Doctoral Dissertation

A Belief Revision Mechanism with Trust Reasoning based on Extended Reciprocal Logic for Multi-Agent Systems

マルチエージェントシステムのための拡張相互論理に基づく信頼推論を用いた信念更新メカニズム

Sameera Basit
18DM005

Graduate School of Science and Engineering,
Saitama University

Supervisor: Professor Yuichi Goto

March 2023

Abstract

Many reciprocal relationships must concern two parties, e.g., parent-child relationship, relative relationship, friendship, cooperative relationship, complementary relationship, trade relationship, buying and selling relationship, and so on. A trust relationship is one of the most important reciprocal relationships. It is the basis for communication between agents (human to human, human to system, and system to system) and the basis for decision-making. These trust relationships play a crucial role in open and decentralized systems. Before communicating with an agent in such systems, it is hard to know whether they can be trusted. However, to establish trustworthiness and security in communication, it is essential to know whether the agent can be trusted or not.

Trust reasoning is an indispensable process for establishing trustworthy and secure communication under open and decentralized systems that include multi-agents. Thus, we should calculate the degree of trust of the target agent by using already-known facts, hypotheses, and observed data. Trust reasoning is a process to calculate the degree of trust or to decide which target can be regarded as trusted.

Various reasoning methods have been introduced in the literature to realize trust reasoning, e.g., probability reasoning, statistical reasoning, and logic-based reasoning. Logic-based reasoning method is qualitative, whereas probability and statistical reasoning approaches are quantitative. Logic-based reasoning methods can be applied to various target domains because the method is application-independent. Several logic systems have been proposed in previous works about logic-based trust reasoning. Those logic systems can deal with various trust properties, e.g., sincerity, validity, vigilance, credibility, cooperativity, and completeness. Such trust properties are essential for trust relationships. However, the logic systems are a conservative extension of classical mathematical logic, i.e., they are unsuitable logic systems underlying trust reasoning.

On the other hand, reciprocal logic was proposed and is an expectable candidate for a logic system underlying trust reasoning. Reciprocal logic is based on strong relevant logic. Strong relevant logic and its conservative extensions are suitable logic systems underlying reasoning. However, reciprocal logic cannot deal with the trust properties essential for trust relationships.

Thus, the right fundamental logic system underlying trust reasoning 1) should be based on strong relevant logic, and 2) can deal with trust properties, e.g., sincerity, completeness, validity, etc., essential for trust relationships. Such a logic system that satisfies the two conditions is demanded. However, there has not been such a logic system just now.

We proposed a right fundamental logic system that fulfills the above two conditions. The current reciprocal logic does not satisfy both conditions. We extended the current reciprocal logic, named extended reciprocal logic (ERL). For such purposes, we first surveyed and identified trust properties from the literature review essential for complex trust relationships. After that, we introduced them to our

proposed logic system for complex trust relationships. ERL is a logic system based on strong relevant logic, i.e., a suitable logic system for trust reasoning. It can specify, verify, and reason about various trust relationships. It is a logic system that provides us with the criteria of logical validity of reasoning as well as representation and specification language. The extension of extended reciprocal logic is two folds: 1) we introduced trust properties essential for trust relationships in multi-agent systems. Moreover, we provided logical formulas for the introduced trust properties, and 2) we regard messages from another agent as a proposition and individual constants.

Additionally, we proposed a belief revision mechanism with trust reasoning based on ERL for multi-agent systems. An agent in a multi-agent system has a set of beliefs (observed facts, previously given theories, and hypotheses). When an agent receives messages from other agents, it does belief revision, i.e., 1) it obtains new belief related to the messages; 2) it can deduce implicitly unknown beliefs from the obtained belief and own belief set; 3) if it has a contradiction in the belief set, then it solves the contradiction. In our mechanism, trust reasoning based on ERL is used for the deduction process in the belief revision. Our mechanism also gives a procedure to solve a contradiction in the belief revision.

Finally, we showed two case studies to show the generality of our proposed belief revision mechanism. The first case study is a scenario about public key infrastructure. The second case study is a scenario about a spy novel. Although the domains of those case studies are different, we can apply our proposed mechanism to both scenarios. Thus, we can conclude that our proposed mechanism is general-purpose.

The rest of this thesis is organized as follows. Chapter 1 presents the background, motivation, and purpose of this study. Chapter 2 shows the necessary conditions of the right fundamental logic systems, and their related basic notions, and notations. Chapter 3 describes our proposed extension, named extended reciprocal logic, and its usage. Chapter 4 proposes a belief revision mechanism with trust reasoning based on extended reciprocal logic for multi-agent systems and its application in various domains. Finally, discussion, and concluding remarks are given in Chapters 5, and 6 respectively.

Acknowledgments

First and foremost, I would like to praise and thank Allah, the almighty, who has granted countless blessing, knowledge, and opportunity, so that I have been finally able to accomplish my Ph.D journey.

Following this, it is my privilege to express sincere gratitude to my advisor Professor Yuichi Goto for his continuous support, patience, motivation, enthusiasm, and immense knowledge. His guidance greatly assisted me throughout the entire research and writing process. Having him as an advisor and mentor was the most valuable decision I ever made.

In addition to my supervisor, I would like to thank my co-supervisors: Prof. Yoshiura, Prof. Yamada, and Prof. Matsuta for their encouragement, insightful comments, and thoughtful questions.

My appreciation goes to the graduate school of science and engineering dept., financial support dept., as well as other departments of Saitama University. Also, I am sincerely honored to receive a scholarship from Nakamura Sekizen Foundation.

Also, I would also like to thank AISE lab members and my friends for their support. Last but not the least, This endeavor would not have been possible without the continuous support of my parents, parents-in-law, and especially my children. In the midst of the toughest times, their prayers kept me morally strong.

The final word of thanks goes to my beloved husband, Dr. Basit Sarfaraz. His love, understanding, encouragement, care, and existence are my keystone. I can honestly say that without him, I would not have gone through my Ph.D. journey. We wish each other an everlasting relationship of trust.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

There are many reciprocal relationships that must concern two parties, e.g., parent-child relationship, relative relationship, friendship, cooperative relationship, complementary relationship, trade relationship, buying and selling relationship, and so on [13]. A trust relationship is one of the most important reciprocal relationships and is the basis for communication between agents (human to human, human to system, and system to system), as well as the basis for decision-making.

Trust relationships play a key role in open and decentralized systems. Before communicating with an agent in such systems, it's hard to know whether they can be trusted. However, to establish trustworthiness and security in communication, it is essential to know whether the agent can be trusted or not.

Trust reasoning is an indispensable process for establishing trustworthy and secure communication under open and decentralized systems that include multi-agents. Thus, we should calculate the degree of trust of the target agent by using already-known facts, hypotheses, and observed data. Trust reasoning is a process to calculate the degree of trust or to decide which target can be regarded as trusted. Furthermore, the decision-making process of an agent with reasoning capability, i.e., an intelligent agent in a multi-agent system needs to include trust reasoning.

In order to realize trust reasoning various reasoning methods have been introduced in the literature, e.g, probability reasoning, statistical reasoning, and logic-based reasoning. Logic-based reasoning method is qualitative whereas probability and statistical reasoning approaches are quantitative. However, the logic-based method is useful and general. Currently, several logic systems have been proposed by the author. But, these logic systems are a conservative extension of classical mathematical logic, i.e., they are not suitable logic systems underlying trust reasoning. Whereas, their advantage is that it deals with trust properties which are essential for trusts relationship.

On the other hand, Reciprocal logic [13] is an expectable candidate for a logic system underlying trust reasoning. Because it is based on strong relevant logic rather than classical mathematical logic, i.e., it is a suitable logic system underlying reasoning. But, its disadvantage is that it cannot deal with trust properties which are essential for trusts relationship.

Thus, a right fundamental logic system underlying trust reasoning, 1) should not be a conservative extension of classical mathematical logic but rather based on strong relevant logic, and 2) can deal with trust properties that are essential for a trust relationship. Such a logic system that satisfies the two conditions is demanded. However, there is no such logic system existing until now.

In addition, belief revision must be included in the decision-making process of an intelligent agent in multi-agent systems. Every intelligent agent in a multi-agent system has a set of beliefs as observed facts, already given theories, and assumptions. These agents can deduce beliefs from their belief set using trust reasoning, and then it decides the next actions according to their current belief set.

Usually, the belief set of an agent is not always consistent, due to contradictions between previous and current assumptions, or observed facts. So, Belief revision is a process of solving a contradiction in a target belief set to keep the belief set consistent. Therefore, it is demanded to develop a mechanism in which an agent is capable of performing trust reasoning, and belief revision for its decision-making process.

Although a belief revision mechanism with trust reasoning is demanded to construct multi-agent systems, there is no such belief revision mechanism. On the other hand, a well-known belief revision mechanism is the so-called truth maintenance systems, belief revision systems, or reason maintenance systems [18]. Essentially, the concept of truth maintenance systems is independent of a specific logic system. However, there is no truth maintenance system based on a logic system underlying trust reasoning.

1.2 Purpose and Objectives

The purpose of this research is to construct a belief revision mechanism with trust reasoning based on a prepared logic system underlying trust reasoning. To begin, we prepared the right fundamental logic system underlying trust reasoning. In order to accomplish this, we surveyed and identified the trust properties that are essential for trust relationships, and introduced these trust properties as new trust axioms into the logic system. After that, we constructed a belief revision mechanism with trust reasoning based on the prepared logic system. Lastly, we discuss the application of the belief revision mechanism in different areas.

1.3 Structure of thesis

The rest of this thesis is organized as follows. Chapter 2 shows the necessary conditions of the right fundamental logic systems, and their related basic notions, and notations. Chapter 3 describes our proposed extension, named extended reciprocal logic, and its usage. Chapter 4 proposes a belief revision mechanism with trust reasoning based on extended reciprocal logic and its application in multiple domains. Finally, discussion, and concluding remarks are given in Chapters 5, and 6 respectively.

Chapter 2

Basic Notions and Notations

2.1 Trust Reasoning in Multi-Agent Systems

2.1.1 Multi-Agent systems

The notion of trust has been around for many decades in different disciplines in different disguises. There are numerous studies of trust in various fields. Many researchers have realized that trust has immense significance in multi-agent systems.

A multi-agent system refers to a group of agents in which the agents interact with each other as well as the environment in order to achieve goals. Architectures of multi-agent systems have been broadly classified into two types: centralized structures and decentralized structures. The centralized structure is via the control of agents by one control center, as in a relationship between a master and slave. The agents may be homogeneous and non-communicative in nature [17] whereas a decentralized structure requires both autonomous actions and coordinated interactions between its agents.

In the scope of this thesis, we consider the agents that are intelligent and able to function in a decentralized structure. Multi-agents in a decentralized structure communicate, interact, and exchange information autonomously. These intelligent agents possess reasoning capability for their decision-making process. Because decentralized systems are subject to various risks that require some assumption of trustworthiness on the part of the agents involved in their interactions. Agents may encounter other agents that may be untrustworthy, e.g., encounters with untrustworthy seller agents by a buyer [39]. So based on their reasoning capability agent need to decide on a suitable and trusted agent.

Every agent in a multi-agent system has a trust relationship among them. It occurs when an agent has a perspective regarding the trustworthiness of another agent. Through these trust relationships, it is able to ensure that the other agents with whom it communicates are trustworthy enough based upon whatever criteria are relevant to that communication.

2.1.2 Trust Relationship in Multi-Agent Systems

There are many reciprocal relationships that must concern two parties, e.g., parent-child relationship, relative relationship, friendship, cooperative relationship, complementary relationship, trade relationship, buying and selling relationship, and so on [13]. A trust relationship is one of the important reciprocal relationships in our society and cyberspace. Especially, a trust relationship is the basis of communication between human to human, human to system, and system to system, and the basis of decision-making of human and/or system.

These trust relationships are of fundamental importance when analyzing and designing multi-agent systems. In the literature, different authors have provided various definitions of trust relationships. Actually, there is no consensus in the literature on what a trust relationship is. A trust relationship exists when an entity has an opinion on the trustworthiness of another entity. Thus, trust relationships do not exist between strangers or an entity that has no knowledge about another' s existence [1]. [47] defined the whole syntax of trust relationship under a set of specified conditions, a set of trusters trust that a set of trustees have a set of specified properties (the set of trustees will/can perform a set of actions or have a set of attributes).

In order to have a clear understanding of trust relationships, we categorize trust relationships into two types: 1) Simple trust relationships and 2) Complex trust relationships. Simple trust relationships are relationships that deal with a trustor, and a trustee only, whereas complex trust relationships deal with a trustor, a trustee, and, a trust property.

In multi-agent systems, trust relationships between agents change themselves over space and time. So, it is difficult to know whether a system or an agent that requires to connect to our system can be trusted or not before communicating with it. So agents need to reason about these trust relationships.

2.1.3 Trust Reasoning

Every intelligent agent task contains a set of essential activities. We consider intelligent agents whose activities can be described in general as follows: An intelligent agent receives messages from other agents by observing its external environment or own internal status. Using its reasoning capability, it should perform trust reasoning. Trust reasoning is a process to draw propositions from already known propositions using the degree of trust of an agent or a received message. Any agent in multi-agent systems can extend its belief set by receiving messages from other agents and observing its external environment or own internal status. Especially, an intelligent agent deduces implicitly included propositions from its belief set. After that, the agent decides the next actions according to its current belief set. An intelligent agent in an open and decentralized system should be able to change the way it handles messages from other agents depending on the degree of trust of the agents because not all agents in the system can be trusted. Thus, an intelligent agent should be able to do trust reasoning for its decision-making.

Trust reasoning is presented as a starting point for work involving trust relationships, especially in multi-agent systems. It is argued that an agent who can reason about trust relationships will be more capable of making reliable, and informed decisions more efficiently. Due to the changing nature of trust relationships, we need to calculate these changing complex trust relationships between agents in order to decide on a suitable and trusted agent.

Trust reasoning is one of the calculation methods. It is a process to calculate the degree of trust or to decide which target can be regarded as trust one [9]. Various reasoning methods have been used to reason about the target. Examples include probability reasoning, statistical reasoning, reasoning using possibility theories, fuzzy set logic, argumentation, and logic-based reasoning. [45] focussed on probabilistic reasoning in multi-agents in a complex environment. Author [28] also used a probabilistic reasoning method for reasoning about the mental attitudes of agents. Reasoning based on Fuzzy logic deals with all interactions between the agents that have to operate in such uncertain and constantly changing environments using fuzzy reasoning. The approach takes different trust sources of agents in domains and calculates the overall trust value [27]. In [40] author has performed statistical reasoning within decentralized multi-agent systems. The approach has been applied to multi-agents who share both private and shared information. Various other authors including [1] provided novel algorithms for trust reasoning, these include trustworthiness measures based on generalized experiences and recognition of the different phases of a trust relationship.

However, this thesis focuses on logic based reasoning method. Logic-based techniques are not only the oldest of any reasoning method, but may represent the most straightforward way to human understanding among all other methods. These approaches are at the core of many successful agent-based models and technologies. Usually, agents in multi-agent systems straightforwardly exploit logic-based models and technologies for rational process, knowledge representation, expressive communication, and effective coordination. As [3] provided a flexible declarative language for the specification and implementation of dynamic knowledge construction in a society of agents. Modal logic also provides formalism to specify, verify, and reason in multi-agent systems. It provides criteria for validating and deciding about the behaviors of multi-agents with incomplete knowledge, as observed by [15, 37, 38]. But all these reasoning methods are quantitative in nature and are based on classical mathematical logic and its various conservative extensions, whereas the logic-based reasoning method is qualitative. A logic-based method is useful and general because it is enough abstract representations. Also, these approaches provide powerful mechanisms that can be effectively used for trust reasoning. Such approaches yield formal semantics to reason about trust properties in various applications such as security protocols, information sources, and e-markets [19].

Altogether, this is why this paper focuses on logic-based approaches in multi-agent systems as they are to be counted among the most promising techniques for trust reasoning.

2.1.4 Target of Trust Reasoning

Trust reasoning is an indispensable process for establishing trustworthy and secure communication under open and decentralized systems that include multi-agents. In open and decentralized systems, although it is difficult to know whether an agent that is required to communicate with us can be trusted or not before communication with it, we want to know whether the agent is trusted or not to establish trustworthy and secure communication, e.g., public key infrastructure (PKI). Thus, we should calculate the degree of trust of the target agent by using already-known facts, hypotheses, and observed data. Trust reasoning is a process to calculate the degree of trust of the target agents and messages that come from other agents.

Authors have used trust reasoning in various fields such as in [33] author discussed that trust relationships, play an important role in public key infrastructures, and these trust relationships need to be formalized for providing a reliable modeling methodology to support secure digital communications. Reasoning about security in Information and Communication Technology, short as ICT, systems [23]. Trust reasoning is required in open distributed information systems where no centralized party can verify an agent's trustworthiness [47].

It is possible to detect betrayal in [22] by performing trust reasoning in order to determine whether another opponent is betraying the player or not. In [2], propose for Optimized Link State Routing Protocol, short as OLSR, the integration of trust reasoning into each node, so as to allow a self-organized trust-based control to help nodes to detect misbehavior attacks. In [44] trust reasoning is used to choose enhanced service composition.

Hence, from the literature reviewed above, trust reasoning is typically applied to multi-agent systems where there is an open, decentralized environment in order to secure communication between agents.

2.2 Trust Properties

2.2.1 Overview of Trust Properties for Logic-based Trust Reasoning

Trust relationships have several aspects so we introduce trust properties. A trust relationship indicates that one agent believes that another agent satisfies that property, whereas trust property defines a criterion that should be fulfilled by the agent in a trust relationship. Trust relationships are more tractable with the aid of trust properties.

These trust properties play a very important role when dealing in the domain of multi-agent systems. In multi-agent systems, usually agents have trust relationships with other agents based on the basis of these trust properties, e.g., An agent trust another agent in its sincerity, whereas an agent is sincere if it communicates information that it believes [15].

Basically, trust is established in the context between two agents as trustors, and a trustee. Trustee provides trustworthy data to make a trustor trust in a trustee. For example, home appliance devices (trustee) provide energy-related data for users (trustor) to control these devices in the use case of home energy management [4]. Trust is affected by several subjective such as social status and physical properties, and objective factors such as competence and reputation [31,7]. Trust refers to a trustor's belief regarding the property of the trustee. In the field of trust, numerous amount of work shows the interest to study, and identify various trust properties for trust relationships. However, many trust properties has been identified in literature, as trust in reliability, honesty, credibility [15, 30, 25]. Most authors focused on only one dimension such as trust in the reliability [14, 21], trust in the sincerity [31] and other researchers dealt with trust and cooperation [30]. Demolombe [15] provided a formal definition for 6 trust properties based on modal logic will be discussed in detail in the next section. [7] presented suggestions to evaluate trust with regard to competence, benevolence, integrity, and predictability and also targeted trust in different context and technology areas but its formal representation is not presented.

Literature analysis shows that the authors focus on the importance of trust properties in multi-agent systems as a common objective because trust has been gaining increasing interest in multi-agent systems. We surveyed the state of the art of trust properties in multi-agent systems, and based on the literature and section 2.2.2 we considered these trust properties for the proposition of logic system, which is one of the targets of our research work.

2.2.2 Formal Notion of Trust Properties by Demolombe

Demolombe [15] proposes a framework to reason about trust using several properties. He analyzes the trust that can be associated with agents in information sources. In order to deduce the trustworthiness of an agent, he defines trust in terms of the trust properties of an agent such as sincerity, completeness, validity, etc., and develops axioms that help in the deduction. These properties and axioms enable the use of trust reasoning when there is a message exchange among agents.

Demolombe [15] defined several trust properties. His definitions are as follows.

- *Sincerity*: An agent α trusts in the sincerity of an agent β if β informs α about a proposition p then β believes p.
- *Validity*: An agent α trusts in the validity of an agent β if β informs α about a proposition p then p is the case.
- *Completeness*: An agent α trusts in the completeness of an agent β if p is the case then β informs α about p.
- *Cooperativity*: An agent α trusts in the cooperativity of an agent β if β believes p then β informs α about p.
- *Credibility*: An agent α trust in the credibility of an agent β if β believes p then p is the case.
- *Vigilance*: An agent α trust in the vigilance of an agent β if p is the case then β believes p.

Demolombe also provided a formal definition of the above properties. But, his formalization is based on classical mathematical logic and its conservative extensions. Our research introduces such trust properties in our proposed logic system for trust reasoning. Moreover, Demolombe regarded messages from other agents in an information system as the beliefs of the agents and represented them as propositions (logical formulas) from the viewpoint of predicate logic. From the viewpoint of expressive power, Demolombe's approach is better.

2.2.3 Necessity of the Trust Properties for Trust Reasoning in Multi-Agent Systems

There has been discussion in many forums regarding the need for trustworthy systems. A multi-agent system is composed of multiple agents, which interact in dynamic and uncertain environments in order to achieve their goal [43]. Agents may be heterogeneous, which means that they may have different preferences, behaviors, and the ability of agents to communicate and interact with one another is one of their essential properties [19].

These properties are essential to show an agent's capability and ability to interact and communicate with other agents, and to act consistently with its goals. Studies show that trust properties help to reason in various applications such as security protocols, information sources, and recommendation systems to check whether the desired properties of the system hold or not [8], and to detect undesired behaviors with regard to particular properties.

One approach towards enabling trust properties in a trust relationship involves the use of the agent's behavior and preferences in a variety of critical domains, e.g., commercial, industrial, government, and healthcare systems. It enables the agent in a domain to make a decision about whether or not the other agent, or system is trusted or not. Also, there is a continued rise in the complexity of computer systems as more functionality is required [36]. If such systems are to be used for critical operations, or operate on critical data, it is important that agents are considered to ensure that they conform to trust properties. Especially, the power of autonomous agents lies in their ability to deal with unpredictability, and their dynamism may lead to unexpected behavior. This brings up the question: how can we guarantee that the agent will behave as expected in a domain given that its state continues to change. Here trust properties play a vital role in order to determine the agents behaviour.

In addition, agents can act as attackers and are opportunistic. Despite continued efforts in trustworthy and security system design, the number and level of sophistication of attacks continue to increase. For such purposes, the trust properties help determine and reason about the behavior of a malicious agent.

2.3 Reciprocal Logic

2.3.1 Strong Relevant Logic as an Expectable Candidate for Trust Reasoning

Strong relevant logics were proposed in order to find a satisfactory logic to underlie the relevant reasoning. These logics require that the premises of an argument represented by an entailment include no unnecessary and needless conjuncts and the conclusion of that argument includes no unnecessary and needless disjuncts, and reject those conjunction-implicational paradoxes and disjunction-implicational paradoxes [12].

Relevance principle in strong relevant logic excludes those implicational paradoxes from logical axioms or theorems of relevant logics. Reasoning based on the strong relevant logics is ampliative but not circular and/or tautological. Strong relevant logics reject the principle of Explosion, they can certainly underlie paraconsistent reasoning. They guarantee the relevance between the premises of a valid argument and its conclusion and the validity of its conclusion in a sense of weak relevance. Thus, no existing logic can satisfy all essential requirements for the fundamental logic. So, a new family of conservative extensions of relevant logic, named Reciprocal logic was proposed. Reciprocal logics were obtained by introducing predicates and related axioms about reciprocal relationships into strong relevant logics.

2.3.2 Reciprocal Logic and its Limitation

Reciprocal logic was proposed by [13] as a logic system underlying reasoning for a reciprocal relationship. Classical mathematical logic and its various conservative extensions are not suitable for logic systems underlying reasoning because they have paradoxes of implication [5, 6]. Strong relevant logic has rejected those paradoxes of implication and is considered the universal basis of various applied logic for knowledge representation and reasoning [12]. Thus, strong relevant logic and its conservative extensions are candidates for logic systems underlying reasoning. Reciprocal logic is one of the conservative extensions of strong relevant logic to deal with various reciprocal relationships. Reciprocal logic provides predicates representing trust relationships between an agent and another agent, and between an agent and an organization, defined predicates based on the primitive predicates, and several axioms that include the predicates [13].

Predicates

Let pe_1, pe_2, and pe_3 be individual variables representing agents, and let o_1 and o_2 be individual variables representing organizations. The primitive predicates are as follows [13]:

- $TR(pe_1, pe_2)$: pe_1 trusts pe_2.
- $B(pe_1, o_1)$: agent pe_1 belongs to organization o_1

Defined predicates based on the above primitive predicate are as follows [13]:

- $NTR(pe_1, pe_2) =_{df} \neg(TR(pe_1, pe_2))$ ($NTR(pe_1, pe_2)$ means pe_1 does not trust pe_2.)
- $TREO(pe_1, pe_2) =_{df} TR(pe_1, pe_2) \wedge (TR(pe_2, pe_1))$ ($TREO(pe_1, pe_2)$ means pe_1 and pe_2 trust each other.)
- $ITR(pe_1, pe_2, pe_3) =_{df} \neg(TR(pe_1, pe_2) \wedge TR(pe_1, pe_3))$ ($ITR(pe_1, pe_2, pe_3)$ means pe_1 doe snot trust both pe_2 and pe_3 (Incompatibility))
- $XTR(pe_1, pe_2, pe_3) =_{df} (TR(pe_1, pe_2) \vee TR(pe_1, pe_3)) \wedge (NTR(pe_1, pe_2) \vee NTR(pe_1, pe_3))$ ($XTR(pe_1, pe_2, pe_3)$ means pe_1 trusts either pe_2 or pe_3 but not both (exclusive disjunction).)
- $JTR(pe_1, pe_2, pe_3) =_{df} \neg(TR(pe_1, pe_2) \vee TR(pe_1, pe_3))$ ($JTR(pe_1, pe_2, pe_3)$ means pe_1 trusts neither pe_2 nor pe_3 (joint denial).)
- $TTR(pe_1, pe_2, pe_3) =_{df} (TR(pe_1, pe_2) \wedge TR(pe_1, pe_3)) \Rightarrow TR(pe_1, pe_3)$ ($TTR(pe_1, pe_2, pe_3)$ means pe_1 trusts pe_3 if pe_1 trusts pe_2 and pe_2 trusts pe_3.)
- $CTR(pe_1, pe_2, pe_3) =_{df} (TR(pe_1, pe_3) \Rightarrow (TR(pe_2, pe_3))$ ($CTR(pe_1, pe_2, pe_3)$ means pe_2 trusts pe_3 if pe_1 trusts pe_3.)
- $NCTR(pe_1, pe_2, pe_3) =_{df} (\neg TR(pe_1, pe_3) \Rightarrow (TR(pe_2, pe_3))$ ($NCTR(pe_1, pe_2, pe_3)$ means pe_2 trusts pe_3 if pe_1 does not trusts pe_3)
- $CNTR(pe_1, pe_2, pe_3) =_{df} \neg(TR(pe_1, pe_3) \Rightarrow \neg(TR(pe_2, pe_3))$ ($CNTR(pe_1, pe_2, pe_3)$ means pe_2 does not trusts pe_3 if pe_1 does not trusts pe_3)
- $TRpo(pe_1, o_1) =_{df} \forall pe_2(B(pe_2, o_1) \wedge (TR(pe_1, pe_2))$ ($TRpo(pe_1, o_1)$ means pe_1 trusts o_1.)
- $NTRpo(pe_1, o_1) =_{df} \forall pe_2(B(pe_2, o_1) \wedge (NTR(pe_1, pe_2))$ ($NTRpo(pe_1, o_1)$ means pe_1 does not trusts o_1.)
- $TRop(o_1, pe_1) =_{df} \forall pe_2(B(pe_2, o_1) \wedge (TR(pe_2, pe_1))$ ($TRop(o_1, pe_1)$ means o_1 trusts pe_1.)
- $NTRop(o_1, pe_1) =_{df} \forall pe_2(B(pe_2, o_1) \wedge (NTR(pe_2, pe_1))$ ($NTRop(o_1, pe_1)$ means o_1 does not trusts pe_1.)
- $TRoo(o_1, o_2) =_{df} \forall pe_1 \forall pe_2(B(pe_1, o_1) \wedge (B(pe_2, o_2)) \wedge (TR(pe_1, pe_2)$ ($TRoo(o_1, o_2)$ means o_1 trusts o_2.)
- $NTRoo(o_1, o_2) =_{df} \forall pe_1 \forall pe_2(B(pe_1, o_1) \wedge (B(pe_2, o_2)) \wedge (NTR(pe_1, pe_2)$ ($NTRoo(o_1, o_2)$ means o_1 doe snot trusts o_2.)

Through the above definitions of predicates, we can consider that reciprocal logic focuses on only the trust relationships between an agent and other agents, and between an agent and an organization.

Axioms

Axioms of the reciprocal logic are as follows:

TR1: $\neg(\forall pe_1 \forall pe_2 (TR(pe_1, pe_2) \Rightarrow TR(pe_2, pe_1)))$

TR2: $\neg(\forall pe_1 \forall o_1 (TRpo(pe_1, o_1) \Rightarrow TRop(o_1, pe_1)))$

TR3: $\neg(\forall o_1 \forall pe_1 (TRop(o_1, pe_1) \Rightarrow TRpo(pe_1, o_1)))$

TR4: $\neg(\forall o_1 \forall o_2 (TRoo(o_1, o_2) \Rightarrow TRoo(o_2, o_1)))$

TR5: $\neg(\forall pe_1 \forall pe_2 \forall pe_2 (TR(pe_1, pe_2) \wedge TR(pe_2, pe_3) \Rightarrow TR(pe_1, pe_3)))$

TR6: $\neg(\forall pe_1 \forall pe_2 \forall o_1 (TRpo(pe_1, o_1) \wedge TRop(o_1, pe_2) \Rightarrow TR(pe_1, pe_2)))$

TR7: $\neg(\forall pe_1 \forall pe_2 \forall o_1 (TRop(o_1, pe_1) \wedge TR(pe_1, pe_2) \Rightarrow TRop(o_1, pe_2)))$

TR8: $\neg(\forall o_1 \forall o_2 \forall o_3 (TRoo(o_1, o_2) \wedge TRoo(o_2, pe_3) \Rightarrow TR(o_1, o_3)))$

$TrTcQ =_{df} TcQ + \{\mathrm{TR1}, \ldots, \mathrm{TR8}\}$, $TrEcQ =_{df} EcQ + \{\mathrm{TR1}, \ldots, \mathrm{TR8}\}$, and $TrRcQ =_{df} RcQ + \{\mathrm{TR1}, \ldots, \mathrm{TR8}\}$ are the minimal logic systems of reciprocal logic where TcQ, EcQ, and RcQ are logic systems of the first order predicate strong relevant logics.

Limitation of Reciprocal Logic

Current reciprocal logic provides two primitive predicates: $TR(p1, p2)$ means "person p1 trusts person p2", and $B(p, o)$ means "person p belongs to organization o". Several complex predicates are defined by using primitive predicates, and several axioms as discussed in Section 2.3.2.

The first limitation of reciprocal logic is that these primitive, and complex predicates can only represent and deal with simple trust relationships, and does not provide enough predicates to describe complex trust relations, i.e., a trust relationship comprises of a trustor, a trustee, and a trust property.

Reciprocal logic only provides predicates to deal with trust relationships between agent to agent, agent to organization, and organization to organization. These primitive predicates do not take into account the message received from another agent.

In the current scope of research, we consider that an agent can receive two kinds of messages from another agent. Message as an : Individual constant, and Proposition. An individual constant represents an entity in the world that has a name, e.g., the name Alice is expressed by the individual constant a, whereas a proposition is a statement that can be either true or false, e.g., A sentence like Alice is a man is expressed as $M(a)$.

Besides the 1st limitation, the current reciprocal logic 2nd limitation is that it cannot deal with the messages received from another agent as an individual constant and proposition as well.

So, We need to prepare a logic system that can deal with the above limitations. This thesis aims to prepare, and extend current Reciprocal logic with trust properties, and trust axioms to deal with Reciprocal logic [13] limitations, and we named it Extended Reciprocal logic, short as ERL. It will be discussed in detail in Section 3.

2.4 Conditions of Logic System for trust reasoning

Currently, several logic systems [15, 32, 29, 35], and other which has been discussed in Section 2.2.1 have been proposed. The advantage of these logic systems is that they deal with trust properties which are essential for complex trusts relationship, but a disadvantage of these logic systems is that they are a conservative extension of classical mathematical logic, i.e., they are not suitable logic systems underlying reasoning.

On the other hand, a logic system called Reciprocal logic [13] is proposed as discussed in Section 2.3.2. Its advantage is that it is based on strong relevant logic rather than classical mathematical logic, i.e., it is a suitable logic system underlying reasoning, but its disadvantage is that current reciprocal logic does not cover such trust properties that are essential to deal with complex trust relationships.

So, keeping the advantages, and disadvantages of the logic system in view, we concluded that the right fundamental logic system should fulfill two necessary conditions: 1) it should not be a conservative extension of classical mathematical logic rather it should be based on strong relevant logic, and 2) it should deal with complex trust relationships including such trust properties.

Currently, there is no such right fundamental logic system underlying trust reasoning exists. So, this thesis focus on proposing a right fundamental logic underlying trust reasoning which fulfills above both conditions.

2.5 Chapter Summary

The purpose of this chapter is to summarize that multi-agent system, such as open, and decentralized systems, require trust reasoning since trust is an increasingly important factor as technology advances. At first, we defined what kind of multi-agent systems needs trust reasoning, and why trust reasoning is important. Then, we surveyed, and identified trust properties from the literature review that are essential for complex trust relationships, and make trust reasoning more tractable. Following this, we define conditions that are necessary for the right fundamental logic system underlying trust reasoning. Current reciprocal logic [13] doesn't fulfill both conditions, so we introduced a logic system that satisfies both conditions.

Chapter 3

Extended Reciprocal Logic and its Usage

3.1 Overview and Target of the Proposed Approach

Logic-based reasoning method could be seen as a good representation language for static knowledge [3], but given the dynamic environment such as open and decentralized systems, or multi-agent systems, we need to consider how knowledge from different sources can be represented. Such environments consist of trust relationships that change over time and space between agents. Trust properties are associated with such trust relationships, we consider them complex trust relationships as discussed in section 2.1.1. Also, these trust relationships among agents require reasoning mechanisms as mentioned in Section 2.1.2.

The main motivation behind the proposition of extended reciprocal logic is to represent the evolution of complex trust relationships with associated trust properties of each agent as well as the evolution of complex relationships of trust in space and time. So, representing complex trust relationships between agents, and providing a reasoning mechanism based on represented knowledge is regarded as one of the key targets of extended reciprocal logic.

3.2 Extended Reciprocal Logic

Extended reciprocal logic, ERL for short, is a logic system based on strong relevant logic, i.e., it is a suitable logic system for trust reasoning. Extended reciprocal logic can specify, verify, and reason about various trust relationships. It is a logic system that provides us with the criteria of logical validity of the reasoning as well as representation, and specification language. ERL satisfies both conditions of reciprocal logic as mentioned in Section 2.4, which makes it a right fundamental logic system underlying trust reasoning.

The extension of extended reciprocal logic is two folds: 1) we introduced trust properties essential for trust relationships in multi-agent systems. Moreover, we provided logical formulas for the introduced trust properties, and 2) we regard messages that come from another agent as a proposition as well as individual constants. The usage of extension of extended reciprocal logic will be shown in Section 3.3.

The first point of extension of extended reciprocal logic is achieved by carefully surveying, and identifying trust properties that are essential for trust relationships [9] as mentioned in Section 2.2.

The second point of extension of extended reciprocal logic is achieved by introducing modal operators into extended reciprocal logic. In our extension [9] we regarded messages that come from agents as countable objects (individual constants). Usually, in a multi-agent system, there is a message exchange among agents. We regard this message exchange between agents as the belief of agents as two types of individual constants, and propositions, and represented them as logical formulas from the viewpoint of predicate logic. From the viewpoint of applications of trust reasoning, we should regard the messages from other agents as propositions like Demolombe's logic system [15]. So, with respect to expressive power, Demolombe's approach is better. We adopted his approach in our extension to representing messages from other agents as a proposition in extended reciprocal logic.

To achieve the above two aspects of extension, At first, we proposed two modal operators to enable the expansion of reciprocal logic.

3.2.1 Modal Operators

These modal operators $Bel_i(A)$ and $Inf_{i,j}(A)$ are introduced into reciprocal logic from Demolombe's logic system to represent the trust relationship between agents and message that comes from other agents.

$Bel_i(A)$: an agent i believes that a proposition A is true.

$Inf_{i,j}(A)$: an agent i has informed an agent j about A.

3.2.2 Predicates

Second, we add a predicate "$TR(pe_1, pe_2, PROP)$" where pe_1 and pe_2 are agents, and $PROP$ is an individual constant that represents trust properties: sincerity, validity, completeness, cooperativity, credibility, and vigilance into reciprocal logic. For example, "$TR(pe_1, pe_2, sincerity)$" means "$pe_1$ trusts pe_2 in sincerity", "$TR(pe_1, pe_2, credibility)$" means "$pe_1$ trusts pe_2 in credibility", "$TR(pe_1, pe_2, completeness)$" means "$pe_1$ trusts pe_2 in completeness", and in the same way, we can define a predicate for other trust properties as well. Note that "$TR(pe_1, pe_2, all)$" means "pe_1 trusts pe_2 in all trust properties", i.e., "$TR(pe_1, pe_2)$" in reciprocal logic is as same as "$TR(pe_1, pe_2, all)$" in our new extension.

3.2.3 Axioms

Finally, we add new axioms in extended reciprocal logic using our modal operators, and the newly introduced predicate.

ERcL1: $\forall i \forall j(TR(i, j, sincerity) \Rightarrow (Inf_{j,i}(A) \Rightarrow Bel_j(A)))$

ERcL2: $\forall i \forall j(TR(i, j, validity) \Rightarrow (Inf_{j,i}(A) \Rightarrow A))$

ERcL3: $\forall i \forall j(TR(i, j, vigilance) \Rightarrow (A \Rightarrow Bel_j(A)))$

ERcL4: $\forall i \forall j(TR(i, j, credibility) \Rightarrow (Bel_j(A) \Rightarrow A))$

ERcL5: $\forall i \forall j(TR(i, j, cooperativity) \Rightarrow (Bel_j(A) \Rightarrow Inf_{j,i}(A)))$

ERcL6: $\forall i \forall j(TR(i, j, completeness) \Rightarrow (A \Rightarrow Inf_{j,i}(A)))$

BEL: $\forall i(Bel_i(A \Rightarrow B) \Rightarrow (Bel_i(A) \Rightarrow Bel_i(B)))$

3.2.4 Inference Rules

In the next section, we used two inference rules modus ponen ⇒E, and adjunction ∧I from reciprocal logic [13], and one inference rule necessitation $Bel - Nec$ from [15] in order to enable expansion of reciprocal logic for trust reasoning. The inference rules are as follows.

⇒E: "from A and $A \Rightarrow B$ to infer B" (Modus Ponens)

∧I: "from A and B infer $A \wedge B$" (Adjunction)

Bel-Nec: "if A is a logical formula, then so is $Bel_i(A)$" (Necessitation)

3.3 Usage of Extended Reciprocal Logic

We summarize our new extension of reciprocal logic. Let RcL be all axioms of reciprocal logic. Our new extension is $RcL \cup \{\text{ERcL1}, \ldots, \text{ERcL6}, \text{BEL}\}$.

The generality of extended reciprocal logic provides knowledge representation about complex trust relationships, and reasoning mechanisms from various domains. In order to show its generality few examples followed by a running example will be presented.

3.3.1 Representation using Reciprocal Logic

In order to demonstrate representation, we choose an example from the web-based trust model, which comprises a simple trust relationship between two agents' certificate authority ca and web w [46]. This simple relationship can be represented using by using extended reciprocal logic predicates as $TR(w, ca)$. If we consider the trust relationship between two agents' certificate authority ca and web w as a complex trust relationship, i.e., w trusts ca in its validity then this relationship can be represented using by using extended reciprocal logic predicates as $TR(w, ca, validity)$.

To illustrate the examples on how to represent messages from other agents using extended reciprocal logic, we examine a complex trust relationship between two agents' certificate authority ca and web w where w trusts ca in its validity, and an w received a message as an individual constant from ca. This will be represented as $Inf_{w,ca}(ct)$. In the same way a complex trust relationship between two agents' certificate authority ca and web w where w trusts ca in its validity, and an w received a message as a proposition "*certificate ct is valid*" from ca. This will be represented as $Inf_{w,ca}(isValid(ct))$. We regard such syntax, i.e., $Inf_{w,ca}(isValid(ct))$ as logical formulas.

Further, we will use the example from [10] as a running example to demonstrate our representation using extended reciprocal logic in a more detailed and broader context.

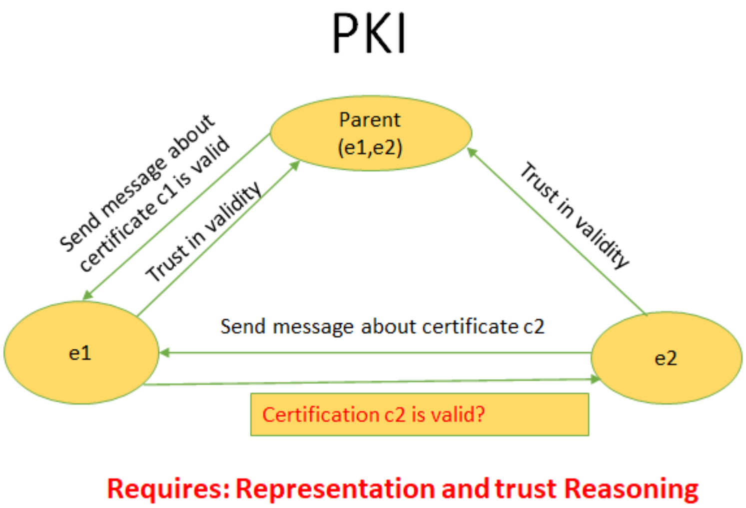

Figure 3.1: Public Key Infrastructure Scenario

Running Example

We present a simple scenario in public key infrastructure, short as PKI, inspired from [33]. We have formalized the scenario and applied the trust reasoning process based on extended reciprocal logic in PKI.

Suppose that a certificate c_2 is signed by the subject of a certificate c_1 with the private key corresponding to the public key of c_1. Agent e_1 trusts the certificate c_1 because c_1 is informed by its parent agent. In PKI, we consider that every agent trusts its parent agent in its validity, i.e., $\forall e(TR(e, parent(e), validity))$. Moreover, agent e_2 informs agent e_1 about certificate c_2. Agent e_1 doesn't trust the certificate but wishes to use certificate c_2. We need to know whether the certificate c_2 informed by agent e_2 is valid or not.

Formulization

To formalize the above scenario, we defined following constants, functions, and predicates.

- Individual variables:
  - entities: e, p, j
  - certification: c, c'
- Individual constants:
  - entities: e_1, p_1, j_1
  - certifications: c_1, c_2
  - *today*: date of today.
- Functions:
  - $I(c)$: Issuer of certification c.
  - $S(c)$: Subject of certification c.
  - $PK(c)$: Public key of c.
  - $SK(c)$: Share key of c.
  - $DS(c)$: Start date of c.
  - $DE(c)$: End date of c.
  - $Sig(c)$: Signature of c.
  - $parent(e)$: The parent of entity e.
- Predicates:
  - $inCRL(c)$: c is in certification revocation list.
  - $isValid(x)$: x is valid.
  - $isSigned(x, k)$: x is message signed by key k.
  - $x = y$: x is equal to y.
  - $x \leq y$: x is equal to or less than y.
  - $x < y$: x is less than y.

In PKI, we can assume following theories.

PKI1: $\forall e(TR(e, parent(e), validity))$
(Any entity trusts its parent entity in validity.)

PKI3: $\forall c(\exists c'((Inf_{parent(e),e}(isValid(c')))\wedge(I(c)=S(c'))\wedge(isSigned(c,PK(c')))) \Rightarrow isValid(Sig(c)))$

PKI4: $\forall c((isValid(Sig(c))\wedge(DS(c)\leq \mathit{today})\wedge(\mathit{today} < DE(c))\wedge\neg inCRL(c)) \Rightarrow isValid(c))$
(PKI3 and PKI4 allows to verify the signature and certificate itself on the basis of another certificate whose validity has been proven.)

From scenario, we can assume following logical formulas.

P1: $I(c_2)=S(c_1)$
(This observed facts are used as a premises in our reasoning process and it is true in this scenario only.)

P2: $isSigned(c_2, PK(c_1))$
(A certificate c_2 is signed by the subject of certificate c_1 with the private key corresponding to the public key of c_1.)

P3: $Inf_{parent(e_1),e_1}(isValid(c_1))$
(The parent entity of e_1 has informed e_1 about "certificate c_1 is valid".)

P4: $Inf_{e_2,e_1}(isValid(c_2))$
(The parent entity of e_1 has informed e_1 about "certificate c_1 is valid".)

P4: $DS(c_2)\leq \mathit{today}$ (observed facts subject to PKI)

P5: $\mathit{today} < DS(c_2)$ (observed facts subject to PKI)

P6: $\neg inCRL(c_2)$ (observed facts subject to PKI)

Trust reasoning process

According to the above formalization, we can reason out a conclusion "$Bel_{e_1}(isValid(c_2))$ " as follows.

1. $Inf_{parent(e_1),e_1}(isValid(c_1)) \wedge (I(c_2) = S(c_1)) \wedge isSigned(c_2, PK(c_1))$ [from P1, P2, and P3 with ∧I]
2. $Bel_{e_1}(Inf_{parent(e_1),e_1}(isValid(c_1)) \wedge (I(c_2) = S(c_1)) \wedge isSigned(c_2, PK(c_1)))$ [from 1 with Bel-Nec]
3. $(Inf_{parent(e_1),e_1}(isValid(c_1)) \wedge (I(c_2) = S(c_1)) \wedge (isSigned(c_2, PK(c_1)))) \Rightarrow isValid(Sig(c_2))$ [Replaced c with c_2 and c' with c_1 in PKI3]
4. $Bel_{e_1}((Inf_{parent(e_1),e_1}(isValid(c_1))\wedge(I(c_2) = S(c_1))\wedge(isSigned(c_2, PK(c_1)))) \Rightarrow isValid(Sig(c_2)))$ [from 4 and 2 with ⇒E]
5. $Bel_{e_1}((Inf_{parent(e_1),e_1}(isValid(c_1))\wedge(I(c_2) = S(c_1))\wedge(isSigned(c_2, PK(c_1))))) \Rightarrow Bel_{e_1}(isValid(Sig(c_2)))$ [from BEL and 5 with ⇒E]
6. $Bel_{e_1}(isValid(Sig(c_2)))$ [from 3 and 6 with ⇒E]

7. $Bel_{e_1}(DS(c_2) \leq today)$, $Bel_{e_1}(today < DS(c_2))$, $Bel_{e_1}(\neg inCRL(c_2))$ [from each of P4 to P6, and 2 with ⇒E]

8. $Bel_{e_1}(isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2))$ [from 7 and 8 with ∧I)

9. $(isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2)) \Rightarrow isValid(c_2)$ [Replaced c with c_2 in PKI4]

10. $Bel_{e_1}((isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2)) \Rightarrow isValid(c_2))$ [from 10 and 2 with ⇒E]

11. $Bel_{e_1}((isValid(Sig(c_2)) \wedge (DS(c_2) \leq today) \wedge (today < DE(c_2)) \wedge \neg inCRL(c_2))) \Rightarrow Bel_{e_1}(isValid(c_2))$ [from 11 and BEL with ⇒E]

12. $Bel_{e_1}(isValid(c_2))$ [from 8 and 12 with ⇒E]

Trust reasoning process based on extended reciprocal provides us with trust relationships, and their properties, and these trust relationships can be regarded as initial trust. Initial trust is the beginning of the trust reasoning process. Change in the initial trust will affect the end result of the trust reasoning process. In our PKI scenario, the trust relationship between agent e_1 and its parent entity $TR(e_1, parent(e_1), validity)$ is considered as an initial trust. Therefore, based on the initial trust agent e_1 believes that certificate c_1 informed by its parent entity is valid. Having completed the trust reasoning process, we can therefore have $Bel_{e_1}(isValid(c_2))$ derived from the fact $Inf_{parent(e_1),e_1}(isValid(c_1))$.

3.4 Chapter Summary

As a result of our extension of reciprocal logic, we satisfy both criteria necessary to qualify as a candidate for the fundamental logic system underlying trust reasoning.

Application of a case study in the field of PKI showed that our approach is consistent in dealing with messages from other agents as a proposition as it is based on strong logic relevant. Modal operators and new trust axioms aid in the reasoning out beliefs of agents in public key infrastructures. In Public key infrastructure, trust relationships play an important role, especially in cases when an agent wants to know whether the certificate informed by another agent is valid or not. Usually, authors have focused on certification relationships [33, 20] instead of trust relationships. Our trust reasoning process focuses on trust relationships as shown in the running example.

One of the advantages of our approach is generality. Trust reasoning based on a new extension of reciprocal logic is general in terms that complex trust relationships between agents in multi-agent systems could be described as well as messages from other agents could be dealt with, and represented as individual constants as well as propositions. Using extended reciprocal various complex scenarios can be described. Also, it provides us with the concept of empirical, and logical theories. Empirical theorems are true things in a target domain, and Logical theory (Logical

Formulas) are undoubtedly true things [26]. Using these empirical, logical theories we can represent trust relationships in various domains. Table 3.1 also depicts the improvement points in our extension. Thus, we believe is an improvement in our new extension of reciprocal logic.

Table 3.1: Comparison between Reciprocal Logic and Extended Reciprocal Logic

Features of Logic Systems	**Reciprocal Logic**	**Extended Reciprocal logics**
Based on relevant logic	√	√
Deals with trust properties	×	√
Deals with complex trust relationships	×	√
Deals message as individual constant	×	√
Deals message as proposition	×	√

Chapter 4

A Belief Revision Mechanism with Trust Reasoning based on Extended Reciprocal Logic

4.1 Neccesity of Belief Revision

In open multi-agent systems, an intelligent agent receives messages from other agents. Upon receiving a message, it does belief revision. A belief revision includes, i) a trust reasoning process, i.e., it obtains new belief related to the messages, and deduces implicitly unknown beliefs from the obtained belief; ii) in the case of contradiction in the belief set, it resolves the contradiction. So, trust reasoning (as discussed in detail in chapter 2), and belief revision must be included in the decision-making process of an intelligent agent in multi-agent systems.

Belief revision is a process of solving a contradiction in a target belief set to keep the belief set consistent. A belief set is consistent if and only if the set does not include both a proposition and its negation. In an open multi-agent system in the real world, the belief set of an agent is not always consistent, because a given assumption and an observed fact, or a previously observed fact and the current observed fact are sometimes explicitly or implicitly contradicted. Thus, an agent should be able to do belief revision. Moreover, in general, a trust relationship is not an eternal relationship. Although an agent is trusted at a point in time, the agent will not be trusted at another point in time. Changing trust relationships among agents, an agent updates its belief set by belief revision.

Although a belief revision mechanism with trust reasoning is demanded to construct multi-agent systems. However, currently, there is no such belief revision mechanism. On one hand, the best-known work on modeling belief revision is the so-called Alchourrón, Gärdenfors, and Makinson's (AGM) theory or AGM model [34, 24, 42]. But the AGM model is not suitable for the belief revision mechanism with trust reasoning because the AGM model adopts classical mathematical logic [12]. Classical mathematical logic is a suitable logic system underlying proving but not reasoning [12]. On the other hand, a well-known belief revision mechanism is the so-called truth maintenance systems, belief revision systems, or reason maintenance systems [18]. It is important to note that truth maintenance systems are not dependent upon a specific logic system. However, there is no truth maintenance system based on a logic system underlying trust reasoning.

For such purpose, we proposed a belief revision mechanism with trust reasoning based on extended reciprocal logic for multi-agent systems. The belief revision mechanism is a Doyle's-style approach (truth maintenance system approach) to deal with the inconsistency in an agent's belief set. The proposed mechanism uses the concept of a derivation path. A derivation path can be viewed as a representation of a belief set that is gradually developed and modified as a result of changes in trust relationships with other agents. If a contradiction occurs in the belief set, a revision process is triggered which allows forward and backtracking within the derivation path to track beliefs that cause inconsistency in the agent' s belief set. Detailed information about how the belief revision mechanism works and its application will be discussed in the following sections.

4.2 Overview of Belief Revision Mechanism

Each time an agent in a domain receives a message from another agent, it undergoes a series of steps as depicted in figure 4.1. The belief revision mechanism is comprised of two parts [11]. First, trust reasoning based on extended reciprocal logic is applied to the deduction process. Extended reciprocal logic is a candidate for a suitable logic system underlying trust reasoning. In the second part, each agent in multi-agent systems should revise their beliefs if the deduced beliefs resulting from the trust reasoning process conflict with pre-existing beliefs. Belief revision resolves the contradiction to maintain consistency within the agent's belief set.

Generally, in decision-making, new information leads to retractions of previously existing beliefs. Retraction and revision of existing beliefs are referred to as non-monotonic reasoning. Due to the fact that extended reciprocal logic is based on strong relevant logic, we are able to achieve non-monotonic trust reasoning, while logic systems based on classical mathematical logic are monotonic, i.e., new messages lead to explosions if they are inconsistent. So, the presented belief revision mechanism is a Doyle's-style approach (truth maintenance system approach) based on a logic system underlying trust reasoning.

According to the proposed belief revision mechanism, if a contradictory belief is entered into the belief set, a belief revision is initiated to work backward through the path following the belief contained in the label, seeking to determine which belief may have contributed to the contradiction. In order to eliminate the contradiction, some of the existing beliefs are removed from the set of beliefs and uses the labels once again to remove all deductions that originated from these beliefs from the set of current beliefs. Following are the details of each sub-process of the belief revision mechanism.

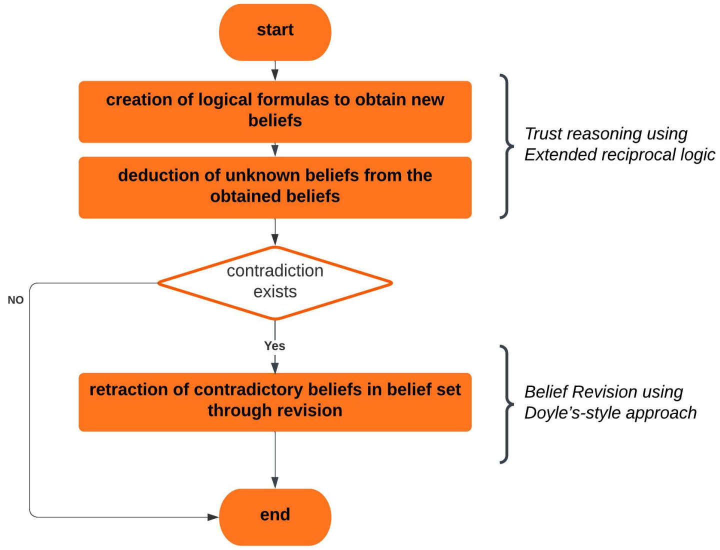

Figure 4.1: Belief Revision Mechanism with Trust Reasoning

4.2.1 Creation of logical formulas to obtain new beliefs related to the received message

Upon receiving messages from other agents within a domain, new beliefs are obtained by generating logical formulas. To generate a logical formula that indicates that an agent has informed another agent about a message, e.g., $m is valid$ is a message informed by agent b to agent a, and as a predicate, it will be represented as $isValid(m)$. Then its related logical formula will be generated using a related logical operator as $Inf_{b,a}(isValid(m))$.

4.2.2 Deduction of unknown beliefs from the obtained beliefs

Through trust reasoning using axioms, and inference rules from the extended reciprocal logic. This deduced implicit unknown beliefs from the obtained beliefs, and this deduced belief becomes the part of agent's belief set. Each agent maintains a belief set as a derivation path. Deduced beliefs are entered into the derivation path. As a result of the deduction process, an agent gradually adds or modifies its beliefs. As new beliefs are added to the belief set at each time instance, the derivation path evolves over time. Additionally, the derivation path identifies which inference rule was utilized, as well as which beliefs were used as premises or sources using the labeled formula concept.

A deduced belief in a derivation path is labeled with the time stamp, i.e., an integer indicating the instance at which this occurred. The time stamp serves as an index indicating the logical formula position in the belief set. Since these deduced beliefs are derived from premises using inference rules. These labels contain a record of which inference rule was used, as well as which beliefs were used as premises, or sources. This way agent knows all the logical consequences of each logical formula in his belief set. A label is defined as an ordered 5-tuple (index, from, to, epistemic entrenchment factor, status) [41], where:

1. index is a non-negative integer, the index, representing the position of the deduced belief in the belief set.

2. from-list contains information about premises, and inference rules used to derive the deduced belief.

3. to-list contains an index of all deduced beliefs where the given deduced belief serves as a premise.

4. epistemic entrenchment factor eef, indicates the value which assists in making decisions regarding belief retraction. In the current example, it is agreed that for all beliefs in the beliefs set the eef values will be 0 whereas if the derived belief includes already deduced beliefs as premises then the value will be 1.

5. status, using values *on* and off, indicates that only beliefs with status *on* can be used as premises in the deduction process. Whenever a deduced belief is first entered into the belief set, it is assigned status *on*.

4.2.3 Retraction of Contradictory Belief

Trust reasoning deduces beliefs that sometimes contradict pre-existing beliefs in the agent's belief set. Upon contradiction, a revision procedure is triggered, which disbelieves previously held beliefs, thus retracting the belief set by the contradictory belief. Usually, beliefs can be obtained as a message received from other agents in the domain, or they can be derived from the trust reasoning process. The procedure has three steps :

1. By backtracking through the belief set, starting with the from-list in the label of the contradictory belief, identify the beliefs that were involved in the derivation of the contradictory belief causing inconsistency in the belief set.

2. Change the status of involved beliefs to off, as many as necessary to invalidate the derivation of the given contradictory belief. The decision as to which status to turn off can be decided by retracting the one that is least believed generally identified by epistemic entrenchment value. In cases where all the involved beliefs are equally believed, a random choice can be made. In some systems, this retraction process may be automated, and in others, it may be human-assisted [41].

3. Forward chains using the to-lists, identify all beliefs whose derivations were based on the retracted belief, and put their status to off as well.

This retraction of beliefs will include those beliefs that cause the agent's belief set to be inconsistent. Changing a belief's status from on to off occurs whenever a contradiction occurs. The objective of the revision procedure is to remove such contradictory beliefs in the agent's belief set.

The following sections will discuss the application of the belief revision mechanism in two case studies, a scenario about public key infrastructure, and a scenario about a spy novel.

4.3 Application of Belief Revision Mechanism

4.3.1 Public Key Infrastructure PKI

Scenario

In the PKI scenario, agents e_1, e_2, and e_3 exchange messages as certificates among themselves. Agent e_1 is informed about certificate c_1 by the parent of the agent. Agents e_2 and e_3 inform agent e_1 about certificates c_2 and c_3 respectively. Agent e_1 doesn't believe the certificates c_2 and c_3 but wishes to use them. Therefore, based on the trust relationships between agents, messages such as certificates can be reasoned out as beliefs through trust reasoning. Later, agent e_4 informs that c_2 is not valid, here if the deduced belief through the trust reasoning process contradicts the existing beliefs of agent e_1 belief set revision process will be invoked.

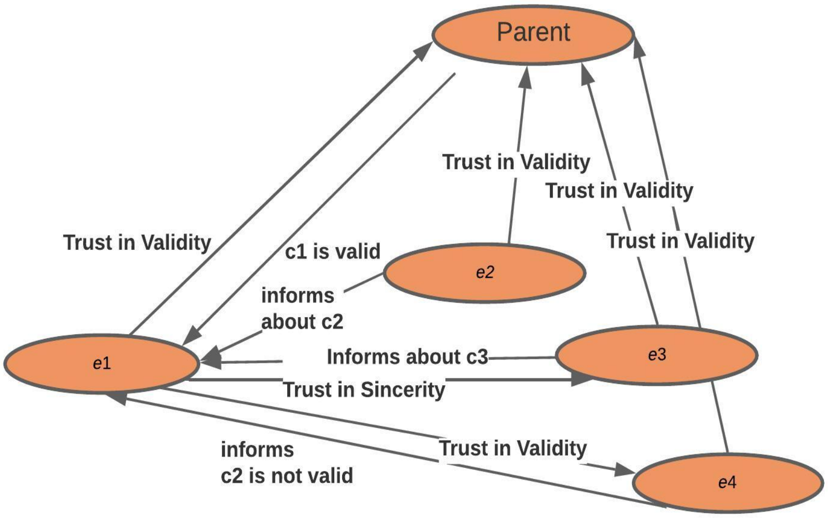

Figure 4.2: Public Key Infrastructure Scenario with Multi-agents

Formalization

To formalize the above scenario, we defined the following constants, functions, and predicates.

- Individual variables:
  - e: an agent
  - c, c': certifications
- Individual constants:
  - e_1, e_2, e_3, e_4: agents
  - c_1, c_2, c_3, c_4: certifications
  - today: date of today
- Functions:
  - $I(c)$: Issuer of certification c.
  - $S(c)$: Subject of certification c.
  - $PK(c)$: Public key of c.
  - $SK(c)$: Share key of c.
  - $DS(c)$: Start date of c.
  - $DE(c)$: End date of c.
  - $Sig(c)$: Signature of c.
  - $\mathit{parent}(e)$: The parent of agent e.
- Predicates:
  - $\mathit{inCRL}(c)$: c is in certification revocation list.
  - $\mathit{isValid}(x)$: x is valid.
  - $\mathit{isSigned}(x, k)$: x is message signed by key k.
  - $x = y$: x is equal to y.
  - $x \leq y$: x is equal to or less than y.
  - $x < y$: x is less than y.

Empirical theories of PKI

We can assume the following empirical theories.

PKI1: $\forall e(TR(e, \mathit{parent}(e), \mathit{validity}))$
(Any agent trusts its parent agent in validity.)

PKI2: $\forall c(\exists c'((\mathit{isValid}(c')))\wedge(I(c) = S(c'))\wedge(\mathit{isSigned}(c, PK(c')))) \Rightarrow \mathit{isValid}(Sig(c)))$

PKI3: $\forall c((isValid(Sig(c)) \wedge (DS(c) \leq today) \wedge (today < DE(c)) \wedge \neg inCRL(c)) \Rightarrow isValid(c))$
(PKI2 and PKI3 allow to verify the signature, and certificate itself on the basis of another certificate whose validity has been proven.)

Logical theories

We can assume the following logical formulas.

P1-1: $I(c_2) = S(c_1)$
(These observed facts are used as premises in our reasoning process and it is true in this scenario only.)

P1-2: $I(c_3) = S(c_1)$

P2-1: $isSigned(c_2, PK(c_1))$
(A certificate c_2 is signed by the subject of certificate c_1 with the private key corresponding to the public key of c_1.)

P2-2: $isSigned(c_3, PK(c_1))$

P3-1: $Inf_{parent(e_1),e_1}(isValid(c_1))$
(The parent agent of e_1 has informed e_1 about "certificate c_1 is valid".)

P3-2: $Inf_{parent(e_3),e_1}(isValid(c_3))$

P3-3: $Inf_{parent(e_4),e_1}(\neg isValid(c_2))$

P4: $TR(e_1, e_3, sincerity)$ $(assumption)$

P4-1: $TR(e_1, e_4, validity)$ $(assumption)$

P5-1: $DS(c_2) \leq today$ $(assumption)$

P5-2: $DS(c_3) \leq today$ $(assumption)$

P6-1: $today < DS(c_2)$ $(assumption)$

P6-2: $today < DS(c_3)$ $(assumption)$

P7-1: $\neg inCRL(c_2)$ $(assumption)$

P7-2: $\neg inCRL(c_3)$ $(assumption)$

Trust Reasoning Process using Extended Reciprocal Logic

Case 1: Agent e_1 receive certificate c_1 as a message from its parent

1. $\mathit{Inf}_{\mathit{parent}(e_1),e_1}(\mathit{isValid}(c_1)) \Rightarrow \mathit{isValid}(c_1)$ [from PKI1, Ercl2 with ⇒E]
2. $\mathit{isValid}(c_1)$ [from P3-1, 2]
3. $\mathit{Bel}_{e_1}(isValid(c_1))$ [from 2 with$Bel{-}Nec$]

Case 2: Agent e_1 receive certificate c_2 as a message from e_2

4. $(I(c_2) = S(c_1)) \wedge \mathit{isSigned}(c_2, \mathit{PK}(c_1)))$ [from $P1-1$ and $P2-1$ with ∧I]
5. $\mathit{Bel}_{e_1}((I(c_2) = S(c_1)) \wedge \mathit{isSigned}(c_2, \mathit{PK}(c_1)))$ [from 4 with $\mathit{Bel}{-}Nec$]
6. $(\mathit{isValid}(c_1)) \wedge (I(c_2) = S(c_1)) \wedge \mathit{isSigned}(c_2, PK(c_1)) \Rightarrow \mathit{isValid}(\mathit{Sig}(c_2))$ [Replaced c with c2 and c with c1 in PKI2]
7. $\mathit{Bel}_{e_1}(isValid(c_1)) \wedge (I(c_2) = S(c_1)) \wedge \mathit{isSigned}(c_2, \mathit{PK}(c_1)) \Rightarrow \mathit{isValid}(\mathit{Sig}(c_2))$ [from 6 with $\mathit{Bel}{-}Nec$]
8. $\mathit{Bel}_{e_1}(isValid(c_1)) \wedge (I(c_2) = S(c_1)) \wedge \mathit{isSigned}(c_2, \mathit{PK}(c_1)) \Rightarrow \mathit{Bel}_{e_1}\mathit{isValid}(\mathit{Sig}(c_2))$ [from BEL and 7 with ⇒E]
9. $\mathit{Bel}_{e_1}\mathit{isValid}(\mathit{Sig}(c_2))$ [from 5 and 8 with ⇒E]
10. $\mathit{Bel}_{e_1}(\mathit{DS}(c_2) \leq \mathit{today})$, $\mathit{Bel}_{e_1}(\mathit{today} < \mathit{DS}(c_2))$, $\mathit{Bel}_{e_1}\neg\mathit{inCRL}(c_2)$ [from each of P5-1, P6-1, and P7-1 with $\mathit{Bel}{-}Nec$]
11. $\mathit{Bel}_{e_1}\mathit{isValid}(\mathit{Sig}(c_2)) \wedge (\mathit{DS}(c_2) \leq \mathit{today}) \wedge (\mathit{today} < \mathit{DE}(c_2)) \wedge \neg\mathit{inCRL}(c_2)$ [from 10 with $\wedge I$]
12. $\mathit{isValid}(\mathit{Sig}(c_2)) \wedge (\mathit{DS}(c_2) \leq \mathit{today}) \wedge (\mathit{today} < \mathit{DE}(c_2)) \wedge \neg\mathit{inCRL}(c_2) \Rightarrow \mathit{isValid}((c_2))$ [Replaced c with c2 in PKI3]
13. $\mathit{Bel}_{e_1}\mathit{isValid}(\mathit{Sig}(c_2)) \wedge (\mathit{DS}(c_2) \leq \mathit{today}) \wedge (\mathit{today} < \mathit{DE}(c_2)) \wedge \neg\mathit{inCRL}(c_2) \Rightarrow \mathit{isValid}((c_2))$ [from 12 with $\mathit{Bel}{-}Nec$]
14. $\mathit{Bel}_{e_1}\mathit{isValid}(\mathit{Sig}(c_2)) \wedge (\mathit{DS}(c_2) \leq \mathit{today}) \wedge (\mathit{today} < \mathit{DE}(c_2)) \wedge \neg\mathit{inCRL}(c_2) \Rightarrow \mathit{Bel}_{e_1}\mathit{isValid}(c_2)$ [from BEL and 13 with ⇒E]
15. $\mathit{Bel}_{e_1}\mathit{isValid}(c_2)$ [from 11 and 14 with ⇒E]

    In cases 1 and 2, beliefs $\mathit{Bel}_{e_1}\mathit{isValid}(c_1)$ and $\mathit{Bel}_{e_1}\mathit{isValid}(c_2)$ are deduced from the trust reasoning process, and these deduced beliefs will be entered into the agent's belief set with their labels, i.e. labels of beliefs $\mathit{Bel}_{e_1}\mathit{isValid}(c_1)$ and $\mathit{Bel}_{e_1}\mathit{isValid}(c_2)$ will be (3, (2, $\mathit{Bel}{-}Nec$), (7, 8), 0, On), and (15, (11, 14, =>E), {}, 1, On) respectively.

Case 3: Agent e1 received certificate c3 as a message from agent e3

16. $(isValid(c_1)) \wedge (I(c_3) = S(c_1)) \wedge isSigned(c_3, PK(c_1))$ [from 2, P1-2, and P2-2 with ∧I]
17. $\exists c'((isValid(c')) \wedge (I(c_3) = S(c')) \wedge (isSigned(c_3, PK(c'))) \Rightarrow isValid(Sig(c_3))$ [Substitute c3 for c in PKI2]
18. $isValid(Sig(c_3))$ [from 16 and 17 with ⇒E]
19. $isValid(Sig(c_3)) \wedge (DS(c_3) \leq today) \wedge (today < DE(c_3)) \wedge \neg inCRL(c_3)$ [from 18 and P5-2, P6-2, and P7-1 with ∧I]
20. $isValid(Sig(c_3)) \wedge (DS(c_3) \leq today) \wedge (today < DE(c_3)) \wedge \neg inCRL(c_3) \Rightarrow isValid(c_3)$ [Substitute c3 for c in PKI3]
21. $isValid(c_3)$ [Deduced from 19 and 20 with ⇒E]
22. $Inf_{e3,e1}(A) \Rightarrow Bel_{e_3}(A)$ [from P3-2 and ERcL1 with ⇒E]
23. $Bel_{e_1}(isValid(c_3))$ [from P4 and 22 with ⇒E]

    In case 3, $Bel_{e_1} isValid(isValid(c_3))$ is deduced from the trust reasoning process, and deduced belief will be entered into the agent's belief set with its respective label (23, (P4, 21, ⇒E), $\{\}, 0, On$).

Case 4: Agent e_1 receive a message about the certificate c_2 from agent e_4

24. $Inf_{e_4,e_1}(\neg isValid(c_2)) \Rightarrow \neg isValid(c_2)$ [from P4-1, ERcL2 with $\Rightarrow E$]
25. $\neg isValid(c_2))$ [from P3-3, 25]
26. $Bel_{e_1} \neg isValid(c_2)$ [from 25 with $Bel{-}Nec$]

    In case 4, $Bel_{e_1} \neg isValid(c_2)$ is deduced and deduced belief will be entered into the agent's belief set with its respective label (26, (25, ⇒E), $\{\}, 0, On$).

Revision process

Belief set of agent e_1 represented as BS_{e_1}=$\{\}$. Initially, the belief set will be emptily represented as $BS_{e_1} = \phi$. A belief can be obtained in two ways, i) A belief can be received as a message from other agents in the domain; ii) A belief can be derived as a deduced belief from the trust reasoning process. So, until now four beliefs are part of the agent belief set. Currently, agent e_1 belief set has BS_{e_1}=$\{Bel_{e_1}(isValid(c_1)), Bel_{e_1}(isValid(c_2)), Bel_{e_1}(isValid(c_3), Bel_{e_1} \neg isValid(c_2)\}$. Beliefs are retained in the agent's belief set with their labels which helps to maintain the derivation path. Entries of other beliefs are handled in a similar manner. Now the belief set of agent e_1 consists of two contradictory beliefs, i.e.,

$el_{e_1}(isValid(c_2))$ and $Bel_{e_1}(\neg isValid(c_2))$. Based on the epistemic values indicated in their respective labels as 1, and 0. So, belief with the least entrenched values will be selected to retract, so $Bel_{e_1}(\neg isValid(c_2))$ will be retracted. After retraction, a new belief set will be $BS_{e_1}=\{Bel_{e_1}(isValid(c_1)), Bel_{e_1}(isValid(c_2)), Bel_{e_1}(isValid(c_3)\}$.

Additionally, if the belief set contains beliefs that are equally believed, a random choice may be made, e.g., belief $Bel_{e_1}(isValid(c_1))$, and $Bel_{e_1}(\neg isValid(c_2))$ has same epistemic entrenchment value and belief $Bel_{e_1}(isValid(c_1))$ is selected to be retracted than revision procedure forward chains through to-lists, changing the status of deduced belief at 7, and 8 from on to off. To this point, beliefs $Bel_{e_1}(isValid(c_1))$, $Bel_{e_1}(isValid(c_2))$ will have their statuses off, leaving $BS_{e_1}=\{Bel_{e_1}(isValid(c_2)), Bel_{e_1}(isValid(c_3)), Bel_{e_1}\neg isValid(c_2)\}$. Using this method, agents would retain their beliefs, but their status would be set to off. As a result, it will be possible to trace the beliefs, but at the same time prevent the agent from re-acquiring them, therefore making belief revisions a practical, and useful process.

4.3.2 Spy Novel Scenario

We considering another scenario from [16] in which multiple agents exchange messages with each other as an information source.

We consider three agents a_1, b_1, and c_1 who are interested to exchange information about the two facts "there is a spy in the train T", denoted by p_1, and "the train T has arrived at the railway station", denoted by q. In this situation agent a_1 trusts b_1 in regard to his validity for p_1, and in regard to his sincerity for q_1, and a_1 trusts c_1 in regard to his completeness for q_1. a_1 trust may be supported, for instance, by the fact that b belongs to some intelligence service, and c_1 is an employee of the railway station who stands on the platform where the train is supposed to arrive. In this situation, b_1 has informed a_1 information p_1, and he has also informed q_1, and c_1 has not informed a_1 information q_1. The formalization of the above scenario is as follows :

Formalization

- Individual variables:
    - agents: a, b, c
    - facts: p, q
- Individual constants:
    - agents: a_1, b_1, c_1
    - facts: p_1, q_1
- Predicates:
    - $isFact(x)$: x is a fact.

Empirical and Logical theories

We can assume the following theories.

IS1: $TR(a_1, b_1, validity)$
(Agent a_1 trusts b_1 in his validity)

IS2: $TR(a_1, b_1, sincerity)$
(Agent a_1 trusts b_1 in his sincerity)

IS3: $TR(a_1, c_1, completeness)$
(Agent a_1 trusts c_1 completeness)

IS3-1: $TR(a_1, c_1, sincerty)$
(Agent a_1 trusts c_1 sincerity)

IS4: $Inf_{b_1,a_1}(isFact(p_1))$
(b_1 has informed to a_1 about $isFact(p_1)$)

IS5: $\neg Inf_{c_1,a_1}(isFact(q_1))$
(c_1 has not informed to a_1 about $isFact(q_1)$)

IS6: $Inf_{c_1,a_1}(\neg isFact(q_1))$
(c_1 has informed to a_1 about $\neg isFact(q_1)$)

IS7: $\neg Inf_{b_1,a_1}(isFact(q_1))$
(b_1 has not informed to a_1 about $isFact(p_1)$)

Trust Reasoning Process

Case 1: Agent a_1 receive information about p_1 as a message from agent b_1

1. $Inf_{b_1,a_1}(isFact(p_1) \Rightarrow isFact(p_1))$(from $IS1$ and $ERcL2$ with ⇒E)
2. $isFact(p_1))$ [from IS4 and 1 with $\Rightarrow E$]
3. $Bel_{a_1}(isFact(p_1)$ [from 2 with Bel–Nec]

   After deduction, we have $Bel_{a_1}(isFact(p_1)$. The deduced belief will be added to the belief set of agents a_1 with its respective label (3, (2, Bel–Nec), 11, 0, On).

Case 2: Agent a_1 receive information about q_1 as a message from agent c_1

4. $Bel_{a_1}(\neg Inf_{c_1,a_1}(isFact(q_1))$ [from IS5 with BEL–Nec]
5. $A \Rightarrow Inf_{c_1,a_1}(A)$ [from IS3 and ERcL6 with $\Rightarrow E$]
6. $(isFact(q_1) \Rightarrow Inf_{c_1,a_1}(isFact(q_1)$ [from 5]

7. $\neg Inf_{c_1,a_1}(isFact(q_1)) \Rightarrow \neg isFact(q_1))$ [contraposition of 6]
8. $Bel_{a_1}(\neg Inf_{c_1,a_1}(isFact(q_1)) \Rightarrow \neg isFact(q_1))$ [from 7 with BEL–Nec]
9. $Bel_{a_1}(\neg Inf_{c_1,a_1}(isFact(q_1)) \Rightarrow (\neg isFact(q_1)))$ [from 8 with BEL]
10. $Bel_{a_1}(\neg Inf_{c_1,a_1}(isFact(q_1))$ [from 4 and 9 with $\Rightarrow E$]
11. $Bel_{a_1}(isFact(p_1) \wedge \neg isFact(q_1))$ [from 3 and 10 with $\wedge I$]

    After deduction we have $Bel_{a_1}(isFact(p_1) \wedge \neg isFact(q_1))$. The deduced belief will be added to the belief set of agents a_1 with its respective label (11, (3, 10, $\wedge I), \{\}, 1, On)$.

    Case 3: Agent a_1 receive information about p_1 as a message from agent c_1 with a change in a trust relationship.

12. $Inf_{b_1,a_1}(A) \Rightarrow Bel_{b_1}(A)$ [from IS2 and ERcL1 with $\Rightarrow E$]
13. $Inf_{b_1,a_1}(isFact(p_1)) \Rightarrow Bel_{b_1}(isFact(p_1))$ [from 12]
14. $Bel_{b_1}(isFact(p_1))$ [from IS4 and 13 with $\Rightarrow E$]
15. $Bel_{a_1}(Inf_{c_1,a_1}(\neg isFact(q_1))$ [from IS6 with Bel–Nec]
16. $Inf_{c_1,a_1}(A) \Rightarrow Bel_{c_1}(A)$ [from IS3-1 and ERcL1 with $\Rightarrow E$]
17. $Inf_{c_1,a_1}(\neg isFact(p_1)) \Rightarrow Bel_{c_1}(\neg isFact(p_1))$ [from 16]
18. $Bel_{c_1}(\neg isFact(p_1))$ [from IS6 and 17 with $\Rightarrow E$]
19. $Bel_{a_1}(Bel_{c_1}(\neg isFact(p_1))$ [from 18 with BEL–Nec]
20. $Bel_{a_1}(Bel_{b_1}(isFact(p_1))$ [from 14 with BEL–Nec]
21. $Bel_{a_1}(Bel_{b_1}(isFact(p_1) \wedge Bel_{c_1}(\neg isFact(p_1))$ [from 19 and 20 with $\wedge I$]

After the trust reasoning process, $Bel_{a_1}(Bel_{b_1}(isFact(p_1) \wedge Bel_{c_1}(\neg isFact(p_1))$ has been deduced. The deduced result will be added to the belief set of agent a1 with its respective label (21, (19, 20, $\wedge I), \{\}, 0, On)$.

Change in a trust relationship from completeness to sincerity between agent a1 trusts c1 deduces different reasoning results. Therefore, it is evident from the deduced results that a change in trust relationships leads to different deduced results. Moreover, $Bel_{a_1}(isFact(p_1)$, and $Bel_{a_1}(isFact(p_1) \wedge \neg isFact(q_1))$ has respective labels of 0,1.

Belief $Bel_{a_1}(isFact(p_1)$ with the least entrenched value will be selected. After that, the revision procedure forwards chains through to, and from lists, and changes the status of beliefs from *on* to off. To this point, the contradictory belief causing inconsistency will have their statuses both subsequent beliefs will have their statuses off, leaving $Bel_{a_1}(Bel_{b_1}(isFact(p_1) \wedge Bel_{c_1}(\neg isFact(p_1))$, and $Bel_{a_1}(isFact(p_1) \wedge \neg isFact(q_1))$ in the belief set. Using this method, agents would retain their beliefs, but their status would be set to off. As a result, it will be possible to trace the beliefs, but at the same time prevent the agent from re-acquiring them. Thus, the resulting belief set will be consistent.

4.4 Chapter Summary

In this chapter, we presented a belief revision mechanism with trust reasoning based on extended reciprocal logic (ERL) for multi-agent systems. A single mechanism that includes trust reasoning, and belief revision for the decision-making process of an agent in multi-agent systems. Trust reasoning based on ERL is used for the deduction process because extended reciprocal logic is a suitable logic system underlying trust reasoning. As a result, an agent maintains its belief set. If a contradiction occurs in the agent's belief set, a revision process based on Doyle' s procedural approach is triggered. Doyle's procedural approach uses the concept of derivation path which allows forward, and backtracking to track beliefs that cause inconsistency in the agent's belief set. Furthermore, we demonstrated the application of the belief revision mechanism in the field of public key infrastructure PKI and spy novel. Both have different domains. A unique feature of the belief revision mechanism is that it is based on extended reciprocal logic, which makes it a general mechanism.

Chapter 5

Discussions

Although we have shown through an application of a case study in various domains, i.e., PKI and spy novel, that our approach is consistent because it is based on strong relevant logic, but still there are some problems that should be addressed.

Firstly, our definition of trust relationships is that trustors, trustees, and trust property are all considered individuals, i.e., the trust relationship is based on a single trustor, single trustee, and one piece of property. But it cannot describe trust relationships based on a group of trusters, a group of trustees, and a group of properties. It is possible that in some cases all possible entities in the trustee set trust all possible entities in the trustee set possessing a specified property, or sets of properties.

Secondly, usually, a trust relationship has some specified conditions, or requirements. So, a parameter of conditions should be included in the definition of trust relationships, i.e. Under certain conditions, a trustee will trust a trustee, holding a trust property. So trust relationships with better abstractions, and more predicates are needed to deal with special cases of trust relationships between agents.

Thirdly, our current extension provides limited trust properties predicates and axioms that are more toward the rationality of agents such as sincerity, cooperativity, etc. In order to ensure the security of agents, we need to provide axioms, and predicates to deal with a number of properties such as authenticity, reliability, etc.

The second phase of our thesis deals with the belief revision process in a multi-group system. However, the limitation of the belief revision mechanism is that currently, epistemic entrenchment factors consider values of 0, and 1. In the future, we need to consider a range of values from 0 to 1 in order to deal with different degrees of beliefs.

Chapter 6

Concluding Remarks

6.1 Contributions

Multi-agent system refers to a group of agents in which the agents interact with each other as well as the environment in order to achieve goals. In the scope of this thesis, we consider the agents whose activities can be described in general as follows: The agent is associated with an environment or a problem domain of interest and carries representation or some prior knowledge about trust relationships with other agents in the domain. Its goal is to maintain its own belief state in some desirable way. To do so, it takes action from time to time. For each action, it makes observations about the other agents in a domain, calculates the trust relationships of agents in the domain based on the observations and its prior knowledge about the domain, maintains its own belief state, and determines the most appropriate action based on its belief state. We refer to the activity of calculating the trust relationship with other agents in the domain from prior knowledge and observations as reasoning.

For such purpose, we define conditions that are necessary for the right fundamental logic system underlying trust reasoning. The necessary condition to qualify for a right fundamental logic is that first, it should not be a conservative extension of classical mathematical logic rather it should be based on strong relevant logic, and second, it should deal with complex trust relationships including such trust properties Current reciprocal logic by Cheng doesn't fulfill both conditions, so we introduced a logic system that satisfies both conditions.

Extension of current reciprocal logic includes surveying and identifying trust properties to represent complex trust relationships for trust reasoning. Extended reciprocal logic is extended with new axioms, and modal operators to deal with messages receiving other agents as a proposition. We have also depicted the usage of our extension in the domain of PKI. The description of extension and its applications can be found in Chapter 2, and Chapter 3.

As the next step, we proposed a belief revision mechanism with trust reasoning based on extended reciprocal logic (ERL) for multi-agent systems. The belief revision mechanism is comprised of two parts. First, trust reasoning based on extended reciprocal logic is applied to the deduction process. Extended reciprocal logic is a candidate for a suitable logic system underlying trust reasoning. In the second part, each agent in multi-agent systems should revise their beliefs if the deduced beliefs resulting from the trust reasoning process conflict with pre-existing beliefs. Belief revision resolves the contradiction to maintain consistency within the agent's belief set. The description of the belief revision mechanism can be found in Chapter 4.

6.2 Future Work

Currently, we provided syntactic structure without the aid of semantics syntax of extended reciprocal logic, in the future we will provide, and specify the semantics of extended reciprocal logic. We will also look into providing a hybrid approach by calculating trust properties with probability, or statistical approaches as individuals, and then using the logic-based approach to combine the results of that calculations. It will be a hybrid of quantitative, and qualitative approaches. Despite the fact that our extension is concerned with the change in a relationship over time t, it is important to consider predicates from spatial and temporal logic-based systems as a hybrid approach as well in order to maximize the effectiveness and value of our extension. Also, we will analyze and apply our belief vision mechanism based on our extension in various domains to verify its usefulness.

Publications

Refereed papers

- Sameera Basit and Yuichi Goto : An Extension of Reciprocal Logics for Trust Reasoning. In: Nguyen, N., Jearanaitanakij, K., Selamat, A., Trawiski, B., Chittayasothorn, S. (eds) “Intelligent Information and Database Systems. ACIIDS 2020.” Lecture Note in Computer Science, Vol. 12034, pp. 65–75. Springer, Cham (2020), `https://doi.org/10.1007/978-3-030-42058-1_6`
- Sameera Basit and Yuichi Goto : An Extension of Reciprocal Logic for Trust Reasoning: A Case Study in PKI. In: Nguyen, N., et al. (eds) “Intelligent information and Database Systems. ACIIDS 2022.” Lecture Note in Computer Science, Vol. 13757, Springer, Cham (2023). `https://doi.org/10.1007/978-3-031-21743-2_39`
- Sameera Basit and Yuichi Goto: A Belief Revision Mechanism with Trust Reasoning Based on Extended Reciprocal Logic for Multi-agent Systems. In: Wyld, D.C., Nagamalai, D. (eds) “Computer Networks Communications. 10th International Conference on Computer Networks Communications (CCNET 2023).” Vol. 13, No. 4, Vancouver, Canada (Hybrid), (2023).

References

1. Alfarez Abdul-Rahman: A framework for decentralized trust reasoning, Doctoral dissertation, University of London, 2005

2. Asmaa Adnane, Rafael de Sousa Jr, Christophe Bidan, and Ludovic Mè: Integrating trust reasonings into node behavior in OLSR, Proceedings of the 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks, Q2SWinet '07, pp. 152–155, Crete Island, Greece, 2007.

3. José Júlio Alferes, Pierangelo Dell'Acqua, Evelina Lamma, João Alexandre Leite, Luís Moniz Pereira, and Fabrizio Riguzzi: A logic based approach to multi-agent systems, The Association for Logic Programming Newsletter, vol. 14, No. 3, August 2001.

4. Leila Amgoud and Robert Demolombe: An argumentation-based approach for reasoning about trust in information sources, Argument Computation 5(2-3), pp. 191–215, 2014. `https://doi.org/10.1080/19462166.2014.881417`

5. "Alan Ross Anderson and Nuel D. Belnap Jr.: Entailment: The Logic of Relevance and Necessity, Princeton University Press, 1975.

6. "Alan Ross Anderson and Nuel D. Belnap Jr.: Entailment: The Logic of Relevance and Necessity, Princeton University Press, 1992.

7. Donovan Artz and Yolanda Gil: A survey of trust in computer science and the semantic web, SSRN Electronic Journal 5(2), pp 58–71, 2007. `https://doi.org/10.2139/ssrn.3199355`

8. Najwa Abu Bakar and Ali Selamat: Runtime verification of multi-agent systems interaction quality, in Selamat, A., Nguyen, N., and Haron. H., (Eds.), "Intelligent Information and Database Systems", pp. 435–444, Lecture Notes in Computer Science, 2013. `https://doi.org/10.1007/978-3-642-36546-1_4`

9. Sameera Basit and Yuichi Goto : An Extension of Reciprocal Logics for Trust Reasoning, in Nguyen, N., Jearanaitanakij, K., Selamat, A., Trawiski, B., Chittayasothorn, S. (Eds) "Intelligent Information and Database Systems, ACIIDS 2020," Lecture Note in Computer Science, Vol. 12034, pp. 65–75. Springer, 2020. `https://doi.org/10.1007/978-3-030-42058-1_6`

10. Sameera Basit and Yuichi Goto: An Extension of Reciprocal Logic for Trust Reasoning: A Case Study in PKI, in Nguyen, N., et al. (Eds) “Intelligent information and Database Systems, ACIIDS 2022,” Lecture Note in Computer Science, Vol. 13757, Springer, 2023. `https://doi.org/10.1007/978-3-031-21743-2_39`

11. Sameera Basit and Yuichi Goto: A Belief Revision Mechanism with Trust Reasoning Based on Extended Reciprocal Logic for Multi-agent Systems, in Wyld, D.C., Nagamalai, D. (Eds) “Computer Networks Communications, 10th International Conference on Computer Networks Communications (CCNET 2023),” Vol. 13, No. 4, Vancouver, Canada (Hybrid), 2023.

12. Jingde Cheng: A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery, in E. Kawaguchi, H. Kangassalo, H. Jaakkola, and I. A. Hamid (Eds.), ”Information Modelling and Knowledge Bases XI,” Frontiers in Artificial Intelligence and Applications, Vol. 61, pp. 136-159, IOS Press, February 2000.

13. Jingde Cheng: Reciprocal logic: Logics for specifying, verifying, and reasoning about reciprocal relationships. in Khosla, R., Howlett, R.J., and Jain, L. C., (Eds.), ”Knowledge-Based Intelligent Information and Engineering Systems”, pp. 437–445. Springer Berlin Heidelberg, 2005.

14. Kinzang Chhogyal, Abhaya C. Nayak, Aditya Ghose and Hoa Khanh Dam : A Value-based Trust Assessment Model for Multi-agent Systems, in Kraus, S., (Eds.), ”Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence”, pp. 195–200, ijcai.org, 2019.

15. Robert Demolombe: Reasoning About Trust: A Formal Logical Framework, in Jensen, C., Poslad, S., Dimitrakos, T. (Eds.), ”Trust Management”, iTrust, Lecture Notes in Computer Science, vol. 2995, pp 291–303, 2004. `https://doi.org/10.1007/978-3-540-24747-0_22`

16. Robert Demolombe: To trust information sources: A proposal for a modal logical framework, in Castelfranchi, C., and Tan, Y., (Eds.), ”Trust and Deception in Virtual Societies”, pp. 111–124, Dordrecht, 2004.

17. Aris Dimeas and Nikos Hatziargyriou: Control Agents for Real Microgrids, Proceedings of 15th International Conference on Intelligent System Applications to Power Systems, pp. 1–5, 2009. `https://10.1109/ISAP.2009.5352865`

18. Jon Doyle: A truth maintenance system, Artificial Intelligence, Vol. 12, No. 3, pp 231-272, 1979. `https://doi.org/10.1016/0004-3702(79)90008-0`

19. Nagat Drawel: Model Checking Trust-based Multi-Agent Systems, Doctoral Dissertation, Concordia University, 2019.

20. Hanan El Bakkali and Bahia Idrissi Kaitouni: A logic-based reasoning about PKI trust model, Proceedings of Sixth IEEE Symposium on Computers and Communications, pp. 42-48, Hammamet, Tunisia, 2001. `https://10.1109/ISCC.2001.935353`

21. Tuan-Fang Fan and Churn-Jung Liau: A logic for reasoning about evidence and belief, Proceedings of the International Conference on Web Intelligence, WI '17, pp. 509–516, Leipzig, Germany, 2017.

22. André Ferreira, Henrique Lopes Cardoso, and Luis Paulo Reis: Dipblue: A diplomacy agent with strategic and trust reasoning, Proceedings of the International Conference on Agents and Artificial Intelligence, ICAART 2015, pp. 54-65, Lisbon, Portugal, SCITEPRESS - Science and Technology Publications, 2015, `https://doi.org/10.5220/0005205400540065`

23. Andreas Fuchs, Sigrid Gürgens, and Carsten Rudolph: A formal notion of trust –enabling reasoning about security properties, in Nishigaki, M., Jøsang, A., Murayama, Y., and Marsh, s., (Eds.), 4th IFIP WG 11.11 International on Trust Management (TM), Trust Management IV, pp. 200–215, Springer, 2010.

24. Dov M Gabbay, C J Hogger, and J A Robinson: Handbook of Logic in Artificial Intelligence and Logic Programming, Oxford University Press, 1998.

25. Salvatore Garruzzo and Domenico Rosaci: The roles of reliability and reputation in competitive multi-agent systems, in Meersman, R., Dillon, T., and Herrero, P., (Eds.), On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science, pp. 326-339, Springer Berlin, Heidelberg, 2010.

26. Yuichi Goto: Automated Forward Deduction Based on Strong Relevant Logics and Its Applications, Doctoral Dissertation, Saitama University, 2004.

27. Andriy Hnativ and Simone A. Ludwig.: Evaluation of trust in an ecommerce multi-agent system using fuzzy reasoning, Proceedings of the 18th International Conference on Fuzzy Systems, FUZZ-IEEE'09, pp. 757–763. IEEE Press, 2009.

28. Xiaowei Huang, Marta Kwiatkowska, and Maciej Olejnik: Reasoning about cognitive trust in stochastic multiagent systems, ACM Transactions on Computational Logic 20, pp. 1-64, 2019.

29. Audun Jøsang, Roslan Ismail, and Colin Boyd: A survey of trust and reputation systems for online service provision, Decision Support System 43(2), pp. 618–644, 2007.

30. Rabiya Khalid, Omaji Samuel, Nadeem Javaid, Abdulaziz Aldegheishem, Muhammad Shafiq, and Nabil Alrajeh: A secure trust method for multiagent system in smart grids using blockchain, IEEE Access 9, pp. 59848-59859, 2021.

31. Christopher Leturc and Grégory Bonnet: A normal modal logic for trust in the sincerity, Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18, pp. 175–183, Stockholm, Sweden, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

32. Churn-Jung Liau: Belief, information acquisition, and trust in multi-agent systems: A modal logic formulation, Artificial Intelligence 149(1), pp. 31–60, 2003.

33. Chuchang Liu, Maris A. Ozols, and Anthony Cant: An axiomatic basis for reasoning about trust in PKIs, in Varadharajan, V., Mu, Y., (Eds.), Information Security and Privacy, Lecture Notes in Computer Science, vol. 2119, pp. 274-291, Springer Berlin Heidelberg, 2001.

34. François Lévy: A survey of belief revision and updating in classical logic, International Journal of Intelligent Systems 9(1), pp. 29–59, 1994. `https://doi.org/10.1002/int.4550090104`

35. Stephen Marsh: Formalising Trust as a Computational Concept, Doctoral Dissertation, Ontario Tech University, 1999.

36. Cornelius Namiluko: An Architectural Approach for Reasoning about Trust Properties, Docotal Dissertaion, University of Oxford, 2012.

37. Linh Anh Nguyen: Reasoning about epistemic states of agents by modal logic programming, in Toni, F, and Torroni, P., (Eds.), Computational Logic in Multi-Agent Systems, pp. 37–56, Springer Berlin Heidelberg, 2006.

38. Birgit Lugrin, Catherine Pelachaud, and David Traum: The Handbook on Socially Interactive Agents: 20 years of Research on Embodied Conversational Agents, Intelligent Virtual Agents, and Social Robotics, Association for Computing Machinery New York NY United States, 2021.

39. Lars Rasmusson and Sverker Jansson: Simulated social control for secure internet commerce, Proceedings of the 1996 Workshop on New Security Paradigms, NSPW '96, pp. 18–25, Lake Arrowhead, California, USA, Association for Computing Machinery, 1996.

40. Steven Reece: Statistical reasoning in decentralised and distributed multi-agent systems, https://www.robots.ox.ac.uk/ argus/ papers/StatisticalReasoningin DecentralisedandDistributedMulti-agentSystems0NCAF2004Oxford.pdf (accessed 2022-01-12).

41. Daniel G. Schwartz: Nonmonotonic reasoning as a temporal activity, arXiv.org, abs/1404.7173, 2014.

42. Léa Sombé: A glance at revision and updating in knowledge bases, International Journal of Intelligent Systems 9(1), pp. 1–27, 1994. `https://doi.org/10.1002/int.4550090103`

43. Michael Wooldridge and Nicholas R. Jennings: Intelligent agents: theory and practice, The Knowledge Engineering Review 10(2), pp. 115–152, 1995.

44. Xiaona Xia, Jiguo Yu, and Baoxiang Cao: SCP-trust reasoning strategy based on preference and its service composition process of context-aware process, Journal of Computer and Communications 02(09), pp. 38–45, 2014.

45. Yang Xiang: Probabilistic Reasoning in Multi-Agent Systems: A Graphical Models Approach, Cambridge University Press, USA, 2002.

46. Mingde Zhang, Xuefeng Zheng, Shuwang Lv, and Yike Yu: Improved approach on modeling and reasoning about PKI/WPKI, Proceedings of 6th International Conference on Wireless Communications Networking and Mobile Computing, WiCOM, pp. 1–4, Chengdu City, China, IEEE, 2010.

47. Weiliang Zhao, Vijay Varadharajan, and George Bryan: Modelling trust relationships in distributed environments, in Katsikas, S., Lopez, J., and Pernul, G., (Eds.), Trust and Privacy in Digital Business, TrustBus 2004, Lecture Notes in Computer Science, vol. 3184, pp. 40–49, Springer Berlin Heidelberg, 2004.