Cooperative Relations*, chapter Trust as a Commodity, pages 49–72. Department of Sociology, University Oxford, 2000.

[20] C. Dellarocas. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. In *In ICIS*, pages 520–525, 2000.

[21] eBay. `http://www.ebay.com`, Feb 18 2009.

[22] R. Falcone and C. Castelfranchi. *Trust and Deception in Virtual Societies*, chapter Social Trust: A Cognitive Approach, pages 55–90. Kluwer Academic Publishers, 2001.

[23] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111, New York, NY, USA, 2004. ACM.

[24] I. Foster. The anatomy of the grid: enabling scalable virtual organizations. pages 6–7, 2001.

[25] K. K. Fullam and K. S. Barber. Dynamically learning sources of trust information: experience vs. reputation. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[26] K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 512–518, New York, NY, USA, 2005. ACM.

[27] D. Gambetta. Can we trust trust. In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[28] T. Grandison and M. Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3(4), September 2000.

[29] C.-W. Hang, Y. Wang, and M. P. Singh. An adaptive probabilistic trust model and its evaluation. In *AAMAS (3)*, pages 1485–1488, 2008.

[30] D. Houser and J. Wooders. Reputation in auctions: Theory, and evidence from ebay. *Journal of Economics & Management Strategy*, 15(2):353–369, 06 2006.

[31] T. Huynh, N. R. Jennings, and N. Shadbolt. Developing an integrated trust and reputation model for open multi-agent systems. In *7th International Workshop on Trust in Agent Societies*, pages 65–74, 2004.

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

[33] S. L. Jarvenpaa, N. Tractinsky, and M. Vitale. Consumer trust in an internet store. *Inf. Technol. and Management*, 1(1-2):45–71, 2000.

[34] N. R. Jennings. An agent-based approach for building complex software systems. *Commun. ACM*, 44(4):35–41, 2001.

[35] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[36] A. Josang and R. Ismail. The beta reputation system. In *In Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

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

[38] R. Jurca and B. Faltings. Towards incentive-compatible reputation management. In *In Proceedings of the AAMAS 2002 Workshop on Deception, Fraud and Trust in Agent Societies*, pages 92–100. ACM Press, 2002.

[39] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[40] R. Kerr and R. Cohen. Smart cheaters do prosper: Defeating trust and reputation systems. In *(To appear) AAMAS '09: Proceedings of the eighth international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, 2009. ACM.

[41] D. M. Kreps and R. Wilson. Reputation and imperfect information. *Journal of Economic Theory*, 27(2):253–279, August 1982.

[42] H. E. Kyburg, Jr. Bayesian and non-bayesian evidential updating. *Artif. Intell.*, 31(3):271–293, 1987.

[43] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction.* Morgan & Claypool, San Rafael, CA, 2008.

[44] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

[45] Z. Liu, S. S. Yau, D. Peng, and Y. Yin. A flexible trust model for distributed service infrastructures. In *ISORC '08: Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing*, pages 108–115, Washington, DC, USA, 2008. IEEE Computer Society.

[46] R. Marimon, J. P. Nicolini, and P. Teles. Competition and reputation. In *In Proceedings of the World Conference Econometric Society*, Seattle, 2000.

[47] S. Marsh. Formalising trust as a computational concept, 1994.

[48] S. A. Mcilraith, T. C. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16:46–53, 2001.

[49] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot. *Handbook of Applied Cryptography.* CRC Press, Inc., Boca Raton, FL, USA, 1996.

[50] T. M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[51] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7*, page 188, Washington, DC, USA, 2002. IEEE Computer Society.

[52] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 280–287, New York, NY, USA, 2002. ACM.

[53] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393:573–577, 1998.

[54] D. Olmedilla, O. F. Rana, B. Matthews, and W. Nejdl. Security and trust issues in semantic grids. In *Semantic Grid: The Convergence of Technologies*, volume 05271 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, July 2005.

[55] A. Oram, editor. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.

[56] J. E. Ormrod. *Human learning (4th ed.)*. Prentice Hall, 2003.

[57] W. Poundstone. *Prisoner's Dilemma: John Von Neumann, Game Theory and the Puzzle of the Bomb*. Doubleday, New York, NY, USA, 1992.

[58] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *Knowl. Eng. Rev.*, 19(1):1–25, 2004.

[59] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *The Economics of the Internet and E-Commerce*, 11:127–157, 2002.

[60] J. I. C. Rubiera, J. M. Molina, and J. D. Muro. Trust management through fuzzy reputation. *Int. J. Cooperative Inf. Syst.*, 12(1):135–155, 2003.

[61] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[62] J. Sabater and C. Sierra. Regret: A reputation model for gregarious societies. In *Fourth Workshop on Deception Fraud and Trust in Agent Societies*, pages 61–70, 2001.

[63] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.

[64] A. Salehi-Abari and T. White. Detecting and dealing with naive agents in trust-aware societies. In *Trust '09: Proceedings of the 12th International Workshop on Trust in Agent Societies*, 2009 (to appear).

[65] A. Salehi-Abari and T. White. Towards con-resistant trust models for distributed agent systems. In *IJCAI '09: Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, 2009 (to appear).

[66] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societies. *In Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*, 14(8):825–848, September 2000.

[67] H. Schmeck, T. Ungerer, and L. C. Wolf, editors. *Trends in Network and Pervasive Computing - ARCS 2002, International Conference on Architecture of Computing Systems, Karlsruhe, Germany, April 8-12, 2002, Proceedings*, volume 2299 of *Lecture Notes in Computer Science*. Springer, 2002.

[68] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In *AAMAS 02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 288–293. ACM Press, 2002.

[69] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, New York, 2009.

[70] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[71] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 997–1004, New York, NY, USA, 2005. ACM.

[72] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Journal of Autonomous Agents and Multi-Agent Systems*, 12:2006, 2006.

[73] G. Tesauro. Temporal difference learning of backgammon strategy. In *ML92: Proceedings of the ninth international workshop on Machine learning*, pages 451–457, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[74] T. Tran and R. Cohen. Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 828–835, Washington, DC, USA, 2004. IEEE Computer Society.

[75] A. Whitby, A. Josang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of 7th International Workshop on Trust in Agent Societies*, 2004.

[76] M. Woolridge and M. J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[77] B. Yu, M. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, pages 1–10, 30-31 Aug. 2004.

[78] B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *CIA '00: Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace*, pages 154–165, London, UK, 2000. Springer-Verlag.

[79] B. Yu and M. P. Singh. An evidential model of distributed reputation management. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301, New York, NY, USA, 2002. ACM.

[80] B. Yu and M. P. Singh. Detecting deception in reputation management. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 73–80, New York, NY, USA, 2003. ACM.

[81] G. Zacharia. Collaborative reputation mechanisms for online communities. *Masters thesis, Massachusetts Institute of Technology*, September 1999.

[82] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, (14):881–907, 2000.

[83] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms in electronic marketplaces. In *HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 8*, page 8026, Washington, DC, USA, 1999. IEEE Computer Society.