Modelling and Evaluating Trust Relationships in Mobile Agents Based Systems

Ching Lin and Vijay Varadharajan

Information and Networked System Security Research
Department of Computing
Division of Information and Communication Sciences
Macquarie University
Sydney, NSW 2019, Australia
{linc, vijay}@ics.mq.edu.au

Abstract. This paper considers a trust framework for mobile agent based systems. It introduces the concept of trust reflection and presents a novel method using views of the trustor to derive the trust symmetry. This new approach addresses the trust initialization issues and allows for different levels of initial trust to be established based on trustor's initial assessment and then enables different trust dynamics to evolve in the course of future interactions in situations where full trust appraisal is not possible at the beginning of interactions. This is an important aspect for security in distributed systems, especially in mobile agent based systems due to the fact that the agent owner principals may not have all the information to appraise full trust in other entities(e.g. foreign hosts to be visited by the mobile agent) when composing an itinerary prior to the deployment of mobile agent. Our framework proposes a new formalism to capture and reason about trust reflection and presents an algorithm for updating trust during system operation. This is then applied to a simulated mobile agent system and an analysis of the results is presented.

Keywords: Security, Trust, Mobile Agent System

1 Introduction

Over the recent years, mobile agent technologies have been receiving a great deal of interest, as they have a lot to offer towards achieving the vision of usable distributed systems in a heterogeneous network environment. The ability to move computations across the nodes of a wide area network helps to achieve the deployment of services and applications in a more flexible, dynamic and customizable way than the traditional client-server paradigm. They provide several advantages over remote procedural call and message passing such as reduced network usage, increased asynchrony between clients and servers, increased concurrency and addition of client-specified functionality into servers. However, there lie some fundamental issues that need to be addressed in the area of security. The key to achieving this is the systematic understanding and developing a comprehensive

J. Zhou, M. Yung, Y. Han (Eds.): ACNS 2003, LNCS 2846, pp. 176–190, 2003.

security model for mobile agent systems [1]. Presently, the mobile agent security research focuses primarily on the design of cryptography based security mechanisms for the protection of mobile agents and hosts. Typical solutions include the following: a) for host protection: state appraisal [15], signed code [16] PCC [19]; b) for agent protection: execution tracing [23], computing with encrypted functions [21], hardware based protection [24]. c) combined host and agent security solutions include [18,1]. A more comprehensive summary on mobile agent security can be found in [14,17,18,20].

In this paper, we identify trust as a fundamental component of the mobile agent security architecture and propose a trust framework for mobile agent system. Though trust is often recognized as the foundation of secure systems, it is seldom represented and reasoned about explicitly when it comes to the practical design of secure systems and applications. Our aim in this paper is to represent trust explicitly and reason about it in the context of mobile agent based system operation. In particular, we aim to be able to reason about the trust properties such as the execution trust and mobile code trust, which will enable us to make more informed decisions to achieve our security goals in a mobile agent system.

The paper is organized as follows. Section 2 discusses some security related trust problems in a mobile agent system. It poses six trust questions that should be answered at different stages of the life cycle of a typical mobile agent operation and introduces the trust reflection problem. Section 3 proposes a new formalism to capture and reason about trust reflection and presents an algorithm for updating trust during system operation. Section 3.4 applies the new formalism and the algorithm to the mobile agent security problems introduced in Section 2. Finally, Section 5 concludes with some ideas for further work.

2 Security Related Trust Problems in Mobile Agent System

Conceptually, trust on an entity can be thought of as a combination of honesty, competence, reliability and availability of the entity with respect to interaction and cooperation. In this paper, we refine our approach to modelling trust in mobile agent systems by considering a new form of trust which we call *reflective trust* which we will introduce and discuss in detail in the following sections. Broadly speaking, in the context of mobile agents, we believe two types of trust that need to be addressed to cater for host and mobile code security issues. From the mobile agent owner point of view, we have trust on execution: this is the trust that the hosts will faithfully run mobile code and then migrate it to next destination (see figure 2); this trust is related to the underlying mechanisms for mobile code security, which provide preventive measures using TTPs to verify and certify the hosts' capacity for running the mobile code [4]; this trust is also related to the detection mechanisms such as signed security tags and chained hashing algorithms for mobile agent data integrity [1]. From the executing host point of view, we have the trust on mobile code; this trust is based on the ability of the creator principal to produce good code and on the honesty of the prior

sender principals for not tampering with it or making it malicious; this trust is related to the countermeasures employed in secure mobile agent systems such as the passports for mobile agent credentials and signed security tags for code integrity which can be verified by executing hosts [1], and sand-boxing for host protection [5,1].

Let us now refine these two types of trust using the *Security Enhanced Mobile Agent* (SeA) [1] as an example. We derive six security related trust questions (Q1-Q6) that should be answered at different stages of the life cycle of a typical mobile agent operation.

For a detailed description of the system operation of the SeA refer to [1, 2]. Below we specifically look at the mobile agent operation from trust relation perspective abstracting away the details of the actual operation.

Stage One – Before the deployment of an agent from the owner host: There can be two execution trust questions:

- **Q1**: Before deploying the mobile agents, the agent owner needs to ask if all the foreign hosts on the itinerary list (to be visited) are "trustworthy" in terms of faithfully executing the mobile code (an *execution trust* question).
- **Q1.1**: If yes, then the agent owner needs to determine if all the foreign hosts will have enough trust in the agent owner itself to cooperate with it (that is, a *code trust* question involving *trust symmetry* problem).
- **Q2**: The executing host will not tamper the agent or steal the agent's information, and will migrate the mobile agent to the next host. As will be discussed later this trust is difficult to obtain as it may not be possible to know the trust that the interim executing host has on the remaining hosts. That is, the interim host will need to make a decision on the trustworthiness of the host on the next hop from its point of view which may not be known to the agent owner (an *execution trust* question). Even though having the hosts on the itinerary indicates the initial trust the home base (i.e. the agent owner principal) has on the hosts on the list (including the current host), it does not guarantee that the current (an intermediary) host will consider the next host to be trustworthy. In many cases such trust relationship is often implicitly assumed.

Stage Two – Mobile agent in transit: An execution host may need to ask the mobile *code trust* question as well as the *execution trust* question.

- **Q3**: After receiving a mobile agent and before executing the code, the foreign host may need to be sure that the agent owner host is trustworthy in terms of generating proper mobile code, and every prior executing host is trustworthy in terms of execution and migration. This is thus a hybrid question involving both the *code* trust and *execution* trust.
- **Q4**: The current host needs to determine whether the next host on the itinerary is trustworthy for execution (an *execution* trust question).

Stage Three – Last host: **Q5**: The last host on the itinerary list will need to ask the trust questions as in **Q3** and **Q4**, with the difference that the entity in question is now the agent owner host (and not just the next executing

host on the list), and that the trust on the agent owner is already answered in **Q3** above.

Stage Four – After mobile agent returns to the owner host: Q6: The agent owner needs to check the integrity of the mobile agent and data. The hosts on the propagation list (visited by the agent) may also need to be examined to gain the trust on the results.

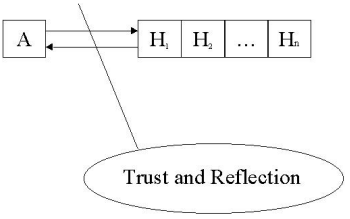

Fig. 1. *Trust* and *reflection* between the agent owner host A and the hosts $H_1 \ldots H_n$ on the itinerary list

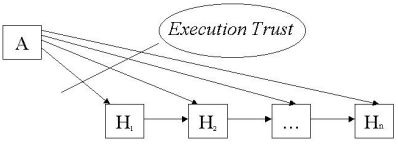

Fig. 2. *Execution trust* the agent owner host A has in the hosts $H_1 \ldots H_n$ and the intermediary execution host has on the next host on the itinerary list

2.1 Trust Symmetry – A Reflection Problem

Let us first focus on **Q1.1**, which is a different type of trust question as opposed to the normal ones such as **Q1**. In this question the agent owner asks whether all foreign hosts will have enough trust in the agent owner to cooperate with it (see Figures 1 and 3). This is a difficult question which relates to *trust symmetry* that may not be known to the agent owner. Hence it can only be answered in a non-deterministic manner, as the agent owner needs to position itself as the other hosts (on the itinerary list) to estimate their trusts on the agent owner from agent owner's point of view, that is, to evaluate the trust symmetry. This is difficult without knowing other agent's mind set (trust parameters). Since in this case the agent owner may not have all the hard facts about other hosts' trust on the agent owner. In this case a trust decision needs to be made with not

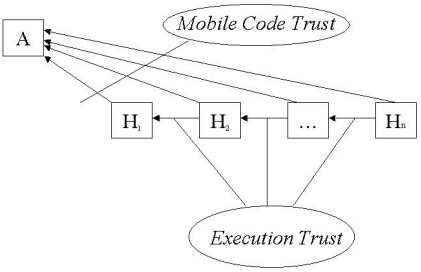

Fig. 3. *Trust reflection* - execution trust on the agent owner host A from the hosts $H_1 \ldots H_n$ and *execution trust* on the previous host from the next host on the itinerary list

only incomplete information and uncertainty in appraising agent owner's trust on other principals (executing hosts), but also taking into account the difficulties in reflecting other principals opinions on oneself (the agent owner). We call this type of trust *reflective trust* and in this paper propose a new approach to capture this in a computationally tractable way. As far as we are aware, such questions involving *trust reflections* in mobile agents have not been raised before.

3 Our Contributions

In this paper, we propose a new method to capture the reflective trust based on Marsh's trust formalism [7]. In particular, our approach provides the following:

1. We formalize the estimating procedure of other agent's trust on the trusting agent itself, thus providing a clear framework to deal with situations where knowledge among interacting agents is incomplete.
2. Our method enables agents to learn from the interactions and dynamically update trust parameters. In additional to Marsh's basic parameter set, the new parameters of trust reflection are included and can be manipulated to aid trust based security decisions.
3. We provide a new trust formalism by including trust reflection, which yields a more general framework than Marsh's. In fact, Marsh's framework can be viewed as a subset of our framework when the trust initiating agents only reason about the trust on other agents without evaluating trust reflection.
4. We apply the new framework to address the mobile agent trust problems raised in Section 2.

Before describing our new formalism and approach, it is useful to briefly review Marsh's trust formalism and notations. We will only mention those elements that are relevant for our discussion. For a full description of the model, refer to [7,8,9].

3.1 Formalizing Trust as a Computational Concept – Marsh's Approach

Marsh presented an approach to formally capture trust as a computationally tractable concept. However, his approach does not provide a clear formalism to deal with the reflection type trust questions that we have raised above. His approach is based on knowledge, utility, situation and experience. Hence this enables dynamic learning through the operating of the system by dynamically re-evaluate the situational trust.

Basic Definitions. The intuition of trust, trust values, trust threshold and their relationships with system knowledge, utility, situation and experience are captured by the following definitions [7].

Definition 1. *An agent (a...z) is an independent entity existing in a world populated with other such entities; each agent is a member of set of agents(A).*

Definition 2. *A situation is a specific point in time relative to a specific agent. Different agents at the same point in time may not consider themselves to be in identical situation.*

Definition 3. *An agent, as a trusting entity, has a basic trust 'value', derived from previous experience. This value, which is dynamically altered in the light of all experiences, is used in the formation of trusting relationships with unknown agents. It is represented as Tx, normalized over (0,1).*

Definition 4. *Trust value is a numerical representation of the fact that x trusts y, as* $T_x(y), x, y \in A$. *The value is normalized over (0,1). This is a trust in the entity only.*

Definition 5. *Situational Trust is the amount of trust in a particular agent in a given situation. It is represented as* $T_x(y, \alpha_x)$, *normalized over (0,1) for x's situational trust in or reliance on y to perform correctly in situation* (α_x). *Stating explicitly trust in entities in a specific situation adds power to the definition of trust by allowing considerations over time in different circumstances.*

Definition 6. *The importance value* $I_x(\alpha_x)$ *is the agent's estimate of how important a situation* (α_x) *is to itself and is normalized over (0,1). This notion is related to the utility of the situation* (α_x).

Definition 7. *The cost of a situation is measured in terms of the problems associated with incompetent or malevolent behaviors on the part of another agent. This is written as* $C_x(\alpha_x)$ *and is normalized over (0,1).*

Definition 8. *The benefit of a situation is the expected benefit of trustworthy behavior from agent(s) being worked with; this notion plays a large part in decisions of whether or not to cooperate in the first place. This is written as* $B_x(\alpha_x)$*, and is normalized over (0,1).*

Definition 9. *The utility of a situation is the expected utility gained by the trusting agent from the trustworthy behavior from agent(s) being worked with; In other words it represents the value and satisfaction gained from from a trustworthy transaction. This is written as* $U_x(\alpha_x)$*, and is normalized over (0,1).*

Remark 1. **Relationships between costs, benefits and agent situational importance:** Importance goes further than weighing of costs and benefits. It may also include some knowledge or assumptions about future benefits and preparation for further cooperation. As argued by Marsh that the importance adds subjective measure for the situation as opposed to the rational measures by utility.

Remark 2. **Trust, experience and situation:** Since trust is based on an agent's experience of previous interactions and situations to a large extent, and is subjective in that it depends on individuals, some method of showing whether or not an agent is known is needed. This is referred to as acquaintance in that an agent becomes acquainted with another. This has been simplified to a Boolean concept of whether an agent knows another or not. The concept of knowledge, or acquaintance is represented as $K_x(y)$.

Here is a summary for the basic trust notations used by Marsh [7]:

- Situations are represented by: $\alpha_x \ldots \omega_x$.
- Individual agents are represented by $a...z$, and are members of A, the set of all agents.
- Basic trust value for agent x: T_x.
- General trust agent x has in agent y: $T_x(y)$.
- Situational trust (Reliance) x has in y in situation α: $T_x(y, \alpha)$.
- Importance of situation α_x to agent x: $I_x(\alpha)$.
- Potential costs to agent x following untrustworthy behavior from another trust agent in situation x: $C_x(\alpha)$.
- Potential benefits to agent x following trustworthy behavior from another trust agent in situation x: $B_x(\alpha)$.
- Representation of whether agent x knows (is acquainted with) agent y: $K_x(y)$.

Remark 3. We will drop the subscript to situation in the rest of this paper, as the situations shall be mutually shared between interacting agents.

Basic Equations. Approximation of the trust theory can be captured by the following equation of situational trust [7]:

$$T_x(y, \alpha) = T_x(y) * U_x(\alpha) * I_x(\alpha) \tag{1}$$

Remark 4. The above may not always be binding as the decision to trust a specific agent may also be related to the competence of the agent in the given situation, as observed or experienced in previous situations. Thus a notion of trust threshold needs to be introduced which allows the competence to play a role in the trust decision making process. Listed below is the cooperation rule developed by Marsh [7]:

Cooperation Rule:

$$If\ T_x(y, \alpha) > Cooperation\_Threshold_x(\alpha) \Rightarrow\ Will\_Cooperate\ (x, y, \alpha) \tag{2}$$

Where:

$$Cooperation\_Threshold_x(\alpha) = \frac{Perceived\_Risk_x(\alpha)}{Perceived\_Competence_x(y, \alpha)} * I_x(\alpha) \tag{3}$$

$$Perceived\_Risk_x(\alpha) = \frac{C_x(\alpha)}{B_x(\alpha)} * I_x(\alpha) \tag{4}$$

$$Perceived\_Competence_x(y, \alpha) = T_x(y, \alpha)\ or\ T_x(y)\ or\ T_x \tag{5}$$

3.2 The New Approach

Going back to the discussion in section 2.1, we basically need to answer the question: "For any given situation, given our own utility preference on this situation (utility, cost, benefit and importance), and our trust on other agents (either in this situation or in general), what is trust the other agents have on us ?"

Obviously, based on Marsh's existing framework, to answer this question we need to have utility preference of other agents; in the worst case where we do not know this, the existing framework will not work.

Hence we need a new approach where the other agent's trust on us can be derived from our own utility preference and trust on other agents. We propose a novel approach where three different views of the trusting agent can be modelled for a given situation, namely *partnership, optimistic and pessimistic views.* In a partnership view, the other agent will treat the current situation in the same way we think about the situation, that is, they consider that in this situation their satisfaction level is the same as ours (utilities). Hence they would trust us equally like we trusted them (the trust value). Based on this intuition, we can formulate procedures using all the base parameters in the initial Marsh model to derive from our point of view other agent's trust in ourselves in a computationally tractable way. In the case of an optimistic view, we say that the other agents will treat the current situation more favorably than ourselves; that is, they would have a high utility value for the situation and higher trust for us. Finally, a pessimistic view is the opposite of the optimistic view.

3.3 Derivation of Other Agent's Trust on Ourselves – A Reflective Approach

Now we can formalize the intuitive concept developed above. First, we introduce some new syntactical notations: we add superscript to Marsh's notations to indicate owner agent's estimate of other agent's utility on the situation and trust on owner agent. Also included in the superscript is the view the owner agent holds when making such estimates. For example, to represent other agent's trust (other agents denoted as x) on the owner agent A in situation α, from A's point of view, one can write: $T_x^{V(A)}(A, \alpha)$, where: $V(A)$ is the view held by the home base when making the estimate. *V(A)* ::= *Partnership* | *Optimistic* | *Pessimistic*. Naturally the owner agent's trust on other agents (x) is represented as: $T_A^{V(A)}(x, \alpha)$. However, for simplicity we drop the superscript in this case, as it conveys no additional meaning when an agent is estimating its own view in this case.

Now we can restate our original question raised in Section 2.1, with the new notations as follows:

Let Home base (ourselves)be A; the set of agent hosts in the network be X; hosts on the mobile agent's itinerary list be $x = H_1 \dots H_n \subseteq X$; and situation be α.

We have the following assumptions:

Trust: $T_A(x, \alpha), T_A(x), T_A$.
Utility: $U_A(\alpha), I_A(\alpha), C_A(\alpha), B_A(\alpha)$.
Decision Functions: $Cooperation\_Threshold_A(\alpha), Will\_Cooperate(A, x, \alpha)$.
Decision Rule: $T_A(x, \alpha) \geq Cooperation\_Threshold_A(\alpha) \Rightarrow Will\_Cooperate(A, x, \alpha)$.

We then need to calculate the following in order to solve the trust *reflection* problem raised in Section 2.1.

Trust: $T_x^{V(A)}(A, \alpha)$, $T_x^{V(A)}(A)$, $T_x^{V(A)}$.
Utility: $U_x^{V(A)}(\alpha)$, $I_x^{V(A)}(y)$, $C_x^{V(A)}(\alpha)$, $B_x^{V(A)}(\alpha)$.
Decision Functions: $Cooperation\_Threshold_x^{V(A)}(\alpha), Will\_Cooperate^{V(A)}(x, A, \alpha)$.
Decision Rule: $T_x^{V(A)}(A, \alpha) \geq Cooperation\_Threshold_x^{V(A)}(\alpha) \Rightarrow Will\_Cooperate^{V(A)}(x, A, \alpha)$.

In order to complete the above trust computation, we need to arrive at some formulae for:

- A's point of view on x's view of utilities, cost and benefit.
- A's point of view on x's view of importance of the situation.
- A's point of view on x's trust on A.

Derivation of Reflective Trust Parameters. We present here three views that the owner agent may hold at any given situation, namely the views of *Partnership, Optimism and Pessimism.*

Case One – Partnership View. On *utility* estimation, one could intuitively argue that if A does not know the view of x (in terms of utility, i.e. cost and benefits) for a given situation, then in this partnership view, A believes that there is no advantage or disadvantage for x (neutral): $C_x^{V(A)=partnership}(\alpha) = B_x^{V(A)=partnership}(\alpha)$. On *importance* estimation in this partnership view, A believes that this situation is as important to x as it is to itself; thus $I_x^{V(A)=partnership}(\alpha) = I_A(\alpha)$. The trust value estimation is independent of the views. In terms of the trust values, we may have four categories of estimation that are applicable to all the views A may have on x's utility and the situation. 1) Since A does not know anything, the minimum case here is that A believes that x will associate the default (generic) trust on A and A use its own trust value as the estimate where a global default is not available. Thus $T_x^{V(A)} = T_A$. 2) A does not know x, but x may know A, so $T_x(A)$ may indeed be possessed by x. But A does not know this. So in answering the question (i.e. A is asking will x cooperate with A. A will still associate the trust as above. 3) In this case, A knows that x knows A and thus:$T_x^{V(A)} = T_x(A)$. 4) Finally there can be a situation where the trust value can be thought as possessed by A already: $T_x^{V(A)}(A, \alpha) = T_x(A, \alpha)$.

Case Two – Optimistic View. On *utility* estimation, in this view, A believes that x believes that the current situation will give a higher satisfaction level than it would give to A, and that the *cost* will be *lower* relative to the $benefit$. Then the following is obtained: $C_x^{V(A)=partnership}(\alpha) < B_x^{V(A)=partnership}(\alpha)$. On *importance* estimation, in this optimistic view, A believes that this situation is more important to x as it is to itself A. Thus $I_x^{V(A)=partnership}(\alpha) > I_A(\alpha)$. On *trust* value estimation, this is view independent and thus it is the same as in *Case One* above.

Case Three – Pessimistic View. On utility estimation, this view considers the case when A believes that x believes that the current situation will give a lower satisfaction level than it would give to A , and that the *cost* will be higher relative to the $benefit$. Then the following can be obtained: $C_x^{V(A)=partnership}(\alpha) < B_x^{V(A)=partnership}(\alpha)$. On *importance* estimation, in this optimistic view, A believes that this situation is less important to x as it is to A; thus $I_x^{V(A)=partnership}(\alpha) < I_A(\alpha)$. On *trust* value estimation, it is view independent and thus is the same as in *Case One* above.

3.4 Algorithm for the Trust Model Update

One needs to note that the reflexive trust is only an estimate, and can be wildly wrong [8]. The main idea here is to provide a starting point for the trustor to estimate the initial trust reflection from the trustee and as more interactions occur this value should converge to the real trust reflection. This could mean that even when the initial estimate of reflexive trust is below the cooperation threshold, the trustee may still cooperate implying that the

trustee has a more positive view than the trustor has estimated initially. Therefore, we need a procedure to update the trust values dynamically in a course of on-going interactions such that the converging trust value can be computed. Let us now consider such a trust update algorithm. We look at both successful and failed interactions. The requirements of the mobile agent trust model is to be able to evaluate the trust reflection to aid low level security decisions. More specifically we need to be able to use our trust value and utility preferences as a starting point to set up some initial estimates for trust reflection to enable interactions in the mobile agent context. We then need to be able to update the trust values based on the observed behaviors from the other agents.

Trust Update Algorithm:

1. Calculate trust on all the potential candidates for cooperation from A's point of view using A's own utility preference and trust knowledge.
2. Select the agents that have passed the cooperation threshold for both trust and trust reflection from A's point of view.
3. For all the selected but unknown agents ($K_x(y)$= false), derive their utility and trust on A from A's point of view by adopting the *partnership* view to enable first time interaction.
4. For an unknown agent, if interaction failed at first time, then it will be captured as an unfriendly/malicous agent and A stops any further interactions. However, if the first interaction is successful then the interaction will continue even when the cross over margin is negative.
5. For a known agent($K_x(y)$= true) , if the interaction is successful then it will be associated with an *optimistic* view for a future interaction. Its utility and trust data are updated as given below:
   - Utility, benefit will be incremented and cost decremented by a set amount.
   - Importance will be incremented by a set amount.
   - Trust value will be incremented by a set amount.
   - Recalculate, risk, competence, situational trust and trust threshold for future interactions.
6. If the interaction is unsuccessful then it will be associated with a *pessimistic* view for a future interaction. Its utility and trust data will be updated as below:
   - Utility and benefit will be decremented and cost incremented by a set amount.
   - Importance will be decremented by a set amount.
   - Trust value will be decremented by a set amount.
   - Recalculate risk, competence, situational trust and trust threshold for future interactions.

4 Application to Mobile Agent Security

Now we can apply the new trust model and the algorithm to mobile agent security problems that we introduced in Section 2.1.

Example 1. Assume that the following is given for the agent owner A and host x, and that A knows nothing about x initially. Here we set out to evaluate the reflexive trust the x has on A using a *partnership view*. We also assume seven successful interactions followed by one (or more) failed interaction(s). We set margin value to be 0.05; the increment delta to be 0.1 and decrement delta to be 0.2. These are intended to simulate the social norm that trust is hard to build but easy to destroy.

Table 1. Initial Parameter Values

Initial Parameters	Values
$K_A(x)$	$false$
$U_A(\alpha)$	0.45
$I_A(\alpha)$	0.60
$B_A(\alpha)$	0.70
$C_A(\alpha)$	0.50
T_A	0.4
$Risk_A(\alpha)$	0.43
$Competence_A(\alpha)$	0.4
$T_A(x, \alpha)$	0.108
$Threshold_A(\alpha)$	0.64
$Margin$	-0.53
$Will\_Cooperate(A, x, \alpha)$	Yes

Margin[1] is used to determine the threshold crossover point to avoid oscillation. Cooperation [2] is enabled for the first time interaction for an unknown host using *partnership view*. The increment/decrement amounts chosen here for various utility and trust parameters are for illustrative purpose only. Refer to [11] for a detailed study on how the amount of increment/decrement and their ratio can impact on the trust evolution. We obtain the simulated results by applying the developed trust derivation rules in Section 3 and the update algorithm in Section 3.4. See Table 2 for the results.

4.1 Remarks on the Simulation Results

The model will have an upper limit of 1 for the parameters that are normalized to 0 to 1, and a lower limit to 0.1 which is the smallest increment or decrement in the system when updating. It can be observed from the simulation that it takes seven successful interactions to bring the trust level to the upper limit for the given condition and only take 1 failed interaction to take agent to below

[1] Cross above if $T_x(A, \alpha) > Will\_Cooperate(x, A, \alpha) + margin$; Cross under if $T_x(A, \alpha) < Will\_Cooperate(x, A, \alpha)$ - $margin$

[2] See algorithm in Section 3.4

Table 2. Trust Evolution in Light of Experience, Partnership = +, Pessimistic = -

Parameters	Init State	1st	2nd	3rd	4th	5th	6th	7th	1st(failure)
$U_x(\alpha)$	0.45	0.55	0.65	0.75	0.85	0.95	1	1	0.65
$I_x(\alpha)$	0.60	0.7	0.8	0.9	1	1	1	1	0.8
$B_x(\alpha)$	0.70	0.8	0.9	1	1	1	1	1	0.65
$C_x(\alpha)$	0.50	0.4	0.3	0.2	0.1	0.1	0.1	0.1	0.8
T_x	0.4	0.5	0.45	0.55	0.65	0.75	0.85	1	0.8
$Risk_x(\alpha)$	0.43	0.35	0.27	0.18	0.1	0.1	0.1	0.1	0.3
$Competence_x(\alpha)$	0.4	0.5	0.45	0.55	0.65	0.75	0.85	1	0.45
$T_x(A, \alpha)$	0.108	0.135	0.234	0.37	0.55	0.71	0.85	1	0.234
$Threshold_x(\alpha)$	0.64	0.47	0.29	0.29	0.15	0.13	0.11	0.1	0.53
$Margin$	-0.53	-0.51	-0.24	0.07	0.39	0.58	0.73	0.9	-0.29
$Will\_Cooperate(x, A, \alpha)$	Yes	Yes	Yes	**Yes**	Yes	Yes	Yes	Yes	**No**
$Views$	+	+	+	+	+	+	+	+	**-**

the cooperation level, thus enable the trust model to reflect closely the norms of trust in general society.

Cooperation crossover margins also follow a similar trend. If required it can play a role in determining the optimum partnerships among a set of selected agents by using the margin as a safety device. However, this aspect is not considered in this example as we mainly focus on the reflection trust relationships the agent owner has with a host or hosts.

5 Concluding Remarks

In this paper, we have considered a trust framework for mobile agent based systems. We have discussed some fundamental trust questions that arise in the various stages of a mobile agent based system operation. Within this context, we have considered trust properties such as execution trust and mobile code trust. Then we introduced the problem of trust reflection in mobile agents and proposed a new formalism to capture and reason about trust reflection in the decision making process. Our framework captured the intuition of reasoning about other agents' view on the owner agent from owner agent's point of view. We have presented three possible views and the utility and trust update algorithm. Finally we applied the new formalism and the developed algorithm to the mobile agent trust questions raised earlier and provided an analysis of the simulated results. We believe that the work presented in this paper provides a starting point for setting up a general framework for reasoning about trust in mobile agent based systems.

Currently we are refining the trust update algorithm to enable the usage of agent authenticity information in addition to the observed evidence for trust [3], thus allow different trust update dynamics for authenticated and non-authenticated agents. We are also implementing the proposed trust model in a secure mobile agent system prototype [2], which would be helpful to compare the real performance results with the simulated ones.

References

1. Varadharajan,V., Security Enhanced Mobile Agents, Proc. of 7th ACM Conference on Computer and Communication Security, 2000.
2. Varadharajan, V. and Foster, D., A Secure Architecture and Demonstration of a Secure Mobile Agent Based Application. In the Proceedings of IASTED International Conference on Networks, Parallel and Distributed Processing and Applications 2002.
3. Lin, C. and Varadharajan,V., On the Design of a New Trust Model for Mobile Agent Security, submitted for publication, 2003.
4. Tan, H. K., and Moreau, L., Trust Relationships in a Mobile Agent System," presented at Fifth IEEE International Conference on Mobile Agents, Atlanta, Georgia, December, 2001.
5. Gong, L., Java Security Architecture (JDK1.2), Draft document, Revision 0.5," available from Java Developer Connection, http://java.sun.com.
6. Chess, D. M., Security Issues in Mobile Code Systems, Mobile Agents and Security, Editor Vigna, LNCS 1419, Springer-Verlag, 1998.
7. Marsh, S., Formalising Trust as a Computational Concept, PhD thesis, University of Stirling, 1994.
8. Marsh, S., Trust in DAI, In Castelfranchi and Werner (ed), Artificial Social Systems, Springer Lecture Notes in Artificial Intellegence, vol. 830, 1994.
9. Marsh, S., Optimism and Pessimism in Trust, Technical Report CSM-117, Department of Computer Science, University of Stirling, 1994.
10. in, C. and Varadharajan, V., On Design of a New Trust Model for Mobile Agent Security, submitted for publication.
11. Yu, B. and Singh, M.P., A Social Mechanism for Reputation Management in Electronic Commerce, In Proceedings of the 4th International Workshop on Cooperative Information Agents, pp. 154–165, 2000.
12. Abdul-Rahman, A. and Hailes, S. A Distributed Trust Model. In Proceedings of the 1997 New Security Paradigms Workshop, pages 48–60. ACM, 1997.
13. Abdul-Rahman, A. and Hailes, S., Supporting Trust in Virtual Communities. In Proceedings of the Hawaii International Conference on System Sciences 33,Maui, Hawaii.
14. Bierman, E. and Cloete, E., Classification of Malicious Host Threats in Mobile Agent Computing. In the proceedings of SAICSIT 2002.
15. Farmer, W.M., Guttman, J.D. and Swarup, V., Security for Mobile Agents: Authentication and State Appraisal. in In Proceedings of the 4th European Symposium on Research in Computer Security, (Berlin, 1996), Springer Verlag, pp 118–130.
16. Gray, R.S., A Flexible and Secure Mobile Agent System, 4th Annual Tcl/Tk Workshop Proc.
17. Jansen, W., Countermeasures for Mobile Agent Security, Comupter Communications, Special Issue on Advances of Network Security.
18. Karnik, N. and Tripathi, A., Security in Ajanta Mobile System. Software Practice and Experience, John Wiley and Sons.
19. Necula, G. Proof-Carrying Code. Proceedings 24th Annual Symposium on Principles of Programming Languages (Paris, France, Jan. 1997),ACM, New York. 106–119.
20. Oppliger, R. Security Issues Related to Mobile Code and Agent-Based Systems. Computer Communications, 22 (12). pp. 1165–1170, July 1999.

21. Sander, T. and Tschudin, C., Towards Mobile Cryptography. in Proc. of the IEEE Symposium on Security and Privacy, USA, (1998), IEEE Computer Society.
22. Tan, H.K. and Moreau, L., Trust Relationships in a Mobile Agent System. in Fifth IEEE International Conference on Mobile Agents, (Atlanta, Georgia, December, 2001), Springer-Verlag.
23. Vigna, G., Protecting Mobile Agents through Tracing, LINCS 1419, 1998.
24. Wilhelm, U.G., Staamann, S. and Buttyán, L., Introducing Trusted Third Parties to the Mobile Agent Paradigm. in Secure Internet Programming: Security Issues for Mobile and Distributed Objects, LNCS 1603, (1999), Springer-Verlag.