The Chinese University of Hong Kong
Department of Computer Science and Engineering

Ph.D. – Term Paper

Title:	Trust Model Based Self-Organized Routing Protocol for Secure Ad Hoc Networks
Name:	Li Xiaoqi
Student I.D.:	02053280
Contact Tel. No.:	26098344
Email A/C:	xqli@cse.cuhk.edu.hk
Supervisor:	Prof. LYU Rung Tsong Michael
Markers:	Prof. LEUNG Kwong Sak and Prof. LEE Kin Hong
Mode of Study:	Full-time
Submission Date:	April 22, 2003
Term:	2
Fields:	
Presentation Date:	April 29, 2003
Time:	9:00am-10:15am
Venue:	Rm. 1027, Ho Sin-Hang Engineering Building

Trust Model Based Self-Organized Routing Protocol for Secure Ad Hoc Networks

Abstract

An ad hoc network is a kind of wireless network without centralized administration or fixed network infrastructure, in which nodes perform routing discovery and routing maintenance in a self-organized way. However, this flexible network topology introduces a lot of security problems and the existing routing protocols for ad hoc networks such as AODV [32], DSR [17], DSDV [33] and so on have no effective measures to prevent themselves from being attacked. Some researchers have proposed several secure routing protocols to protect the ad hoc networks such as SAODV, Ariadne, SEAD and so on. But most of these secure routing protocols need some centralized units or some trusted third-parties to issue digital certificates or monitor network traffics, which actually destroy the self-organization nature of ad hoc networks. In this paper, we introduce the idea of trust model into ad hoc networks and propose a self-organized secure routing protocol based on our trust model. Each node in this ad hoc network has its opinions about some other nodes' trustworthiness, which are obtained by directly communicating with other nodes or by combining other nodes' recommendations. Then the node will decide whether to exchange routing information with another according to its opinion about that nodes' trustworthiness.

Keywords: Ad Hoc Networks, Trust model, Secure Routing Protocol, Self-organized, Wireless security

Contents

1 Introduction

One of the most important characteristics of Ad Hoc networks is that they require no centralized administration or fixed network infrastructure [15]. Nodes in the ad hoc networks cooperate with each other to do routing discoveries and data transmissions in a self-organized way. Self-organization is the nature of ad hoc networks. It will beat others when used in the applications without existing infrastructure or any other trusted authorities. However, it is this self-organization nature that makes mobile ad hoc networks insecure. Without any third-party authority, everybody including some malicious nodes can join and leave the ad hoc network freely. These malicious nodes would then perform all kinds of attacks to eavesdrop information, interrupt normal communications, or even make the whole network denial-of-service. The trust relationship between the existing nodes will be disturbed, and no one can definitely trust with each other any more.

Many security schemes for mobile ad hoc networks have been proposed in order to protect the routing information or data packages during communications. However, most of these schemes assume that there are trusted third parties or centralized servers who are responsible for issuing cocksure certificates and keys or monitoring the behaviors of other nodes. Centralized servers or trusted parties make the network more controllable but they destroy the self-organizing nature and reduce the scalability of mobile ad hoc networks. Even some schemes distribute the servers into many nodes, there are still bottlenecks due to centralization. If one scheme distributes the functions of servers into each node of the network, it will introduce huge performance overhead. Thus, we need a self-organized security scheme for mobile ad hoc networks.

This paper focuses on designing a secure routing protocol for ad hoc networks in a self-organizing way instead of using centralized servers. There are also some self-organized secure schemes for ad hoc networks [40]. They usually make the unrealistic approach to let each node monitor its neighbors. This will bring high overhead and even worse security [4]. Our solution, on the other hand, introduces the idea of "Trust Model" to try to solve this problem. One node, n_1 can judge whether another node, n_2 could be trusted according to the trust opinion that n_1 obtains about n_2, thus n_1

can decide whether to go on communicating with n_2 or require n_2 to prove itself by some other ways. n_1 can also obtain a more credible trust value of n_2 by exchanging values with other nodes and calculating the new trust level value of n_2 using a certain algorithm. In this way, we can achieve a real self-organized trust model between all the nodes. Moreover, some authentication measures, such as digital signature, can be performed in a more flexible way based on trust level and the system overhead can be greatly reduced.

In our design, each node maintains a trust table mainly containing the trust opinion its neighbors or nodes who ever communicated with it. Opinion is a 3-dimensional metric which contains one node's belief, disbelief and uncertainty value about another node. These values can be any value in the interval $[0, 1]$. In our research, we propose to divide it into five degrees logically. If one node performs normal communications to some extent, its opinion in other nodes' trust table will be increased. On the contrary, if one node has some malicious behaviors, it will be denied by the network at once or be isolated from the network ultimately in terms of the compromise it has brought.

Our secure routing protocol is based on AODV routing protocol [32]. AODV is an on-demand routing protocol which is quite suitable for our concept of trust level. However, the idea of trust level can also be used in other ad hoc routing protocols, such as DSR [17], DSDV [33] and so on.

The background about AODV and the mathematical theory used by trust level calculation algorithms are described in Section 2. Another contribution of this section is that we summarize almost all the attacks to the ad hoc networks in a general way. Section 3 is the survey of trust theory and some existing trust models. We present our design of self-organized secure routing protocol based on trust model in section 4. In section 5, we give our research plan and future work. Finally, section 6 concludes our work and proposes a future direction.

2 Background

2.1 Overview of AODV

AODV (Ad hoc On-demand Distance Vector) routing protocol is one of the most popular routing protocols for mobile ad hoc networks. One distinguishing feature of AODV is its use of a destination sequence number for each route entry [32], which ensure loop freedom at all times. There are three main routing messages used in AODV, which are RREQ (Routing REQuest message), RREP (Routing REPly message) and RERR (Routing ERRor message). Routing discovery and routing maintenance are two basic operations in AODV protocol.

Routing discovery happens when a node wants to communicate with a destination while it has no proper route entry for that destination. In this situation, this source node (originator) will broadcast a RREQ message to all its neighbors. Each neighbor who receives this RREQ will check in its own routing table if it has the route entry for that destination. If not, it will set up a reverse path towards the originator of RREQ and rebroadcast this routing request. Any node who received this RREQ will generate a RREP message if it either has a fresh enough route to satisfy the request or is itself the destination. Then this intermediate or destination node will generate a RREP message and unicast to the next hop toward the originator of the RREQ, as indicated by the route entry for that originator. When a node receives a RREP message, it first updates some fields of the routing table and the routing reply, and then forwards it to the next hop towards the originator. In this way, this RREP will ultimately reach the source node and a bidirectional route path has been established between the source and destination. Thus, these two ends can communicate with each other using the just set up routing path.

Routing maintenance is performed through two ways. One is that a node may positively offers connectivity information by broadcasting hello messages locally so that its neighbors can determine connectivity by listening for hello packets. The other way is that a node can maintain local connectivity to its next hops using some link or network layer mechanisms, such as the detection mechanism of IEEE802.11 MAC

(Media Access Control) protocol.

RERR messages will be issued when a node detects a link break for the next hop of an active route or this link break cannot be local repaired. On receiving the routing error message, a node will notify some related nodes that certain nodes are unreachable using a modified RERR message.

When a node firstly joins this ad hoc network, it will broadcast hello messages to claim its existence. And if a node leaves this network, the route entry for it will be eventually removed from the routing tables of all the nodes after a certain period of timeout.

2.2 Attacks to Ad Hoc Networks

To achieve a secure network environment, a number of security attributes are provided in the services: Confidentiality, Authenticity, Integrity, Availability, Non-repudiation and Access control [22]. They are described as follows:

- Confidentiality: Stored or transmitted information is accessible only by authorized parties.
- Authenticity: Identity of the origin of the message is correctly identified.
- Integrity: Only authorized parties can modify stored or transmitted information and system assets.
- Availability: Network resources are available to authorized parties.
- Non-repudiation: Sender or receiver cannot deny the transmission.
- Access Control: Information resources are controlled.

Generally, attackers can launch all kinds of attacks to destroy one or more of the above secure services. We will give our taxonomy about attacks to ad hoc networks in terms of attack methods. Attackers may use one or more attack methods to achieve their different goals. These attack methods are eavesdropping, masquerading, modification, tunneling, flooding, and package-oriented attack method including dropping, replaying and delaying.

Eavesdropping Ad hoc network is a kind of wireless network. The property of wireless connections results in the possibility of being eavesdropped. Attacker can analyze the payloads of the packets and obtain the content information. This attack method will destroy the confidentiality of information. Usually, encryption is needed to prevent against eavesdropping.

Masquerading It means that attackers will forge some artificial packages in order to impersonate good nodes. This attack will badly affect the authenticity of information. Almost all the traditional routing protocols for ad hoc networks do not perform routing information verification and they trust each routing request or reply by default. Let's take AODV routing protocol for example. In AODV there are three main routing messages: Routing request, Routing reply and Routing Error, as described in Section 2.1. For routing request messages, attackers can impersonate a source node by forging a routing request message with his address as the originator address. Because the destination sequence number of a node can be set in AODV, the originator of a routing request message can put a much bigger destination sequence number than the real one so as to improve the living chance of this routing request. For routing reply messages in AODV, furthermore, attackers can then forge a reply message to a node to claim a faked shorter path. An attacker can form routing loops by sending faked shorter path reply messages to several nodes in the neighborhood of a node N_i one by one. Finally by forging routing error messages, the attacker can lie to others and convince them that node N_i is unreachable.

Modification This attack often happens when a node forwards routing messages. A malicious node will intercept messages and alter their contents before passing them on to the intended recipient. "Man-In-Middle" [25] attack belongs to this category. Integrity of information is tampered by this kind of attack. Let's also take AODV routing protocol for example. When forwarding a routing request message, a malicious node can reduce the *hop count* field in the message to increase the chances of being in the route path. The malicious node can also increase the destination sequence number of the incoming message in order to update the other intermediate nodes' routing table.

Tunnelling Tunnelling attack needs the cooperation of two or more malicious nodes. One of them may encapsulate the routing messages and exchange the encapsulated messages with the other malicious node through the normal route path of the network. Then the two nodes will decapsulate the messages and forward them out. Thus, they can establish a ”virtual”, ”direct” path between them. By the tunnelling attack, attackers can confuse the good nodes that there is a shorter path going though the malicious nodes. This attack will influence the availability of certain network resources. It is difficult to prevent and detect the tunnelling attack. Nearly none of the existing secure routing protocol can solve this difficult problem.

Flooding Attackers may launch a great lot of messages in a short time to a node, a channel or some other network resources to make them too ”crowed” to accept any more requests. This is a kind of Denial-Of-Service (DOS) attack.

Packet Oriented Attacks This category includes many attack methods baring the similar properties. That is, these attacks only focus on the quantity or transmitting time of packages, including both routing and data packages. Dropping, replaying, and delaying packets all belong to this category. Dropping violates the non-reputation property in the secure services. In AODV routing protocol, a malicious node will not forward certain routing requests, routing replies and even data messages. This kind of attacks usually cannot be correctly detected because the transmission errors have the same effect [42]. Replaying means storing intercepted messages and sending them later again. Delaying is just sending messages in a later time. These attacks can take effect without prior knowledge of the authentication policy or decrypting the messages.

We can conclude all the purposes and results of these attack methods in Table 2.2.

From the analysis in Section 2.2, we can see that it is not enough for a routing protocol to prevent attacks by only using some authentication and cryptography methods. Consequently, we shall present a trust model for existing secure routing protocol to adopt in order to prevent more attacks and improve the robustness of secure ad hoc networks.

Attack Method	Motive/Result	Influence to Security Services
Eavesdropping	Obtain contents of messages	Loss of Confidentiality
Masquerading	Impersonate good nodes Routing Redirection Routing table poisoning Routing Loop, etc.	Loss of Authenticity
Modification	Make a node denial of service Obtain keys, etc.	Loss of Integrity
Tunnelling	Attract traffic Routing Redirection	Loss of Confidentiality and Availability
Flooding	Denial of Service	Loss of Availability
Dropping	Destroy normal routing progress	Loss of Non-reputation and Availability
Replaying/Delaying	Destroy normal routing progress Destroy normal data transmission	Loss of Access Control and Integrity

Table 1. Attack methods and their results to ad hoc networks

3 Trust Theory

3.1 Definition of trust

All kinds of transactions, interactions, and communications in human life are based on one fundamental aspect: trust. People often take trust into account when they do everything, even if they are not aware of it. For example, an employer may give a job to a 'stranger' after a short time interview and that 'stranger' will come to work on time on the second day. They both take risk and must have basic trust between each other. So do the computer networks nowadays. Ad hoc networks may contain many peer nodes. Each node is a 'stranger' to another. These nodes also need trust before they exchange information. Before we answer "How do they trust each other?" Let's first look at the question: "What is trust?"

There are different definitions about trust in different fields and aspects, such as social psychology, sociology, and philosophy [29] [24].

In the Oxford English Dictionary, the word "trust" is defined as follows [24]:

n. 'confidence, strong belief, in the goodness, strength, reliability of something or somebody', 'responsibility'.

v. have trust in - 'believe in the honesty and reliability of someone or something', 'have confidence in', 'earnestly hope'

3.1.1 Trust in Psychology

In the category of psychology, one popular definition about trust is given by Morton Deutsch in [8], which is [29]:

1. *If an individual is confronted with an ambiguous path, a path that can lead to an event perceived to be beneficial (Va+) or to an event perceived to be harmful (Va-)*;
2. *He perceives that the occurrence of Va+ or Va- is contingent on the behavior of another person; and*

3. *He perceives that strength of Va- to be greater than that strength of Va+.*

   If he chooses to take an ambiguous path with such properties, I shall say he makes a trusting choice; if he chooses not to take the path, he makes a distrustful choice.

Deutsch defined trust as a kind of subjective behavior. Whether an individual will take the path or not is from his own point of view. And he also needs 'another person' to update his perceptions. Different individuals will have different viewpoints on the same thing. So 'the estimate of costs ($Va-$) and benefits ($Va+$) will be different' [29]. Then cooperation is needed to judge whether an ambiguous path is beneficial or harmful. But "one should spend hours analyzing the costs and benefits of each situation in order to derive the maximum benefit from it. However, time is valuable too, and clearly the sensible approach to this problem of processing limits is to develop a scheme in which extensive intellectual work is only done under certain circumstances" [29] [13] [11].

This definition was updated by Deutsch later in his 1972 book: 'The Resolution of Conflict' [9]. He "expands the definition further, and presents clarifications, eventually arriving at the definition of trust as confidence, which is confidence that one will find what is desired from another, rather than what is feared" [29]. This definition describes such a process to make a trusting choice: First, one may feel both desired and feared to the ambiguous path. Then because of the existence of fear, one must take a risk before he has confidence towards the beneficial outcome. Trust eventually becomes the confidence on the beneficial path ($Va+$).

3.1.2 Trust in Sociology

The Reduction of Complexity

Niklas Luhmann's approach to trust is sociological. His main idea is that 'trust is a means for reducing the complexity of society' [29]. With more and more relations and interactions among human life the complexity of our everyday world is increasing faster and faster. Luhmann suggests that 'in condition of increasing social complexity

man can and must develop more effective ways of reducing complexity' [26]. "What this means is that every time we face a complex or even a simple decision-making situation, we have to make some assumptions taking into account the particular situation and the particular environment and then make some trusting choice" [24].

Bernard Barber's Definition

Barber is a socialist and his work is also inherently sociological which was published in "Logic and Limits of Trust" [2]. He viewed trust as an aspect of all social relationships and presents some fundamental meanings of trust as follows:

1. *Expectation of the persistence and fulfillment of the natural and moral social orders.*

2. *Expectation of "technically competent role performance" from those we interact with in social relationships and systems*

3. *Expectation that partners in interaction will "carry out their fiduciary obligations and responsibilities, that is, their duties in certain situations to place others' interests before their own."*

For 1, it means that "in a general sense, trust is an expectation that the natural, physical and biological order will continue to hold true". For 2, it refers that, for example, "we trust our doctors to perform operations well, or we trust those we elect to govern the country in a sensible and efficient manner". For 3, "that is, for those members of society who have moral obligation and responsibilities, we expect that this will be done" [29].

This idea emphasis an inherent social order based on trust. People in this society can not know everybody very much, thus they must make some assumptions that another entity will not use his power against them and they must trust each other [29].

3.1.3 Trust in terms of Mathematics

Diego Gambetta gave the definition about trust in terms of mathematics in the article "Can We Trust Trust?" [10] which is collected in "Trust: Making and Breaking Coop-

erative Relations" [11]. He defined trust as a probability, whose value is in the range 0 to 1 [29]. His definition is [10]:

> *Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently or his capacity ever to be able to monitor it) and in a context in which it affects his own action.*
>
> *When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him.*
>
> *Correspondingly when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain from doing so.*

This definition gives us a different viewpoint of trust. The introduction of probability provides us a mathematical model to measure trust. Trust now "becomes more concrete than abstract compared to other definitions presented earlier" [24]. "The use of values does allow us to talk succinctly and precisely about specific circumstances in trusting behavior. In addition, it allows the straightforward implementation of formalism" [29]. Diego said that "trust is better seen as a threshold point, located on a probabilistic distribution of more general expectations, which can take a number of values suspended between complete distrust (0) and complete trust (1), and which is centered around a mid-point (0.50) of uncertainty" [10].

This definition recognizes that trust is relevant only when there is a possibility of distrust, betrayal, exit or defection [24]. That is, when someone is trusted there is a chance that the action he performs may be non-beneficial to us [24].

With this definition we will have a theoretical basis to establish our own trust model.

3.2 Properties of Trust Relationships

Trust relationships inherit a list of characteristics and different forms from different angles. In this section we will describe these basic properties and forms.

3.2.1 Relativity of Trust Relationships

Trust relationships are not absolute. That is, two entities will keep a trust relationship only in a certain category or class. One ”trusts a trustee with respect to its ability to perform a specific action or provide a specific service within a context” [37]. For example, we trust our dentists when we have toothaches, but we'd better not trust them when we sprain our ankles. Also because of the large amount of trust information, we can only give certain trust to some specific information. So the trust relationship contains certain relativity.

Because of the relativity of trust relationships, many trust models use trust categories to represent which aspect of trust they are referring to [1].

3.2.2 Arity of a Trust Relationship

A trust relationship can be one-to-one between two entities. It can also be one-to-many and many-to-one relationships such as the relationship between one employer and his many employees. It can also be many-to-many such as the mutual trust between members of a group or a committee. ”In general, the entities involved in a trust relationship will be distributed and may have no direct knowledge of each other so there is a need for mechanisms to support the establishment of trust relationships between distributed entities” [14].

If one takes the view that a set is a collection of one or more entities, then a trust relation can be generalized as a relation between two sets: the trustor set and the trustee set. Thus, a trust relationship can be viewed as a binary relation, since it occurs between trustors and trustees [14].

3.2.3 Asymmetry

In general, trust relationship is not symmetric. One can trust another but not vice versa. That is the trustworthiness in the reverse direction need not exist at the same time. Thus, the trust relationship can be viewed as a one-way or unidirectional relationship.

If mutual trust exists between the same entities, some trust models such as [1] often represent them as two separate trust relationships. This allows each of these relationships to be manipulated independently.

3.2.4 Transitivity

The literature [34] has mentioned that trust relationships should not be transitive as many suggestions said; however, some trust scenarios do exhibit transitivity. The concept of trust delegation is a prime example of the application of trust transitivity. When I delegate my trust decisions to another person, for example, John, I authorize John to make trust decisions on my behalf. Thus, when I delegate to John and John trusts an unknown entity (say Tim), John is essentially stating that I trust Tim. According to Christianson and Harbison in [6], the concept of transitivity should be avoided, as it can result in entity B adding trust assertions to an entity A's trust base without A's explicit consent leading to unintentional transitivity.

In [37], they agree that transitivity of trust may have unexpected and adverse results but this may be necessary in some situations. So they viewed transitivity as inherent in some relationships and should be considered in the analysis of trust systems in order to determine which undesired side effects should be prevented.

In the trust model of [1], Alfarez defined the trust relationship as a conditionally transitive relationship.

3.2.5 Measurability of a Trust Relationship

One's belief is a measurable concept. To offer this capability, a trust level is often associated with a trust relationship [31]. The trust level is a measure of one's belief

in another entity and thus by our definition, it is a measure of one's belief in the honesty, competence and dependability of this entity. It is not a measure, however, of the *actual* competence, honesty, security or dependability of a trustee). Some entities may be trusted more than others with respect to performing an action. It is not clear whether this level should be discrete or continuous. If discrete values are used, then a qualitative label such as high, medium or low may be sufficient. Some systems support arithmetic operations on trust recommendations so numeric quantification is more appropriate. It is also possible to provide a mapping from qualitative to numeric labels. However, there is still a problem relating to representation of ignorance (or the unknown) with respect to trust. For specification, one cannot state the unknown by definition. Specification is done with respect to known facts and desired behavior in the face of uncertainty and ignorance. Thus, a specification mechanism needs not worry about specifying the unknown, just about the required behavior when presented with it. However, but trust management, there is a need to search for experience information, which is out of the scope of one" realm of knowledge, to help one make a decision" [14] [37].

Jϕsang's Opinion Model, based on subjective logic, may be a suitable technique for assigning trust values in the face of uncertainty [19], [20], [21]. An opinion is a representation of a belief and is modelled as a triplet, consisting of b (a measure of one's belief), d (a measure of one's disbelief) and u (a measure of ignorance), such that $b + d + u = 1$. It is assumed that b, d and u are continuous and between 0 and 1 (inclusive). This model's strength lies in the ability to reason about the opinions (on a mathematically sound basis) and its consensus, recommendation and ordering operators [18]. However, its major weakness is that it cannot be guaranteed that users will accurately assign the values appropriately to them [37].

3.3 Different Forms of Trust Relationship

Although the previous section summarizes the different definitions of trust, other people have treated trust differently. Sometimes they have defined trust from a different perspective and sometimes they have linked trust with something else such as cooperation and commodity. This section highlights some of the extensions of trust and how

trust is related to some other things like cooperation, commodity etc [24].

3.3.1 Trust and Cooperation

Gambetta [10] relates trust with cooperation in the sense that cooperation makes some demand on the level of trust. If trust is only unilateral then cooperation cannot succeed. Similarly, if there is complete distrust between the agents involved then there cannot be any cooperation between them. A higher level of trust generally leads to a higher likelihood of cooperation. It can be argued that blind trust can make cooperation work since there is no possibility of distrust; however, the important thing to note in case of blind trust is that there can be an incentive to deception.

3.3.2 Trust and Recommendation

Recommendation plays a significant role in trust. In any decent-sized society it is impossible for everyone to know and to trust everyone else. In a situation where we do not know whether to trust someone or not, we tend to ask a third person, who we know and trust, if the person under consideration is trustworthy or not. Based on the third person's recommendation we may decide whether or not to trust this person under consideration. We normally consider how much we can first trust this third person and how much this third person trusts the person under consideration. If the third person does not know him then he can get a recommendation from another person who knows him and so on. Generally, the longer the recommendation chain becomes, the more difficult it is to trust the person under consideration. The person is still trustable, it is just that the value of trust can be low. There is no magical formula here, it is simply the way we perceive trust.

3.3.3 Trust as Commodity

Patha Dasgupta in [7] gives another view of trust. He believes that although trust does not have any units in which it can be measured, one can still measure its value and its

worthwhileness. It is similar to commodities such as knowledge or information. Dasgupta's view of trust resembles some of the definitions described earlier. Furthermore, trust, in some senses, can be a way of dealing with the freedom of others [29]. Later in the article he concludes that trust is based on reputation and that reputation has ultimately to be acquired through behaviour over time in well-understood circumstances.

3.4 Trust Models

Several trust models have been published. In this section, we survey them one by one.

3.4.1 Trust Model Using Direct and Recommendation Trust

In [3], the authors present a method for the valuation of trust. They indicated that the semantics of direct trust values is different from that of recommendation trust values.

This trust model was derived originally from the work of Yahalom, Klein and Beth in [38]. When doing authentication in open networks an entity often requires other entities' recommendations. These entities can be viewed as Authentication Servers (AS). To prevent contradicting or malicious recommendations from different authentication servers, it is necessary to provide a means of estimating the trustworthiness of AS. This trust model is to solve this problem. It introduces a formal way to represent trust relationships using trust values and shows how to derive and combine trust values from existing ones.

There are two types of trust in this model: direct trust and recommendation trust. Direct trust means that an entity can trust another entity directly using all existing experiences it has about that entity. Recommendation trust expresses "the belief in the capability of an entity to decide whether another entity is reliable in the given trust class and in its honesty when recommending third entities" [3].

Direct Trust

Direct trust is defined as follows:

$$P \ trusts_x^{seq} \ Q \ value \ V \tag{1}$$

A direct trust relationship exists if all experiences with Q regarding to trust class x, which P knows about, are positive experiences. Seq is the sequence of entities that mediated the experiences (recommendation path) excluding P and Q. V is the value of a trust relationship, which is an estimation of the probability that Q behaves well when being trusted. This is based on the number of positive experiences with Q, which P knows about.

Let p be the number of positive experiences with Q which P knows about with regard to the trust class x. Then the value v_z of these experiences is computed as follows:

$$v_z(p) = 1 - \alpha^p \tag{2}$$

This value is the probability that Q has a reliability of more than α, founded on the information P possesses about Q. The reliability is the probability that Q turns out to be reliable when being entrusted with a single task. α should be chosen reasonably high to ensure sufficiently safe estimations.

Recommendation Trust

The authors of [3] defines recommendation trust like this:

$$P\ trusts.rec_x^{seq}\ Q\ when.path\ S_p\ when.target\ S_t\ value\ V \tag{3}$$

A recommendation trust relationship exists if P is willing to accept reports from Q about experiences with third parties with respect to trust class x. This trust is restricted to experiences with entities in St (the target constraint set) mediated by entities in Sp (the path constraint set). V is the value of the trust relationship. It represents the portion of offered experiences that P is willing to accept from Q and is based on the experiences P with the entities recommended by Q.

If p and n represent positive and negative experiences respectively with the recommended entities, the recommendation trust value Vr is computed according to the following formula:

$$Vr(p,n) = \begin{cases} 1-\alpha^{p-n} & : \quad p > n \\ 0 & : \quad else \end{cases} \tag{4}$$

This value can be regarded as a degree of similarity between P and Q, taking into account that different entities may have different experiences with a third party.

A recommending entity may not behave well all the time, so it is sufficient to state certain dissimilarity and to lower the trust value. This is modelled by the following properties of Equation 5:

$$\begin{aligned} &-v_r(p,n) = 0 \quad \text{for} \quad p = 0. \\ &-v_r(p,n) \quad \text{approaches 1 with growing } p \text{ and fixed } n. \\ &-v_r(p,n) \quad \text{approaches 0 with growing } n \text{ and fixed } p. \end{aligned} \tag{5}$$

If the negative experiences outnumber the positive experiences, the value becomes zero and the entity is excluded from the recommendation constraint set.

Deriving Trust Relationships

We present an example showing how new trust is established when a recommendation is performed. With the help of some defined rules, a new trust relationship can be derived from a given set of initial relationships. Figure 1 depicts the derivation of trust relationships.

Consider the trust relationship shown in the left side of Figure 1 , where V_1 and V_3 represent recommendation trust and V_2 represents direct trust.

Based on these existing trust relationships, new trust relationships between A and C as well as A and D can be derived. These derived trusts can be represented by Equation 6:

Derived direct trust between A and C which is denoted by $V_1 \odot V_2$ is:

$$V_1 \odot V_2 = 1 - (1 - V_2)^{V_1} = 1 - (1 - (1 - \alpha^p))^{V_1} = 1 - \alpha^{V_1 \cdot p} \tag{6}$$

p in Equation6 is the number of positive experiences B has about C.

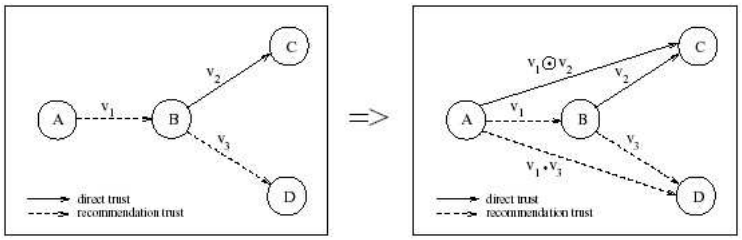


Figure 1. Derivation of trust relationships

Derived recommendation trust between A and D which is denoted by $V_1 \bullet V_3$ is:

$$V_1 \bullet V_3 = \text{simply multiplication between } V_1 \text{ and } V_3 \tag{7}$$

This multiplication shows that the value of derived recommendation trust decreases as the recommendation path grows.

The rules of inference used in the above example are defined also in [3]:

RULE1: New Direct Trust

$$\begin{aligned} & P\ trusts.rec_x^{seq_1}\ Q\ when.path\ S_p\ when.target\ S_t\ value\ V_1 \\ \wedge\ & Q\ trusts.rec_x^{seq_2}\ R\ value\ V_2 \\ \wedge\ & R \in S_t \\ \wedge\ & \forall X : (X \in seq_2 \Rightarrow (X \in S_p \wedge X \notin P \circ seq_1)) \\ \Rightarrow\ & P\ trusts_x^{seq_1 \circ Q \circ seq_2}\ R\, value\ (V_1 \odot V_2) \end{aligned} \tag{8}$$

RULE2: New Recommendation Trust

$$\begin{aligned} & P\ trusts.rec_x^{seq_1}\ Q\ when.path\ S_{p_1}\ when.target\ S_{t_1}\ value\ V_1 \\ \wedge\ & Q\ trusts.rec_x^{seq_2}\ R\ when.path\ S_{p_2}\ when.target\ S_{t_2}\ value\ V_2 \\ \wedge\ & \forall X : (X \in seq_2 \Rightarrow (X \in S_{p_1} \wedge X \notin P \circ seq_1)) \\ \Rightarrow\ & P\ trusts_x^{seq_1 \circ Q \circ seq_2}\ R \\ & when.path\ (S_{p_1} \cap S_{p_2})\ when.target(S_{t_1} \cap S_{t_2})\ value\ (V_1 \bullet V_2)) \end{aligned} \tag{9}$$

Trust derivation algorithms are required to track down all entities which can be trusted by an entity P with respect to a trust class x. One of the trust derivation algorithms

is proposed in [38] which tries all the recommendation trust expressions to derive as many new trust expressions as possible. Then remove from or insert in this set the considered recommendation trust, until the set is empty. The complexity of this algorithm is exponential. Another distributed algorithm presented in [39] is used especially for tree-like network structures. The complexity is reduced to logarithmic.

Combination of Trust Values

Sometimes there are several recommendation paths so the trust relationships of the same trust class between two entities are often not unique. The trust values can then be used as collective information to compute a combined value [3].

First we show how to combine recommendation trust. Given n values of recommendation trust relationships between the same entities and with respect to the same trust class $V_i (i = 1 \ldots n), V_i \neq 0$, their combination V_{com} can be computed according to Equation10:

$$V_{com} = \frac{1}{n} \sum_{i=1}^{n} V_i \tag{10}$$

When there are several direct trust relationships between two entities with respect to the same trust class, the combination of these trust values can be obtained by Equation11:

$$\begin{aligned} V_{com} &= 1 - \prod_{i=1}^{m} n_i \sqrt{\prod_{j=1}^{n_i} (1 - V_{i,j})} \\ &= 1 - \prod_{i=1}^{m} n_i \sqrt{\prod_{j=1}^{n_i} \alpha^{\bar{V}_{i,j} \cdot p_i}} \\ &= 1 - \prod_{i=1}^{m} \alpha^{\frac{1}{n_i} (\sum_{j=1}^{n_i} \bar{V}_{i,j}) \cdot p_i} \\ &= 1 - \alpha^{\sum_{i=1}^{m} \frac{1}{n_i} (\sum_{j=1}^{n_i} \bar{V}_{i,j}) \cdot p_i} \end{aligned} \tag{11}$$

Conclusion

This trust model divides trust relationships into two types: direct trust and recommendation trust. It introduces trust values to substantiate the trust, then to derive new trust

value from existing ones. Different trust values can be combined together to evaluate the trustworthy of an entity.

The idea of recommendation trust was adopted by many other trust model applications which I will mention later.

The recommendation trust in this trust model often comes along a path. This will effective when the one-hop trust value has been known. But how to get the one-hop trust value is also a major problem that this trust model does not deal with.

3.4.2 A Distributed Trust Model with Recommendation Protocol

This distributed trust model was proposed by Alfarez et.al. in [1]. They introduced an approach to the problem of trust management with a distributed trust model, and a recommendation protocol. Different from the trust model mentioned in Section 3.4.1 which focuses on the derivation and combination of trust values, this trust model proposes a Recommendation Protocol to facilitate the propagation of trust information. According to [1], their proposal extends and generalizes some current approaches to security and trust management, based upon four goals:

1. To adopt a decentralized approach to trust management.
2. To generalize the notion of trust.
3. To reduce ambiguity by using explicit trust statements.
4. To facilitate the exchange of trust-related information via a recommendation protocol.

Trust Model Description

In this trust model, trust relationship is also divided into two types: direct trust relationship and recommendation trust relationship. Also the model uses trust value to represent the different levels of trustworthiness of an entity. But it adopts discrete trust levels instead of continuous values with any meaningful accuracy. The direct

Value	Meaning	Description
-1	Distrust	Completely untrustworthy.
0	Ignorance	Cannot make trust-related judgement about entity.
1	Minimal	Lowest possible trust.
2	Average	Mean trustworthiness. Most entities have this trust level.
3	Good	More trustworthy than most entities.
4	Complete	Completely trust this entity.

Table 2. Direct trust Value Semantics

Value	Meaning	Description
-1	Distrust	Completely untrustworthy.
0	Ignorance	Cannot make trust-related judgement about entity.
1	'Closeness' of recommender's judgement to own judgement about trustworthiness	
2		
3		
4		

Table 3. Direct trust Value Semantics

Figure 2. Example: Can Alice trust Eric the mechanic?

and recommendation trust values and their descriptions are given in Table 3.4.2 and Table 3.4.2 [1].

Recommendation Protocol

There are there types of messages used in this model: RRQ (Recommendation Request Message), Recommendation Message, and Refresh Message which is used for refreshing or revoking recommendation. Protocol flow in this recommendation protocol is very similar to the routing discovery process of AODV (Ad hoc On-demand Distance Vector Routing Protocol) which is used for ad hoc networks. The protocol flow can be described using an example from [1] as depicted in Figure 2.

To describe Figure 2 let us assume that Alice (the requestor) is requesting a recommendation from Bob (the recommender) about Eric (the target). Alice is interested in Eric's reputation for servicing cars, especially VW Golfs, one of which Alice drives (trust category ="CarService"). The protocol run is as follows.

1. $Alice \rightarrow Bob : Alice, rrqA01, Eric, [Car - Service], T, 20000101$
2. $Bob \rightarrow Cathy : Bob, rrqB01, Eric, [Car - Service], T, 20000101$
3. $Cathy \rightarrow Bob : Bob, rrqB01, [Cathy], [(Eric, Car{-}Service, 3, 20000131)], PK_{Eric}$
4. $Bob \rightarrow Alice : Alice, rrqA01, [Cathy, Bob], [(Eric, Car{-}Service, 3, 20000131)], PK_{Eric}$

The reputation of entities changes over time so there is a need to update the reputation information in the system. To revoke or refresh the recommendations, a recommender resents the same recommendation with trust value 0. The receiver will treat this as any other 0-value recommendation. Changing the trust value to any other value (i.e. (-1 ,1..4)) will refresh the recommendation.

Computing Trust

The algorithm for computing trust values in this trust model is simple. The following formula is used to compute the trust value of a target for a single recommendation path:

$$tv_p(T) = tv(R_1)/4 \times tv(R_2)/4 \times \ldots \times tv(R_n)/4 \times rtv(T) \tag{12}$$

Where,

$tv(R_i)$ Recommender trust value of recommenders in the return path including the first recommender (who received the original RRQ) and the last recommender (who originated the Recommendation). i is from 1 to n.

$rtv(T)$ The recommended trust value of target T given in the recommendation.

$tv_p(T)$ The trust value of target T derived from recommendation received through the return path p.

A requester may have multiple recommendations for a single target and thus the recommendations must be combined to yield a single value. For the moment, we have adopted the averaging method used by Thomas et. al. in [3]. Averaging evens out the impact of any single recommendation. The final single trust value for target T is computed as follows:

$$tv(T) = Average(tv_i(T), \ldots, tv_p(T) \tag{13}$$

Conclusion

The main advantage of this trust model is that it proposes a recommendation protocol to formalize the propagation of trust information. The main distinctive property of this model is that trust value here is discrete and divided into some trust levels. However, the trust value calculation algorithm is derived largely from intuition and lack of mathematical basis. A standard algorithm is necessary to reduce ambiguity in

trust value recommendations, and to allow most requesters to be confident in the that what is received in recommendations comes close to that from a universal standard. Furthermore, there is also a need to look into monitoring and revising trust of other entities, so that the dynamic and non-monotonic properties of trust in the model can be maintained.

3.4.3 Trust Model based on Dempster-Shafer Theory

This model is presented in the paper [36], which proposes a method to propagate and quantify trust using principles derived from Dempster-Shafer theory of evidence. This trust model is designed mainly for e-commerce environment. There are some other trust models designed for e-commerce and internet security, but most of them are used for the purpose of authenticating a public key to its owner. While this model tries to describe the scenarios where trust exists between a vendor and a customer with several intermediaries involved in a transaction in an e-commerce setting.

Dempster-Shafer Theory

Propagation of trust is a major issue when several entities are involved in e-commerce transactions. This model uses Desmpster-Shafer theory to solve the trust problem because of Desmpster-Shafer theory of evidence is able to represent "Certainty about certainty". The Dempster-Shafer theory of evidence aims to model and quantity uncertainty by degrees of belief. The mathematical model proposed by Shafer [35] is based on the notion of belief functions and Dempster's rule of combination. Ginsberg proposes a procedure for uncertain reasoning using Dempster-Shafer theory in [12], which is a straightforward application of Dempster-Shafer theory.

The most important assumption made by Ginsberg is that his model applies to dichotomous frames only, i.e., those which account for two propositions. Consequently, the $belief$ in a proposition A can be represented by a tuple (a, b) where a measures the extent to which one believes the proposition A and b measures the disbelief, i.e., belief in the complementary proposition $notA$.

To perform combinations, Demspter-Shafer theory gives us a rule. For example, if we

Figure 3. Dempster-Shafer's trust matrix between buyer and trust authority. Upward is belief, while downward is disbelief.

denote $(a, b) + (c, d)$ as the inference obtained by combining the two inferences (a, b) and (c, d), the combination formula is:Equation 14.

$$(a, b) + (c, d) = (1 - \frac{\bar{a}c}{1 - (ad + bc)}, 1 - \frac{\bar{bd}}{1 - (ad + bc)}), \quad if \quad ad + bc \neq 1. \quad (14)$$

Trust matrix

This model uses trust matrix instead of single trust value to represent the trust relationship between two entities. Trust matrices are maintained by certain authorities, called Trust Authorities (TA), and updated based on the information that TAs receive from each completed transaction. In this trust model, the authors define two types trust matrix, one is trust relationship between customer and trust authority, and another is between vender and trust authority. The Figure 3 shows the former relationship.

The trust matrix between vender and customer is shown in Figure 4.

Given the trust matrix between a customer and TA and the matrix between TA and a vendor, the new trust matrix between the customer and teh vendor can be derived by merging these two matrices using the above Dempster-Shafer formula. The newly generated trust matrix is described in Figure 5 [36].

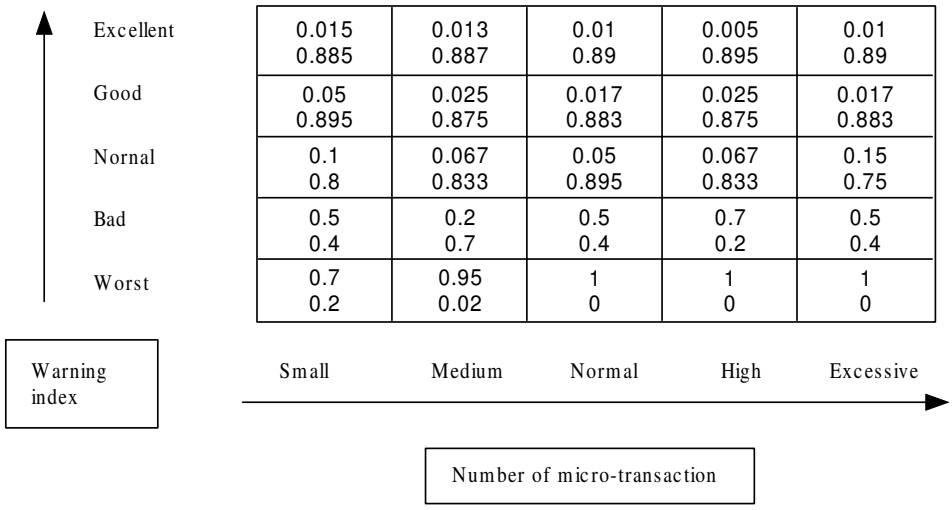


Figure 4. Dempster-Shafer's trust matrix between vendor and trust authority. Upward is belief, while downward is disbelief.

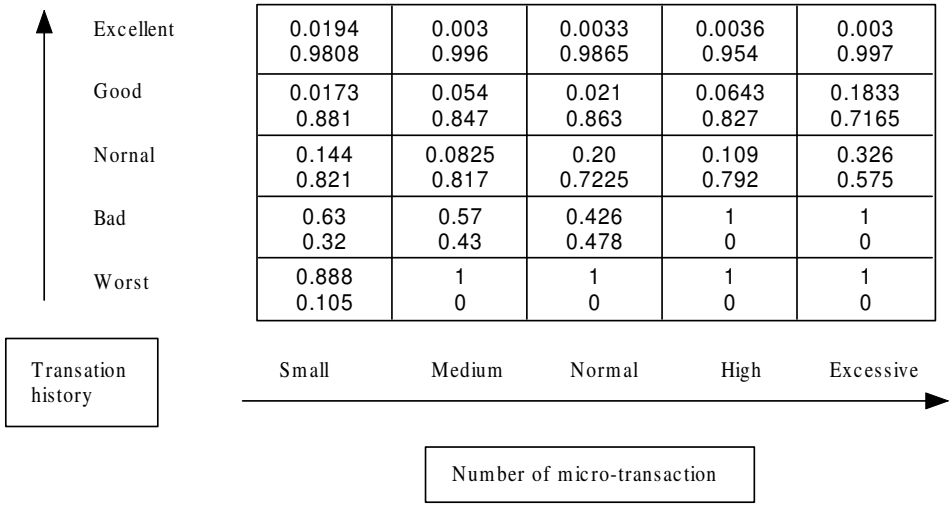


Figure 5. Trust Matrix formed by Merging the Two Trust Matrices of Figure 3 and Figure 4 based on Dempster-Shafer formula.

Conclusion

This trust model uses a trust matrix instead of a single trust value to represent the trust relationship between two entities in e-commerce transactions. Two or more trust matrices are combined into one new trust matrix using Dempster-Shafer's combination rule.

It's easier to maintain trust matrices in e-commerce environment because there are certain trust authorities which can keep the transaction history, warning index and number of micro-transactions etc.,which provide the needed information in the trust matrices in an e-commerce environment. But for ad hoc networks without centralized authorization servers it is much complicated and not realistic to monitor and record all these information details.

In this model the belief to a real event has been interpreted as upper and lower probability bounds, respectively, according to the Dempster-Shafer theory. The two value bounds may increase the computation complexity when performing combination of trust values. But if the trust between two entities can be viewed as a probability and the combination functions can only be used to estimate probability values, there is no need to set the upper and lower bounds of one's belief.

3.4.4 Trust Model using Fuzzy Logic

This model is introduced in [28]. The main difference between this model and the model described in Section 3.4.3 is that this one uses fuzzy logic to combine the trust matrix and verify the transactions so as to extend trust to transacting entities suitably. Furthermore, And this model proposes a trust protocol which can be used in electronic ecommerce to protect the trust information from breach of privacy.

Trust Matrix

This trust model uses Weighted Trust Surface (WTS) and Fuzzy Trust Surface (FTS) to represent the trust relationships between two entities during transactions. These two matrices are shown in the Figure 6 and Figure 7.

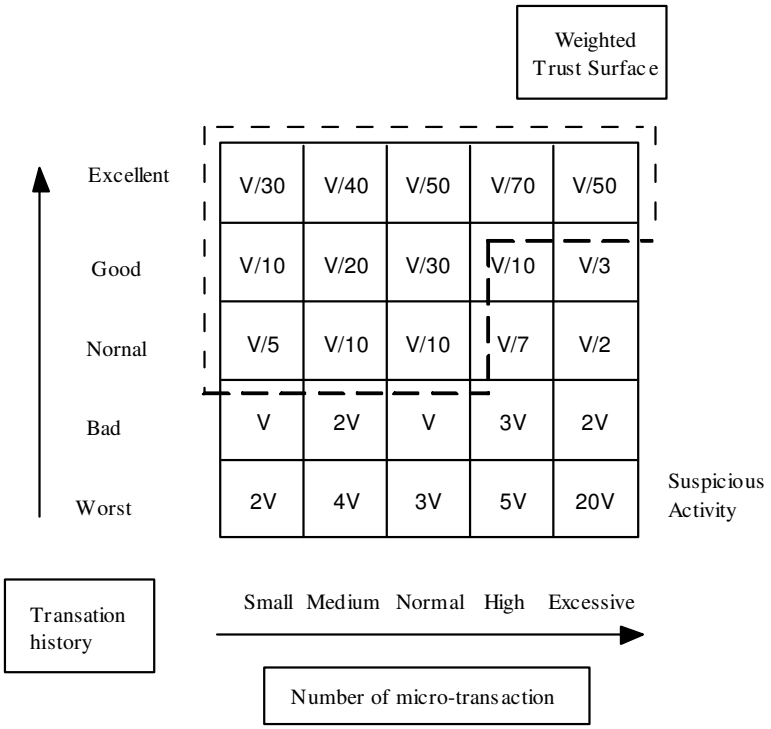


Figure 6. Weighted Verification of Transactions

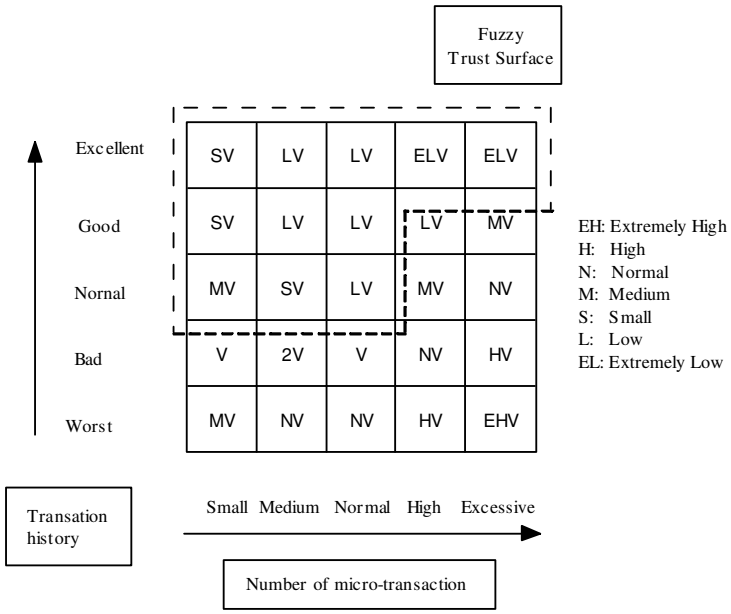


Figure 7. A Fuzzy Trust Matrix

The letter V in Figure 6 and Figure 7 means that the corresponding transaction should be verified. In Figure 6, $V/50$ means that one in fifty transactions needs to be verified. $20V$ means that the corresponding transaction may be verified more thoroughly for 20 times.

The fuzzy trust surface is then generated by replacing the numeric values by fuzzy subsets of linguistic values, shown in Figure 7. This allows easy interpretation of the matrix entities.

Fuzzy Logic Inference

A set of trust matrices is obtained by engaging trust propagation techniques. Fuzzy inference can then be applied on the trust matrices to perform the various actions - verification, indemnity required, etc.

Definition (Zadeh's Compositional Rule of Inference) [41],[5]: Let $R(x)$, $R(x,y)$ and $R(y)$ where $x \in X$, $(x,y) \in X_X Y$, and $y \in Y$ be fuzzy relations in X, $X_X Y$ and Y respectively. Let A and B denote particular fuzzy sets in X and $X_X Y$. Then the compositional rule of inference asserts that the solution of $R(x) = A$ and $R(x,y) = B$ is given by $R(y) = A \circ B$ where $A \circ B$ is the composition of A and B.

Application (Fuzzy Logic Inference) [5]: Let A and B be fuzzy sets defined over X and Y respectively. A fuzzy rule $A \rightarrow B$ is first transformed into fuzzy relation $R_{A \rightarrow B}$ that represents a correlation between A and B and is defined as

$$\mu_R(x,y) = min(\mu_A(x), \mu_B(y)); x \in X, y \in Y. \tag{15}$$

Given a fact A' and a rule $A \rightarrow B$, applying Zadeh's compositional rule gives

$$\begin{aligned} B' &= A' \circ R_{A \rightarrow B} \\ \mu_{B'}(y) &= max_x min(\mu_{A'}(x), \mu_R(x,y)) \\ &= min(\alpha, \mu_B(y)) \end{aligned} \tag{16}$$

where $\alpha = maxmin(\mu_{A'}(x), \mu_A(x))$.

Fuzzy logic inference is used in building fuzzy expert systems to reason on trust parameters. Another example of trust systems (though not from the point of electronic

commerce) using expert systems was discussed in Referee [22]. A fuzzy logic based expert system using Fuzzy CLIPS [23] (Fuzzy CLIPS is an expert system building tool) is used to carry out the inference process.

Conclusion

The main contributions of this trust model can be concluded as follows. Firstly, in this model, the identification and measurement of variables of trust are based on the quantifiable notion for trust. Secondly, this model uses fuzzy verification of transactions. Thirdly, the propagation of trust, and computing a single trust matrix are performed between the customer and the vendor that governs the transaction. Another contribution is that this model proposes a suitable protocol to protect privacy of trust information.

3.4.5 Trust Model using Subjective Logic

Subjective logic was proposed by Audun Jϕsang [19], [20], [21]. Subjective logic is "a logic which operates on subjective beliefs about the world, and uses the term opinion to denote the representation of a subjective belief" [21]. The trust between two entities is then represented by opinion. An opinion can be interpreted as a probability measure containing secondary uncertainty, and as such subjective logic can be seen as an extension of both probability calculus and binary logic.

Subjective logic is different from fuzzy logic. Fuzzy logic "operates on crisp and certain measures about linguistically vague and fuzzy propositions whereas subjective logic operates on uncertain measures about crisp propositions" [21].

The trust models that were introduced before mainly use discrete value or continuous probability to represent trust. But the discrete values are not sufficient because they can only provide a small set of possible trust values, while the continuous values and probability often seem counterintuitive when using their operators to combine trust. That is, some components such as ignorance and uncertainty which can not be reflected by probabilities are missing when modelling trust as probability [20].

To represent uncertain probabilities, subject logic uses elements derived from Dempster-Shafer belief theory. But different from Shafer's theory in which belief functions and

possibility measures have been interpreted as upper and lower probability bounds, the belief functions used in subject logic is to estimate probability values instead of setting bounds because the probability of a real event can never be determined with certainty, and neither can upper or lower bounds be set accordingly.

Opinion Model

Opinion is originally a 3-dimensional metric in representing belief or trust and is extended to contain a 4th redundant parameter for simple usage in combination with logical operators. The definition of opinion is as follows [21]:

> **Definition (Opinion)** *Let Θ be a binary frame of discernment with 2 atomic states x and $\neg\ x$,and let m_Θ be a BMA on Θ where b(x), d(x), u(x), and a(x) represent the belief, disbelief, uncertainty and relative atomicity functions on x in Θ respectively. Then the opinion about x, denoted by ω_x, is the quadruple defined by:*
>
> $$\omega_x \equiv (b(x), d(x), u(x), a(x)). \tag{17}$$

BMA is a belief mass assignment on Θ whose definition is [21]:

> **Definition (Belief Mass Assignment)** *let Θ be a frame of discernment. If with each substate $x \in 2^\Theta$ a number $m_\Theta(x)$ is associated such that:*
>
> $$\begin{aligned} &1.\quad m_\Theta(x) \geq 0 \\ &2.\quad m_\Theta(x) = 0 \\ &3.\quad \sum_{x \in 2^\Theta} m_\Theta(x) = 1 \end{aligned} \tag{18}$$
>
> *then $m_\Theta(x)$ is called a belief mass assignment[1] on θ, or BMA for short. For each substate $x \in 2^\Theta$, the number $m_\Theta(x)$ is called the belief mass[2] of x.*

Note $b(x)$ represent the belief function which is interpreted as an observer's total belief that a particular state is true. $d(x)$ is disbelief function that is interpreted as the total belief that a state is not true. And the uncertainty function $u(x)$ represents an observer's uncertainty regarding the truth of a given state. The sum of $b(x), d(x), u(x)$ is 1, that is:

$$b(x) + d(x) + u(x) = 1 \tag{19}$$

So the uncertainty function can be interpreted as something that fills the void in the absence of both belief and disbelief.

Combination of Opinions

There are two operators for combination opinion in this opinion model: discounting and consensus.

Discounting: Assume two entities A and B where A has an opinion about B and B has an opinion about a proposition x. Entity A can then form an opinion about x by discounting B's opinion about x with A's opinion about B. The discounting definition [21] is:

> **Definition (Discounting)** *Let A and B be two agents where* $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$ *is A'S opinion about B's advice, and let x be a proposition where* $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$ *is B's opinion about x expressed in an advice to A. Let* $\omega_x^{AB} = (b_x^{AB}, d_x^{AB}, u_x^{AB}, a_x^{AB})$ *be the opinion such that*

$$\begin{aligned}
&1.\ \ b_x^{AB} = b_B^A b_x^B \\
&2.\ \ d_x^{AB} = b_B^A d_x^B \\
&3.\ \ u_x^{AB} = d_B^A + u_B^A + b_B^A u_x^B \\
&4.\ \ a_x^{AB} = a_x^B
\end{aligned} \tag{20}$$

then ω_x^{AB} is called the discounting of ω_x^B by ω_B^A expressing A's opinion about x as a result of B's advice to A. By using the symbol '$\otimes$' to designate this operator, we define $\omega_x^{AB} \equiv \omega_B^A \otimes \omega_x^B$.

Consensus: The consensus opinion of two opinions is an opinion that reflects both opinions in a fair and equal way. As presented in [21], the consensus is defined as follows:

Definition (Consensus) *Let $\omega_x^A = (b_x^A, d_x^A, u_x^A, a_x^A)$ and $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$ be opinions respectively held by agents A and B about the same proposition x. Let $\omega_x^{A,B} = (b_x^{A,B}, d_x^{A,B}, u_x^{A,B}, a_x^{A,B})$ be the opinion such that*

$$
\begin{aligned}
&1.\ b_x^{A,B} = (b_x^A u_x^B + b_x^B u_x^A)/\kappa \\
&2.\ d_x^{A,B} = (d_x^A u_x^B + d_x^B u_x^A)/\kappa \\
&3.\ u_x^{A,B} = (u_x^A u_x^B)/\kappa \\
&4.\ a_x^{A,B} = \frac{a_x^B u_x^A + a_x^A u_x^B - (a_x^A + a_x^B) u_x^A u_x^B}{u_x^A + u_x^B - 2u_x^A u_x^B}
\end{aligned}
\tag{21}
$$

where $\kappa = u_x^A + u_x^B - 2u_x^A u_x^B$ such that $\kappa \neq 0$, and $a_x^{A,B} = (a_x^A + a_x^B)/2$ when $u_x^A, u_x^B = 1$. Then $\omega_x^{A,B}$ is called the consensus between ω_x^A and ω_x^B, representing an imaginary agent $[A, B]$'s opinion about x, as if the represented both A and B. By using the symbol '$\oplus$' to designate this operator, we define $\omega_x^{A,B} \equiv \omega_x^A \oplus \omega_x^B$.

This model can be used to assess the testimony from different witnesses. We take the example from [21].

There are three witnesses $W1, W2, W3$ who are giving testimony to express their opinions about a verbal proposition x which has been made about the accused. The judge J has to determine his own opinion about x. This situation is illustrated in Figure 8.

The effect of each individual testimony on the judge J can be computed using the discounting operator, so that for example $W1$'s belief in x is discounted by the judge's

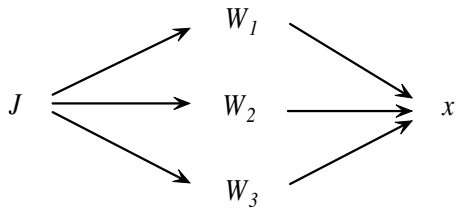


Figure 8. Trust in testimony from witnesses

trust in $W1$. This causes the judge to have the opinion about the truth x as a result of the testimony from $W1$:

$$\omega_x^{JW_1} = \omega_{W_1}^J \otimes \omega_x^{W_1} \tag{22}$$

Assuming that the opinions resulting from each witness are independent, they can finally be combined using the consensus operator to produce the judge's own opinion about x:

$$\omega_x^{J(W_1,W_2,W_3)} = (\omega_{W_1}^J \otimes \omega_x^{W_1}) \oplus (\omega_{W_2}^J \otimes \omega_x^{W_2})) \oplus (\omega_{W_3}^J \otimes \omega_x^{W_3})) \tag{23}$$

Conclusion

This trust model using subjective logic can express the human cognitive phenomenon better than previous trust models. It introduces the term opinion to represent the ignorance and uncertainty about a proposition and it is more suitable for the expression of human's subjective consciousness. The discounting and consensus operators are quite useful in the situations that one entity needs to give its own belief about a proposition when given many different recommendations.

3.5 Comparison of Trust Models

The trust models discussed before have their own application fields. For the application of ad hoc network security, we prefer to use the model of subjective logic because it

inherits many advantages when compared with other trust models. The following is the comparison between subjective logic with the other trust models:

Subjective Logic vs. Fuzzy Logic Fuzzy Logic operates on certain measures about fuzzy propositions while subjective logic operates on uncertain measures about crisp propositions. Because of the mobility and flexibility of ad hoc networks, the nodes in the networks often don't know each other. So a node is often ignorant of the trustworthiness about another node. This kind of uncertainty measures belongs to the category of subjective logic.

Subjective Logic vs. Dempster-Shafer Theory Dempster-Shafer Theory takes uncertainty and ignorance into consideration but it interprets the possibility measures as upper and lower probability. While in reality we usually want to just estimate probability values not to set its bounds. The uncertainty function provided by subjective logic is a more direct way to express the uncertainty.

Opinion in Subjective Logic vs. Continuous Probability The definition about trust in [10] represents trust as a subjective probability. However, this definition misses some important components of human intuitiveness: uncertainty and ignorance. While opinion includes belief, disbelief and also uncertainty, thus it can reflect more consciousness of human beings. Nodes in ad hoc networks are the same that they will also have no ideas about the trustworthiness of other nodes. So

Opinion in Subjective Logic vs. Discrete Trust Value Obviously the discrete trust value can only express limited information about one's belief. But the concept of dividing trust into levels or degrees may be useful when one node need updating trust values if his opinion about another has been changed. We will talk about this concept again when we describe the secure routing protocol which is based on our trust model.

4 Self-Organized Secure Routing Protocol based on Trust Model

4.1 Assumptions

In this work, we make some assumptions to simplify the design complexity and to focus on the key issues that we are interested in.

The ad hoc network we work on contains a number of mobile nodes who communicate with one another through an error-prone, bandwidth-limited and insecure wireless channel. We do not concern the security problem introduced by the instability of physical layer or link layer. Our security design is mainly focus on the network layer and some key management issues belonging to the application layer. Besides we also make other assumptions as follows:

1. Each node in the network has the ability to recover all of its neighbors.
2. Each node in the network can broadcast some essential messages to its neighbors with high reliability.
3. Each node in the network has a unique ID that can be distinguished from others.
4. This ad hoc network has been equipped with some monitor mechanism or intrusion detection units so that one node can easily observe the behaviors of its one-hop neighbors. These mechanisms have been proposed in some previous work, such as intrusion detection system in [43] and watchdog technique in [30]. It is more difficult and expensive to do intrusion detection in ad hoc networks than in wired networks because ad hoc networks inherit openness and dynamic topology by nature with no centralized monitoring point. To develop a light-weight and efficient monitoring mechanism is an important issue for the security of ad hoc networks.

4.2 Design Goals

Our general design goal is to provide a security solution for ad hoc networks. Some researchers have been trying to achieve this goal from different aspects such as secure

routing protocols, cryptographical schemes. Actually, these solutions all imply a common concept of creating trust between entities. Trust is a general and basic concept of human life while ad hoc networks are very similar as human society from the views of communication ways and roles of nodes. It is a natural idea to design a trust model which can be used in an ad hoc network and meet its security requirements. The trust model should be used for the key management and for the design of a secure routing protocol. Therefore, our main design goals are:

1. Design a suitable decentralized trust model that can be used for the security solutions of ad hoc networks.
2. Apply this trust model to design a flexible self-organized key management scheme.
3. Apply this trust model to design a secure and flexible self-organized routing protocol.
4. Articulate and demonstrate the principle of the trust model and the security advantages of the resulting ad hoc networks.

4.3 Our Trust Model for Ad Hoc Networks

4.3.1 Trust Relationships in Ad Hoc Networks

In ad hoc networks, a trust relationship exists between two nodes if one holds a belief about the other's trustworthiness. According to the properties introduced in Section 3.2, the trust relationships in our trust model for ad hoc networks has the following characteristics:

Relativity The trustworthiness between two nodes can be used to issue a certificate of public key and can also be used to perform routing discovery. So trust relationships in ad hoc networks should be classified into categories so that a node can express trust towards another about particular characteristics or aspects of that node.

One-to-one relationship In our trust model, the trust relationship only exists between exactly two nodes. That means a node cannot hold one general belief about a group of nodes.

Asymmetry The trust relationship in our model is non-symmetrical. Node A has an opinion about B's trustworthiness while it is not necessary for B to have an opinion about A's trustworthiness. Even if B has this opinion, these two opinions do not need to be equal.

Conditional Transitivity In our trust model, the transitivity of trust relationship is conditional. A, B, C are three nodes on the same routing path in an ad hoc network. A has a trust belief about B and so does B to C, then the trust belief form A to C cannot be simply passed from A to B to C. We will present a combination algorithm to combine these two beliefs into one.

There are two types of trust relationships in our trust model for ad hoc networks: direct trust and recommendation trust. Direct trust can be obtained from the direct communication with other nodes. It's the estimate about other nodes' trustworthiness from the observations of itself. Recommendation trust is acquired from the combination of the recommendation opinions from other nodes.

4.3.2 Representation of Trust

In our trust model, trust is represented on the basis of subjective logic. We also use the term *Opinion* to represent the trust or belief from one node to another. To make the representation more straightforward, our *Opinion* only contains three elements: belief, disbelief and uncertainty. The forth elements "relative atomicity" in original definition of opinion in subject logic given by [21] is omitted. We may add some new elements in this definition in our future work. So our definition of *Opinion* is as follows:

> **Definition 1 (Opinion)** *let Θ be a binary frame of discernment with 2 atomic states x and $\neg$ x, and let m_Θ be a BMA on Θ where b(x), d(x), u(x) represent the belief, disbelief, and uncertainty functions on x in Θ*

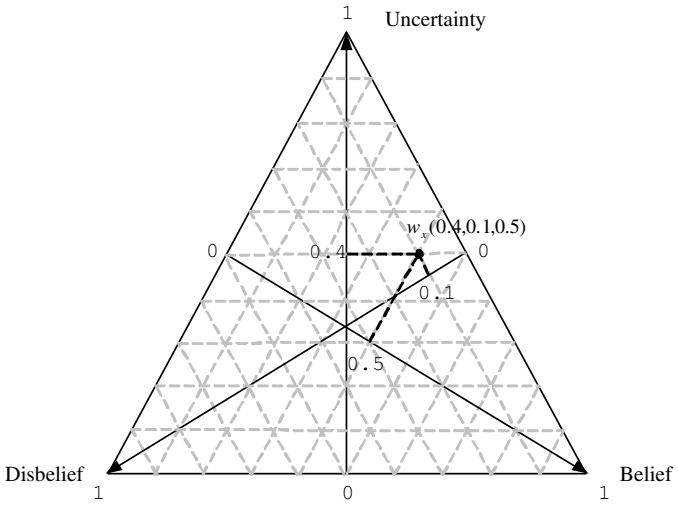


Figure 9. Trust in testimony from witnesses

respectively. Then the opinion about x, denoted by ω_x, is the triple defined by:

$$\omega_x \equiv (b(x), d(x), u(x)) \tag{24}$$

The definition of frame of discernment, BMA, belief, disbelief and uncertainty functions are the same as those definitions in Section 3.4.5 and [21]. For compactness and simplicity of notation we will denote the belief, disbelief and uncertainty as b_x, d_x and u_x respectively in the following of this paper. *Opinion* can be illustrated graphically using triangle as shown in Figure 9.

We have such a theorem about the relationship between b_x, d_x and u_x as following:

Theorem 1 (Belief Function Additivity)

$$b_x + d_x + u_x = 1 \tag{25}$$

Proof 1 The sum of the belief, disbelief and uncertainty functions is equal to the sum of the belief masses in a BMA which according to the definition of BMA in section 3.4.5. So it sums up to 1.

From Equation 25 we can see that the uncertainty function represents an observer's uncertainty regarding the truth of a given state. It can be interpreted as something that

fills the void in the absence of both belief and disbelief. Uncertainty is a very important and useful variable when we use this trust model to design a secure routing protocol for ad hoc networks.

In ad hoc networks a node's opinions about the other nodes' trustworthiness will change after they communicate with each other. So the Opinion should be a dynamic variable. This issue was not mentioned in [21] but it is a common phenomenon in the real applications. We will present a direct opinion update algorithm when we talk about the design of secure routing protocol.

4.3.3 Combination of Recommendation Trusts in Ad Hoc Networks

Because of the high mobility of ad hoc networks, one node can not trust any other nodes definitely. Node A may first listen to other node's opinions about node B before exchanging any information with B, then combine all these different recommendation opinions together according to some rules to get his own opinion about node B. Then A will decide whether to go on communicating with B or not. This procedure is the combination of trust. This subsection we will give out such combination rules of our trust model.

There are two types of combination in our model: Discounting Combination and Consensus Combination.

Discounting Combination

Let's consider such a situation: Node A want to know C's trustworthiness then node B gives his opinion about C. And A already has an opinion about B. So A will combine the two opinions: A to B, B to C to obtain a recommendation opinion A to C. Discounting combination is for this purpose.

Definition 2 (Discounting Combination)

Let A, B and C be three nodes where $\omega_B^A = (b_B^A, d_B^A, u_B^A)$ *is A'S opinion about B's trustworthiness, and* $\omega_C^B = (b_C^B, d_C^B, u_C^B)$ *is B's opinion about C's trustworthiness. Let* $\omega_C^{AB} = (b_C^{AB}, d_C^{AB}, u_C^{AB})$ *be the opinion such that*

1. $b_C^{AB} = b_B^A b_C^B$
2. $d_C^{AB} = b_B^A d_C^B$
3. $u_C^{AB} = d_B^A + u_B^A + b_B^A u_C^B$

then ω_C^{AB} is called the discounting of ω_C^B by ω_B^A expressing A's opinion about C as a result of B's advice to A. By using the symbol '$\otimes$' to designate this operator, we define $\omega_C^{AB} \equiv \omega_B^A \otimes \omega_C^B$.

The discounting combination can be used along a recommendation path.

Consensus Combination

Different nodes may have different, even contrary opinions about one node. To combine these opinions together to get a relative objective evaluation about that node's trustworthiness, we use Consensus combination.

Definition 3 (Consensus Combination)

Let $\omega_C^A = (b_C^A, d_C^A, u_C^A)$ and $\omega_C^B = (b_C^B, d_C^B, u_C^B)$ be opinions respectively held by nodes A and B about node C's trustworthiness. Let $\omega_C^{A,B} = (b_C^{A,B}, d_C^{A,B}, u_C^{A,B})$ be the opinion such that

1. $b_C^{A,B} = (b_C^A u_C^B + b_C^B u_C^A)/k$
2. $d_C^{A,B} = (d_C^A u_C^B + d_C^B u_C^A)/k$
3. $u_C^{A,B} = (u_C^A u_C^B)/k$

where k=$u_C^A + u_C^B - 2u_C^A u_C^B$ such that $k \neq 0$, Then $\omega_C^{A,B}$ is called the consensus between ω_C^A and ω_C^B, representing an imaginary node $[A, B]$'s opinion about C's trustworthiness, as if it represented both A and B. By using the symbol '$\otimes$' to designate this operator, we define $\omega_C^{A,B} \equiv \omega_C^A \oplus \omega_C^B$.

The consensus combination can reduce the uncertainty of one's opinion.

These two types of combinations will normally be used together for a node to judge another node's trustworthiness in ad hoc network applications.

Let's take an example to illustrate these two combination algorithms.

Suppose that in an ad hoc network environment, A has three neighbors N_1, N_2 and N_3, and A want to know B's trustworthiness. Now A's opinion about B should be (0, 0, 1). These neighbors' opinions about B is :

$$\omega_B^{N_1} = (0.90, 0.00, 0.10)$$
$$\omega_B^{N_2} = (0.90, 0.00, 0.10)$$
$$\omega_B^{N_3} = (0.90, 0.00, 0.10)$$

A's opinion about N_1, N_2, N_3 are:

$$\omega_{N_1}^A = (0.90, 0.00, 0.10)$$
$$\omega_{N_2}^A = (0.00, 0.90, 0.10)$$
$$\omega_{N_3}^A = (0.10, 0.00, 0.90)$$

First we will use discounting combination algorithm (Definition 2 in Section 4.3.3) to compute the separate opinions about B, then these opinion are:

$$\omega_B^{A,N_1} = (0.81, 0.00, 0.19)$$
$$\omega_B^{A,N_2} = (0.00, 0.00, 1.00)$$
$$\omega_B^{A,N_3} = (0.09, 0.00, 0.91)$$

Then we use consensus combination algorithm (Definition 3 in Section 4.3.3) to combine these new opinions again into one opinion. The result is:

$$\omega_B^{A,(N_1N_2N_3)} = (0.8135, 0.0000, 0.1865)$$

So A will consider B as 81.35 percents trustable. And the uncertainty about B has decreased from 1 to 0.1865.

Type	*Requestor*	*Recommender*	*Recommendee*	*Opinion*	*Class*	*Expiry*

Type: 0 for TREQ, 1 for TREP, and 2 for TWARN

Opinion include belief, disbelief and uncertainty

Figure 10. Message Structure of Trust Recommendation Protocol

4.3.4 Trust Recommendation Protocol

Some previous trust models seldom concern the exchange of trust information. However, it is very necessary to design an information exchange mechanism when applying the trust models into network applications.

In our trust recommendation protocol, there are three types of messages: Trust Request Message (TREQ), Trust Reply Message (TREP), and Trust Warning Message (TWARN). Nodes who issue TREQ messages are called *Requestor*. Those who reply TREP messages are called *Recommender*. The recommendation target nodes are called *Recommendee*. Any node may be a *Requestor*, a *Recommender*, or a *Recommendee*. These three types of messages share a common message structure, which is shown in Figure 10.

When a node wants to know another node's new trustworthiness, it will issue a TREQ message to its neighbors. TREQ message uses the above structure and leaves the fields of *Recommendor*, *Opinion* and *Expiry* empty. The *Type* field is set to 0. Nodes who receive the TREQ message will reply with a TREP message with the *Type* field set to 1. When a node thinks that another node has become malicious or unreliable, it will broadcast a TWARN message to its neighbors. Now the *Type* is set to 2, and the *Opinion* field is set to (0,1,0).

4.4 Trusted Routing Protocol based on AODV

In this section we will describe our trusted routing protocol in detail. This protocol is based on AODV routing protocol of ad hoc networks which we introduced in Section 2.1.

	Opinion	Success Commu.	Failed Commu.
B	$(0.90, 0.00, 0.10)$	7	0
C	$(0.00, 0.90, 0.10)$	0	1
D	$(0.10, 0.00, 0.90)$	1	0

Table 4. Trust table in node A

4.4.1 Node Model

Each node in an AODV based ad hoc network will have an opinion towards each other. As defined above, opinion is a triple containing belief, disbelief and uncertainty about another node's trustworthiness. Each of them is a continuous value between 0 and 1. For example, if disbelief is 1, it means that the probability of "another node can not be trusted" is 1. The three values in opinion are also divided into levels logically for the use of updating direct trust.

Each node maintains a trust table in its storage space. A's neighbors and nodes who have communicated with node A have entries in A's trust table. This table in node A is something like Table 4.4.1. As shown in this table, A has three entries in its trust table. "Success Commu." and "Failed Commu." mean the successful communication times and failed communication times separately that A performs with B, C or D.

When a node A first joins into an ad hoc network, its opinions about the other nodes' trustworthiness and the opinion that the other nodes think about node A are both $(0, 0, 1)$. That is, the probability of the others totally believe A is 0, the probability of the others totally disbelieve A is also 0, and the probability of the others have no idea of A's trustworthiness is 1. After communicating with the others A will gradually have more certain opinions about the others and so will the others to A.

4.4.2 General Framework

In our trusted routing protocol we also use some cryptographic technologies such as digital signature and hash functions to authenticate the routing information when

needed. So we assume that the keys and certificates needed by these cryptographic technologies have been obtained through some key management procedures before the node performs routing discovery. And these key management schemes should also use the idea of trust model to issue certificates of public keys or something else in a self-organized way.

Trust model is a good supplement to the cryptographic technologies that have been applied into the security of ad hoc networks. [15] and [42] are the latest security schemes for securing ad hoc networks which use digital signatures or one-way hash algorithms. However, these schemes completely neglect the trust relationships that do exist between nodes in ad hoc networks. The basis of their operations is "blind untrust". Although these schemes can provide more secure solutions to the security of routing information, they really decrease the efficiency of routing discovery because the computation of digital signature and hash algorithms in each operation are really a huge time and performance consuming procedure. In our design, trust relationships are taken into account to reduce the unnecessary cryptographic operations so that we can increase the performance of securing routing information without decreasing the security in the mean while.

There are mainly five components in our design of trusted routing protocol: trust recommendation trust combination algorithm, cryptographic procedures, trusted routing discovery and maintenance, and trust updating algorithm. The first two components belong to our trust model and the rest three components are in the trusted routing protocol part. The structure and relationship among these four components are illustrated in Figure 11.

The general procedure of nodes establishing trust relationships among one another and performing routing discovery is as follows.

Let us first imagine the beginning of an ad hoc network which contains few nodes. Each node's opinion towards one another initially is (0,0,1). Suppose node A wants to discover a route path to B. Because the uncertainty element in A's opinion towards others is larger than or equal to 0.5, which means that A is not sure whether it should belief or disbelief any other nodes, then A will use the cryptographic schemes as pro-

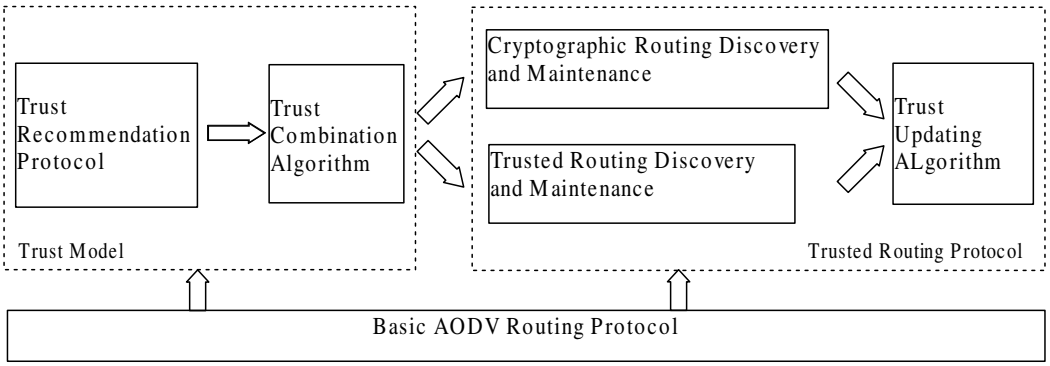


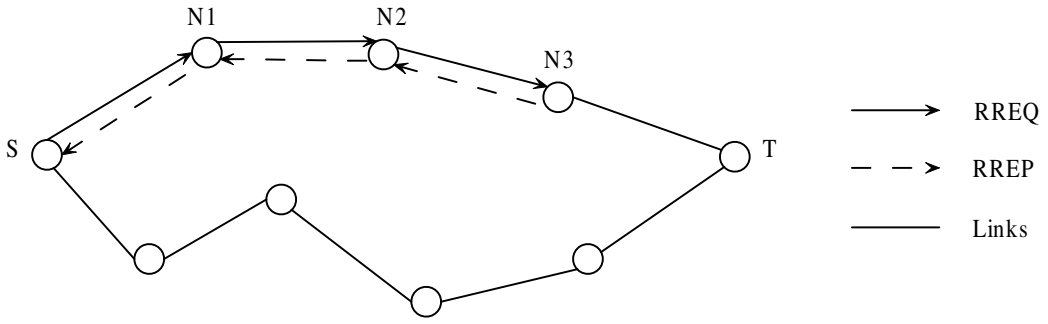
Figure 11. Framework of Trusted Routing Protocol

posed in SAODV [42] to perform routing discovery operations. After some successful or failed communications, A will change its opinions towards other nodes gradually using the trust updating algorithm. The uncertainty elements in its opinions about other nodes will be mostly larger than 0.5 after a period of time. By this way, each node in this ad hoc network will have more certain opinions towards other nodes eventually after this period of initial time.

Once the trust relationship has been established among most of the nodes in this ad hoc network, node can use our trusted routing protocol which is based our trust model to perform routing operations now. Node A now will use trust recommendation protocol to exchange trust information about a node B from its neighbors, then use the trust combination algorithm to combine all the recommendation opinions together and calculate a new option towards B. The following routing discovery and maintenance operations will follow the specifications of our trusted routing protocol.

The situation that one node first joins into this ad hoc network can be handled the same way as the beginning of this whole network.

In this framework, the establishment of trust relationships among nodes and the discovery of routing are all performed in a self-organized way, which is achieved by the cooperation of different nodes to exchange information and achieve agreements without any other third-party's interventions.



Routing Request:

S: issues RREQ.
N1: verifies opinions N1 →S, N1 →T.
N2: verifies opinions N2 →N1, N2 →S, and N2 →T.
N3: verifies opinion N3 →N2.

Routing Reply:

N3: has route entry to T, then issues RREP.
N2: verifies opinion N2 →N3.
N1: verifies opinion N1 →N2.
S: verifies opinion S →N1.

Figure 12. Trusted Routing Discovery Example

4.4.3 Trusted Routing Discovery

ROUTE REQUEST massages (RREQ) and ROUTE REPLY massages (RREP) are two kinds of routing information used in AODV routing protocol. We will describe the detail of our trusted routing discovery process from four aspects: send RREQ, receive RREQ, send RREP and receive RREP. Figure 12 shows a routing discovery procedure from a source S to target D.

When a node S wants to discover a route path to a destination D, it will broadcast a RREP packet to its neighbors with the basic format prescribed in AODV routing protocol.

When a node $N2$ receives from $N1$ a RREQ message whose source is S and target is D and $N2$ has no route entry to the target, it first checks the validity of this routing request. After that $N2$ broadcasts a Trust Request message (TREQ) to its neighbors to request their opinions about $N1$ using Trust Recommendation Protocol. Then $N2$

combines the received recommendation opinions together using trust consensus combination algorithm to get its new opinion about $N1$. If the new opinion does not satisfy:

$$belief \geq 0.5,\ disbelief < 0.5,\ uncertainty < 0.5 \tag{26}$$

$N2$ will discard this RREQ packet and at the same time modify its trust table using trust updating algorithm which will be discussed in the Section 4.4.4. And if the new opinion does satisfy the above criteria, $N2$ will go on calculating the new opinions of the source S and the target T. But if this RREQ has passed three hops to reach here, this node has no need to calculate its new opinions about S and T. Because according to the trust combination algorithm described in Section 4.3.3, the longer the recommendation path is, the smaller the uncertainty value will be. If this RREQ has passed more than two hops, that means, the intermediate nodes along the path from S to T before the current node have certain and positive belief about S and T's trustworthiness. Therefore, the current node is not necessary to verify the trustworthiness of S and T again. After doing all the above verification, $N2$ now will check if it has a route entry for target T. If yes, it will issue a RREP message back to $N1$; and if not, it will re-broadcast this RREQ to its neighbors.

When target T receives the RREQ message, it will also do the above verifications and create a RREP message back to the previous nodes.

When a node $N2$ receives from $N3$ a RREP message, it will do the same verification to $N3$ according to the criteria in Equation 26. If passes, the RREP will go on returning to the source node; if not, this RREP will be discarded. The trust table in $N2$ will be updated at both situations.

When the source node S finally receives the trusted RREP message, it will go on transmitting data packages along this trusted route path. If the verification fails finally or the time has used up, S will issue a RREQ again. Thus the whole procedure of S performing routing discovery has finished.

4.4.4 Trust Updating Algorithm

Nodes in ad hoc networks can become malicious or invalid ones due to being compromised or the instability of underlying wireless links. So the opinions among nodes

are dynamic and must be updated frequently. In this subsection we will propose a trust updating algorithm for nodes in ad hoc networks to update their opinions towards other nodes' trustworthiness.

We logically divide each element in a opinion into five levels. The interval between two levels is 0.25. That is, the belief value will have such five logical levels: 0, 0.25, 0.5, 0.75, 1. And so are the disbelief and uncertainty values.

Each time after a node finishes verifying another node's trustworthiness or authenticity, the node will keep an record of successful and failed verification times in its trust table. If node B's successful times in A's trust table have reached 10, A's opinion about B will be updated according to Equation 27. Node that the sum of belief, disbelief and uncertainty is 1 as described in our trust model.

$$belief+ = 0.25,\ disbelief- = 0.125,\ uncertainty- = 0.125 \tag{27}$$

If node B fails to pass the verification by A or there is a link layer failure with related to B, A's opinion about B will be updated according to Equation 28. Also the failed verification time will be increased by 1.

$$belief- = 0.125,\ disbelief+ = 0.25,\ uncertainty- = 0.125 \tag{28}$$

If now A's belief about B is reduced below 0 or node B's failed verification times in A's trust table increase to 2, node B's opinion by A will become $(0, 1, 0)$ and then A will broadcast this totally disbelief opinion about B to its neighbors using trust recommendation protocol. The neighbors of A will re-calculate the opinion about B using trust combination algorithm.

Each entry in a node's trust table has a expiry time. If there is no any verification or link layer failure during the expiry time, the node will update the corresponding opinion according Equation 29.

$$belief- = 0.375,\ disbelief+ = 0.125,\ uncertainty+ = 0.25 \tag{29}$$

For those nodes' opinions has been deemed as $(0, 1, 0)$, after one or more expiry times these opinions will be updated to $(0, 0, 1)$ in terms of the harm these nodes have introduced.

The trust table can also be updated by the monitor mechanism. If the monitor mechanism detects that a node has some abnormal or malicious behaviors, the corresponding entry in a node's routing table will be updated according to Equation 28. Actually, the monitor mechanism plays an important role to detect the abnormal and abuse behaviors of the network. Bad nodes will finally be detected and denied from the network.

4.4.5 Key Management using Trust Model

There are some self-organized key management systems that have been proposed before, such as [4], [44], [16], [23], [27].

The most interesting idea used in the above literatures is that they all use several nodes, say k, to collaboratively provide authentication services. However, this kind of threshold cryptography has a limitation that it needs at least k neighbors to issue a certificate of public key or do authentication.

The idea of trust model can be used into key management subsystem to provide a more flexible and self-organized scheme. The most important contribution is that by evaluating the trustworthiness of neighbors one node can also issue the certificate of a public key in the situation of less than k neighbors.

4.4.6 Analysis

By introducing the idea of trust model into our design, we are able to establish a more flexible and less overload secure routing protocol for ad hoc networks.

From performance point of view, our self-organized trusted routing protocol decreases the overhead of the whole network. This design does not need to perform cryptographic computations in every packet, which will cause huge time and performance consumption. At the beginning of establishing the trust relationships among nodes, cryptographic computations are needed. After the trust relationships have formed a scale the subsequent routing operations can be performed using trust information. Therefore, the whole overhead of performing routing operations will be greatly decreased with the help of trust information.

From security point of view, our design will detect the misbehavior finally and reduce the harms to the minimum extent. When a good node in the ad hoc network is compromised and becomes a bad one, its misbehavior will be detected by the monitor mechanism. Then with the help of trust update algorithm, the opinions from the other nodes to this node will be updated shortly. Thus this node will be denied access to the network. Also a previous bad node can become a good one if the attacker leaves or the underlying links are recovered. In this situation, our design allows this node's opinion from other nodes' points of view to be updated from $(0, 1, 0)$ to $(0, 0, 1)$ after one or several expiry time units.

5 Research Plan and Future Work

In the future we firstly plan to improve our trust model so that we can express the objective belief about the other nodes' trustworthiness in the situation of insufficient evidence. Because of the high mobility of nodes in ad hoc networks, it is very common that a node cannot collect enough trust information about another node. Therefore, we should design a optimized trust information combination algorithm to help the node to make a wise decision.

Next we will focus our work on the design of a secure, efficient and flexible routing protocol for ad hoc networks in details. Also, trust model can be used into key management subsystem. We will try to combine cryptography and trust model together to design a more flexible key management subsystem that can be used for the secure routing protocol in ad hoc networks

A detailed simulation evaluation will be conducted in terms of message overhead, security analysis, and tolerance to mobile attackers and so on. The most popular network simulation tools are NS-2, Glomosim and so on. We prefer to choose NS-2 for our simulation. Then we should compare our design with several recent secure protocols.

6 Conclusion

It is a promising direction to apply the idea of trust model into the security solutions of ad hoc networks. One fundamental challenge for the security design in mobile ad hoc networks is that such networks do not possess any pre-existing infrastructure support. Therefore, the security solution must be provided in a decentralized or self-organized way. Some previous work for the security of ad hoc networks often assumes that there are some priori third-parties or certificate authorities to provide the assurance of information security. These solutions destroy the nature of ad hoc networks.

In our security design, we introduce trust model to represent the beliefs and trust relationships between the nodes in ad hoc networks. The beliefs about other nodes' trustworthiness are denoted by *Opinion* in our model. Nodes can cooperate together to obtain a relative objective opinion about one node's trustworthiness. By using trust model into the design of secure routing protocol of ad hoc networks and key management subsystem of such networks, we can flexibly choose whether and how to perform cryptography issues to decrease the computational overhead that brought by the encryption, decryption or digital signatures. This kind of solution may be a more light-weight and flexible scheme than other cryptography and authentication designs.

Furthermore, the concept of opinion is similar as human's perceive, which make it feasible to make a judgement in the situation of without enough trust information or evidence.

References

[1] A. Abdul-Rahman and S. Halles. A distributed trust model. In *new Security Paradigms Workshop '97*, pages 48–60, 1997.

[2] B. Barber. *Logic and limits of Trust*. New Jersey: Rutgers University Press, 1983.

[3] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proceedings of the European Symposium on Research in Computer Security*, pages 3–18, Brighton, UK, 1994. Springer-Verlag.

[4] S. Capkun, L. Buttyan, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. Technical report, Swiss Federal Institute of Technology Lausanne (EPFL) Tech., 2002.

[5] T. Chiueh. Optimization of fuzzy logic inference architecture. *IEEE Computer*, pages 67–71, 1992.

[6] B. Christianson and W. S. Harbison. Why isn't trust transitive? In *Security Protocols International Workshop*, pages 171–176. University of Cambridge, 1996.

[7] P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, volume 4, pages 49–72. Department of Sociology, University of Oxford, electronic edition, 2000.

[8] M. Deutsch. Cooperation and trust: Some theoretical notes. In M. R. Jones, editor, *Nebraska symposium on motivation*, pages 275–319. University of Nebraska Press, 1962.

[9] M. Deutsch. *The Resolution of Conflict: Constructive and Destructive Processes*. New Haven, CT: Yale University, 1972.

[10] D. Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. Department of Sociology, University of Oxford, electronic edition, 2000.

[11] D. Gambetta, editor. *Trust: Making and Breaking Cooperative Relations*. Department of Sociology, University of Oxford, electronic edition, 2000.

[12] M. Ginsgerg. Non-monotonic reasoning using dempster's rule. In *Proc. of the AAAI-84*, pages 125–129, 1984.

[13] D. Good. Individuals, interpersonal relations, and trust. In D. GAMBETTA, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 3, pages 31–48. Basil Blackwell, 1988.

[14] T. Grandison. Trust specification and analysis for internet applications. Technical report, Imperial College of Science, Thechnology and Medicine, Department of Computing, 2001.

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking* (MobiCom 2002), sep 2002.

[16] J.-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of MobiHOC01: ACM Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2001.

[17] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). Internet Draft, Apr 2003.

[18] A. Josang. Artificial reasoning with subjective logic. In *2nd Australian Workshop on Commonsense Reasoning*, 1997. http://www.idt.ntnu.no/ ajos/papers.html.

[19] A. Josang. Prospectives for modelling trust in information security. In *Australasian Conference on Information Security and Privacy*, pages 2–13, 1997.

[20] A. Josang. A subjective metric of authentication. In *ESORICS: European Symposium on Research in Computer Security*. LNCS, Springer-Verlag, 1998.

[21] A. Josang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, 2001.

[22] E. K. Koc. etacom 1996 emerging technologies applications in communications wireless security implementation.

[23] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *IEEE ICNP*, 2001.

[24] P. Lamsal. Understanding trust and security, 2001.

[25] G. Lowe. An attack on the needham-schroeder public key protocol. *Information Processing Letters*, 56:131–133, 1995.

[26] N. Luhmann. *Trust and Power Chichester*. John Wiley and Sons, 1979.

[27] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing ad hoc wireless networks. In *IEEE ISCC02*, 2002.

[28] D. W. Manchala. Trust metrics, models and protocols for electronic commerce transactions. In *The 18th International Conference on Distributed Computing Systems*, page 312, 1998.

[29] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, UK, 1994.

[30] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, 2000.

[31] F. L. Mayer. A brief comparison of two different environmental guidelines for determining "levels of trust". In *Sixth Annual Computer Security Applications Conference*, 1990.

[32] C. E. Perkins and E. M. Belding-Royer. Ad hoc on-demand distance vector (aodv) routing. Internet Draft, Feb 2003.

[33] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244. ACM Press, 1994.

[34] D. Povey. Trust management for e business. *E Business Security Conference. Brisbane, Australia*, 1999.

[35] G. Shafer. *A Methematical Theory of Evidence*. Princeton University Press, 1976.

[36] Y. Teng, V. V. Phoha, and B. Choi. Design of trust metrics based on dempster-shafer theory.

[37] M. S. Tyrone Grandison. A survey of trust in internet applications. *IEEE Communications Surveys, Fourth Quarter*, 2000.

[38] R. Yahalom, B. Klein, and T. Beth. Trust relationships in secure systems - a distributed authentication perspective. In *RSP: IEEE Computer Society Symposium on Research in Security and Privacy*, pages 150–164, 1993.

[39] R. Yahalom, B. Klein, and T. Beth. Trust-based navigation in distributed systems. *Special issue "Security and Integrity of Open Systems of the jounal "Computing Systems"*, 1994.

[40] H. Yang, X. Meng, and S. Lu. Self-organized network-layer security in mobile ad hoc networks. In *ACM MOBICOM Wireless Security Workshop (WiSe'02), Atlanta*, Sep 2002.

[41] L. Zadeh78. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[42] M. G. Zapata and N. Asokan. Securing ad hoc routing protocols. In *Proceedings of the ACM workshop on Wireless security*, pages 1–10. ACM Press, 2002.

[43] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the sixth annual international conference on Mobile computing and networking*, pages 275–283. ACM Press, 2000.

[44] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Networks*, 13(6):24–30, 1999.