Promoting Effective Exchanges Between Vehicular Agents in Traffic Through Transportation-Oriented Trust Modeling

Umar Farooq Minhas[†] Jie Zhang[†] Thomas Tran[‡] Robin Cohen[†]
[†]David R. Cheriton School of Computer Science, University of Waterloo, Canada
{ufminhas, j44zhang,rcohen@uwaterloo.ca}
[‡]School of Information Technology and Engineering, University of Ottawa, Canada
{ttran@site.uottawa.ca}

ABSTRACT

In this paper, we focus on the problem of enabling vehicles in mobile vehicular ad-hoc networks (VANETs) to exchange information about traffic and road conditions in a way that makes it possible for each agent to assess the trustworthiness of the reports received. In particular, we develop a multi-faceted trust modeling framework that is designed specifically for VANET contexts, providing for trust modeling that includes reasoning about time and location and about agent roles, as part of the overall processing. We demonstrate the value of our trust modeling framework through simulated traffic environments, clarifying the importance of distinct elements of our multi-faceted model. In addition, we comment on the value of our chosen simulation environment towards future research to support more effective agent exchanges in VANETs.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Intelligent agents, Multiagent systems; C.2.0 [**Computer Communication Networks**]: General – Security and protection; I.6.m [**Computing Methodologies**]: Simulation and Modeling

General Terms

Algorithms, Design, Security, Experimentation

Keywords

Trust modeling, VANET, Traffic management, Information exchange

1. INTRODUCTION

With the advancement in technology more and more vehicles are being equipped with GPS and Wi-Fi devices that enable them to communicate with each other, creating a vehicular ad-hoc network (VANET). Various studies have established the fact that the number of lives lost in motor vehicle crashes world-wide every year is by far the highest among all the categories of accidental deaths [1]. It is apparent that there is a dire need to enhance passenger and road safety which is precisely one of the goals of deploying vehicle to vehicle (V2V) communication systems. Another supporting goal is to be able to effectively route traffic through dense urban areas by disseminating up to date information regarding road condition through the VANET.

Some car manufacturers have already started to fit devices that will help achieve the goals mentioned above. For example, GM has rolled out V2V communication in its Cadillac STS Sedans. GM's proprietary algorithm called "threat assessment algorithm" keeps track of the relative position, speed and course of other cars (also equipped with V2V technology) in a quarter-mile radius and issues a warning to the driver when a crash is imminent [6]. Similar prototypes by other car manufacturers are currently in the testing phase, scheduled to hit the markets over the coming years.

Even though the initial algorithms and protocols that are being proposed by the car manufacturers are proprietary, it is believed that the standardization efforts carried out by Car-2-Car Consortium [25] will help to define a common interface for V2V communication technologies allowing its wide-spread use. Following this, it is very natural to assume that agent applications will be deployed whose main goal will be to assist the user in various ways using V2V communication. One such example is of an agent that gathers road congestion information and calculates the optimal route from a user's origin to destination thus bringing utility to the user. In such a scenario, we can view cars in a VANET as autonomous agents acting on behalf of their owners thus constituting a multi-agent network.

The agent would represent the motives of car owners who might as well decide to behave selfishly every now and then. For example, consider a user who instructs his agent to report the roads on his path as congested with the hope that other agents would avoid using these roads, thus clearing the path. Therefore one important issue among others that may arise in VANETs is the notion of trust among different agents. The goal of incorporating trust is to give incentives for these agents to behave honestly and to discourage self-interested behavior. These details are captured through what is called a *trust model*. Defined formally, "trust is a belief an agent has that the other party *will do what it says it will* (being honest or reliable) or *reciprocate* (being reciprocative for the common good of both), given an opportunity to defect to get higher payoffs" [17]. Here it is important to clarify that our notion of trust always refers to the trust placed by one agent in another agent which is different from the trust placed by the user (or driver) in the agent itself and is beyond the scope of this work. A closely related notion called reputation is defined as the opinion or view of an agent about another agent that is either directly acquired from the environment or from other agents and ultimately leads to building of trust [17]. Given the critical nature of agent applications in the context of VANETs, it is crucial to associate trust with agents and the data that they spread.

With respect to the general topic area of agents in traffic and transportation, our research can be characterized most appropriately as focusing on the challenge of enabling autonomous vehicles to engage in collaborative driving and in intelligent peer-to-peer interactions to enable distributed decision making in traffic.

1.1 The challenges of VANET trust modeling

Modeling trustworthiness of agents in VANETs presents some unique challenges. First of all, the agents in a VANET are constantly roaming around and are highly dynamic. On a typical high-

way the average speed of a vehicle is about 100 kilometers an hour. At high speeds the time to react to an imminent situation is very critical [2], therefore, it is very important for the agents to be able to verify/trust incoming information in *real-time*. Second, the number of agents in VANET can become very large. For example, in dense urban areas the average amount of vehicles that pass through the network may be on the order of millions and several thousand vehicles will be expected to be present in the network at any given time. Also this situation is exacerbated during the rush hours when, for example, majority of the people commute to and back from work in a metropolitan area. This may introduce several issues some of which include network congestion - since vehicles are communicating on a shared channel, information overload - resulting from vehicles receiving a lot of data from the near-by vehicles in a congested area etc. Hence there will be a need to have intelligent vehicle communication systems that are *scalable* and can detect and respond to these potentially hazardous situations by effectively deciding with which agents to communicate [11].

Another key challenge in modeling trust in a VANET environment is that a VANET is a *decentralized*, open system i.e. there is no centralized infrastructure and agents may join and leave the network any time respectively. If an agent is interacting with a vehicle now, it is not guaranteed to interact with the same vehicle in the future [5]. Therefore, it is not possible to rely on mechanisms that require a centralized system (e.g. the Centralized Certification Authority and the Trusted Third Party etc) or social networks to build long-term relationships.

Also, information about road condition is rapidly changing in VANET environments, e.g. a road might be busy 5 minutes ago but now it is free, making it hard to detect if the agent spreading such information is malicious or not. This also brings out an important challenge that the information received from VANETs needs to be evaluated in a particular context. The two key context elements in VANETs are *location* and *time*. Information which is closer in time and location of an event is of more relevance. We explain this in more detail in Section 2.

Various trust and reputation models (e.g. [20] and [30]) have been studied with reference to multi-agent environments, however, given the unique characteristics of agents in VANETs the existing models cannot be applied directly. For example, several trust and reputation models are built around the assumption that the agents can have multiple direct interactions with other agents and hence they fail when applied to VANETs, since the interactions between agents in this environment may be quite sparse.

The main goal of this work is then to develop a framework that can effectively model the trustworthiness of the agents of other vehicles in VANETs. We propose a novel multi-faceted approach for modeling trust in VANET environments that incorporates role-based trust, experience-based trust, priority-based trust and majority-based trust and that is able to restrict the number of reports that are received from other agents. Our expanded trust model is aimed to be decentralized, location/time specific, event/task specific, able to cope with the data sparsity problem, cumulative in order to be scalable, sensitive to privacy concerns, and able to support system-level security. We present the design of this model in detail, clarifying how it meets various critical challenges for trust modeling in VANET environments. We also step through a detailed procedure of computing trustworthiness of agents and generating effective responses to information sent by those agents. We finally demonstrate its value in a simulated vehicular setting. The result is an important first step towards the delivery of effective intelligent vehicular communication, one that is sensitive to the trustworthiness of the vehicular agents.

As will be seen, we introduce a framework that is amenable to dynamically changing networks of agents (a desirable quality for VANETs, as explained in [19]), in contrast with other trust models that are designed to operate in more stable environments (e.g. [27]) or that assume complete knowledge of all the agents in the system (e.g. [15]). In addition, our inclusion of roles as part of the trust modeling framework can be seen as an element to overcome the more typical sparsity of relationships which compromises an approach to trust modeling relying solely on social networks (e.g. [29]).

2. EXPANDED TRUST MANAGEMENT

From the discussion in previous sections, it becomes apparent that no single trust or reputation mechanism can work particularly well for the challenge of modeling trust effectively for VANET environments. Instead of just having one or two trust metrics for evaluating trust, there is a need to have several different trust metrics with various key properties in order to capture the complexity that arises between interacting agents in VANET. We propose that in order to derive a rather complete and comprehensive view of trust for agents in VANET we will need to integrate security solutions (at the system level) for trust management, i.e. secure storage of role identities for role-based trust in our proposal.

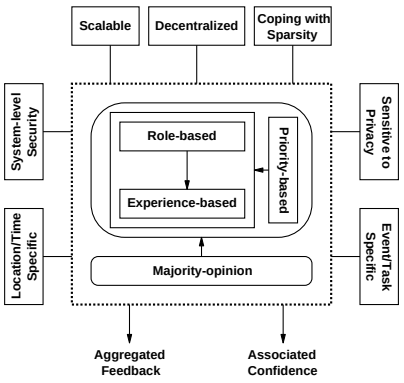

Figure 1: Expanded Trust Management

Figure 1 illustrates the design of our expanded trust management. The core of the management is grouped by the dashed rectangle in the middle. This core consists of two parts. One part maintains trustworthiness of agents in order for trusted agents (advisors) to be chosen to ask for their feedback. More specifically, in this part, the trustworthiness of agents is modeled based on role-based trust and experience-based trust, which are both combined into the priority-based model that can be used to choose proper advisors.

Our role-based trust exploits certain predefined roles that are enabled through the identification of agents (vehicles). For example, agents can put more trust in certain agents as compared to others, i.e. agents identified as law enforcing authorities or owned by government [19]. Our experience-based trust represents a component of trust that is based on direct interactions. It is in the same spirit of incorporating evidence from direct interactions into trust calculation through Interaction Trust as proposed by [8] or the Individual Dimension of trust in the model as proposed by [23]. Implementation and formalization of these two trust metrics will be presented in Section 2.2.

The other part of the core is a majority-opinion approach to aggregate feedback from selected advisors. Detailed procedures for these processes will be further discussed in Section 2.2. More im-

portantly, our management of trust has several key properties represented by rectangles around the core in the figure. Our trust management is aimed to be decentralized, location/time specific, event/task specific, able to cope with the data sparsity problem, cumulative in order to be scalable, sensitive to privacy concerns, and able to support system-level security. These properties will be extensively discussed in Section 2.1, respectively. Note that the property of system-level security is mentioned in different places where we discuss other properties and our model, i.e. secure storage of role identities in Section 2.1.1, verification of time/location of reported events in Section 2.1.3, and identification of agents' roles in Section 2.2.3.

The outcome of our trust management is aggregated feedback for a certain request/event and an associated confidence value for it. The aggregated feedback is eventually affected more heavily by highly trusted advisors. The value of confidence would depend on the reliability of estimated experience-based trust of each other agent and the maximum accept error rate for the aggregated feedback. In general, a higher value of confidence, i.e. a value closer to 1, would result from considering more evidence or metrics having high reliability, for a fixed error rate. We can view confidence as a parameter that adds another dimensionality to the output generated by the model allowing the agent applications to have a richer notion of trust and finally decide how to react on the reported event. Our notion of confidence is somewhat tantamount to the notion proposed in [24, 8].

2.1 Key Properties

We provide here detailed discussion of the seven key properties that our trust management incorporates. These properties guide our design of the expanded trust management, which can be applied to the problem of trust management in VANET.

2.1.1 Decentralized Trust Establishment

Models which depend on a central entity for the reliable establishment of trust are not appropriate for the domain of VANET because of its highly distributed property. Therefore, we propose that trust establishment should be fully decentralized to be applicable to the highly dynamic and distributed environment of VANETs.

Our experience-based trust model makes use of agents' direct interactions to update one agent's belief in the trustworthiness of another. This one-to-one interaction can easily be implemented in a distributed manner. Our role-based trust can also be done in a totally decentralized manner among the vehicles themselves. For this to work, we may involve the car manufacturers, or transportation authorities to issue certificates at the manufacture or registration time respectively. A public-private key infrastructure for verifying each other's roles can be implemented in a distributed manner. Also there would be a need to store these certificates and keys in a way that they cannot be manipulated or tampered with, to archive high security. To this end, researchers [16] who have done studies with the goal of securing VANET communications have unanimously proposed the use of a tamper proof device that stores e.g. the cryptographic keys issued by authorities. If any attempt to tamper the device is made, the keys are destroyed automatically stripping the agent from its ability to communicate with other agents thus effectively destroying its means of deriving any utility at all.

2.1.2 Coping with Sparsity

Effective trust establishment should not be contingent upon a minimum threshold for direct interactions. As we have described at several places, it should not be expected that an agent in VANET would possibly interact with the same agent more than once. However, it is important to clarify here that the trust models should still be able to effectively take into consideration any data available from direct interaction (even though it might happen just once). Thus, in a scenario where the number of agents that are able to spread information has gone down to the extent that the condition of information scarcity or a total lack of information is prevalent, any data might be termed valuable. In the trust calculation, the weight for the data can be raised in this scenario while it may have a lower default value, to cope with the data sparsity problem in VANET.

We also have the role-based trust approach to distinguish trustworthy agents from untrustworthy ones to some extent. When an experience-based trust approach is used, we also introduce the idea of allowing agents to send testing requests, to deal with sparsity. The senders of these testing requests basically know the solution to these requests in advance. Imaging a group of agents driving in a city from one location to another, they remain in contact range for a certain period of time. These agents can send testing requests to each other and evaluate their feedback. Trust between them can then be established through the experience-based trust in our management model.

2.1.3 Event/Task and Location/Time Specific

Since the environment of the agents in VANET is changing constantly and rapidly, a good trust model should introduce certain dynamic trust metrics, capturing this dynamism by allowing an agent to control trust management depending on the situation at hand [19, 4]. Here, we separately deal with two particularly important dynamic factors in the context of VANETs, event/task and location/time.

Agents in general can report data regarding different events e.g. car crashes, collision warnings, weather conditions and information regarding constructions etc. Our trust management should therefore be event/task specific. For example, some of these tasks may be time sensitive and require quick reaction from the agent that receives them. In this case, this agent can only consult a very limited number of other agents to verify whether the reported information is true. In another case, reporting agents having different roles in VANET may have more or less knowledge in different types of tasks. For example, a police may know more about car crash information while city authorities may know more about road construction information. Thus, our role-based trust should be task specific. One way to implement this in our role-based trust model is to have a set of events associated with a set of roles of agents (e.g. law enforcement, municipal authorities). This information can be used later for an agent to choose particular other agents to consult regarding a particular event. Our experience-based trust is also event specific. An agent updates the reporting agent's trust by taking into account the type of the reported event. For example, life-critical events will certainly have more impact on the reporting agent's trust.

We also note that location and time are another two particularly important dynamic metrics. For example, if the origin of a certain message is closer to the location of where the reported event has taken place, it might be given a higher weight, relying on the underlying assumption that an agent closer to the event is likely to report more realistic data about the event (given that they are not malicious themselves). Similarly, we can apply this concept to time. If the message reporting a certain event is received closer to the time when the reported event has taken place, it might be allowed a higher weight in trust calculation. Another suggestion that naturally follows from time based trust is that, since the relevance of data in VANET is highly dependent on when it was received,

it would make sense to assign a decay factor to the message. The message further away from the time of evaluating trust would be assigned a lower weight. In other words, we should decay the impact of message relative to the time of the trust evaluation. The decay factor is somewhat analogous to the time-to-live (TTL) field used in IP packets.

The first issue that may arise with calculating time or location specific trust is how to get location and time of the actual event. We expect that whenever a report regarding an event is generated to be shared among other agents it will hint to the time at which this event has taken place, giving us the required time information. Also we assume that every agent while transmitting the report appends its location with the report. The next issue is to verify whether the time and location information contained within a report is real or spoofed. With this regard, [7] has proposed a method to accurately estimate the location of nearby agents. However, complete treatment of this issue is beyond the scope of this work. Now the next task would be to actually use the location/time information in trust management. In the calculation of subjective reputation as proposed by [23] they use a weighted sum of trust values suggesting that the weights should be adjusted such that higher weights are assigned to the agents closer to the agent which is calculating trust. In a similar fashion, we can extend their model by instead of defining the closeness between agents; we define the location closeness between the actual event and the agent reporting this event. For the time based trust a similar calculation can be done by modifying the notion of time closeness as that between the time when the event has taken place and that of receiving the report.

2.1.4 Scalable

Scalability is an important aspect in trust management in VANET environments. In our system, each agent consults only a number of other trusted agents. This number can be fixed or slightly updated with the changes in, for example, VANET size or the task at hand. However, it is always set to a value small enough to account for scalability.

Establishing trust in VANETs using experience-based trust requires each agent to store the history of past interactions with other agents and to compute their trust based on that information. For the purpose of being scalable, our experience-based trust model updates agents' trustworthiness by accumulatively aggregating agents' past interactions in a recursive manner, similar to [10]. The computation of our experience-based trust is thus linear with respect to the number of interactions. And only the most recent trust values are needed to be stored and used for computation. This design makes our trust management scalable.

2.1.5 Sensitive to Privacy Concerns

Privacy is an important concern in a VANET environment. In this environment, the revealing of a vehicle owner's identity (e.g. the owner's home address) may allow a possibly malicious party to cause damage to the owner. Our trust management makes use of a public key infrastructure (PKI) allowing agents to authenticate each other. In our system, when an agent sends a report to another agent, the sender needs to authenticate itself to the receiver that it has a certain role. Although these keys do not contain any sensitive identities of the sender, the receiver may be able to track them by logging the messages containing the key of the sender. For example, the receiver can track the likely home address of the sender by finding out the route of the sender if the receiver has sufficient information about different locations that the sender has been to, and therefore other identities. This issue can be addressed by changing keys, as suggested in [18]. Each agent in VANET will store a large set of pre-generated keys and certificates. It will change keys while sending information to others regarding some privacy sensitive locations of the sender (i.e. places nearby home), so that others do not recognize this sender as one of the previous senders that they have interacted with. In this way, others will not be able to discover the sender's privacy sensitive identities, while they will still be able to keep track of experience with this sender regarding some insensitive locations of the sender.

2.2 Computation Procedure

In this section, we briefly outline the procedure taken by an agent to make a decision for a (requested) task/event by aggregating reports about this task from other trusted agents and to update their experience-based trust values afterwards.

2.2.1 Scenarios

An agent in a VANET environment may actively send a request to a list of trusted neighboring agents about a task, i.e. weather or direction information. In another scenario, it may passively wait for other agents to send reports about an event, i.e traffic or collision ahead of the agent. Once it receives a report about an event from another agent, it may trust the information if it has high confidence that the report sender can be trusted. Otherwise, it may need to verify (double check) if the information given by the sender is reliable by asking other trusted agents. In both scenarios, the agent will need to aggregate senders' reports. Values calculated in this manner can then be used by the agent to decide whether to believe a particular report and take corresponding actions. For this purpose, each agent in our system keeps track of a list of other agents. This agent updates all report senders' trustworthiness after the truth of their reported events is revealed. The above two processes of aggregating reports and updating trust will take into account the context in general, this agent's notion of which other agents it is interacting with, the notion of which group the other agents belong to or the roles assigned to the other agents, the time of reported event together with the time of message arrival, the relative locations of the other agents, and the actual contents of the message to evaluate task/event specific trust etc. Next, we provide detailed description and formalization of each step in our computation procedure.

2.2.2 Computation Steps

Four elements are incorporated into our overall trust management as its core, shown in Figure 1: 1) Experience-based trust; 2) Role-based trust; 3) Majority opinion (or social network of trust); 4) Priority-based trust. Our computation procedure consists of four steps.

Step 1: Depending on the task at hand, set a value n = number of agents whose advice will be considered. This incorporates task-based trust. For example, if you need a very quick reply, you may limit $n = 2$ or 3; if you are planning ahead and have time to process responses, n could potentially be larger.

Step 2: Using n, construct an ordered list of agents to ask. The list will be partitioned into groups as follows:

$$\begin{bmatrix} G_1: & a_{11}, & a_{12}, & a_{13}, & ..., & a_{1k} \\ G_2: & a_{21}, & a_{22}, & a_{23}, & ..., & a_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ G_j: & a_{j1}, & a_{j2}, & a_{j3}, & ..., & a_{jk} \end{bmatrix}$$

where $kj = n$.[1] This priority list is ordered from higher roles to lower roles, for example, G_1 being the highest role. Within each

[1]There is no need for each group to have the same number of elements. We provide here only an simplified example.

group of agents of similar roles, the group is ordered from higher (experience-based) ratings to lower ratings. Thus, a_{ij} represents the agent in role class i that is at the j^{th} level of experience, relative to other agents at that level. Hence, role-based trust and experience-based trust are combined into this priority-based approach. These two trust metrics will be further discussed later in this section.

Step 3: When an agent requires advice, the procedure is to ask the first n agents the question, receive the responses and then perform some majority-based trust measurement.

Step 3B: The processing of the responses is as follows: if there is a majority consensus on the response, up to some tolerance that is set by the asker (e.g. I want at most 30% of the responders to disagree), then this response is taken as the advice and is followed. We will formalize this majority-based trust in Section 2.2.5.

Step 3C: Once this advice is followed, the agent evaluates whether this advice was reliable and if so, personal experience trust values of those agents are increased; if not, personal experience trust values of those agents are decreased. Detailed formalization of this process will be given in Section 2.2.4.

Step 3D: If a majority consensus cannot be reached, then requiring majority consensus for advice is abandoned. Instead, the agent relies on role-based trust and experience-based trust (e.g., taking the advice from the agent with highest role and highest experience trust value)[2].

Step 4: In order to eventually admit new agents into consideration, when advice is sought, the agent will ask a certain number of agents beyond agent a_n in the list. The responses here will not count towards the final decision, but will be scrutinized in order to update personal experience trust values and some of these agents may make it into the top n list, in this way.

Algorithm 1 is a pseudo code summary of the proposed algorithm. Note that this pseudo code covers the main scenario where an agent actively requests other agents for advice and does not include the exploration/testing step (Step 4).

Algorithm 1: Computation Steps

```
while on the road do
    if in need of advice then
        Choose n; //number of agents to ask for advice
        //according to roles and experience
        Prioritize n agents;
        Send request and receive responses;
        if response consensus > acceptable ratio then
            Follow advice in response;
        else
            Follow advice of agent with highest role and
            highest trust value;
    Verify reliability of advice;
    Update agents' trust values;
```

2.2.3 Role-based Trust

Our role-based trust exploits certain predefined roles assigned to all agents in the system. The underlying assumption here is that the agents identified by authorities are more closely monitored and are expected to behave in a certain way. We can also conceptualize roles as an expected behavior of a certain group or class of agents where all the agents belonging to a group would behave similarly. We propose a role-based approach because the expected number of possible roles and the rules to assign these roles would be very few in the domain of VANETs and thus can be manually managed and/or updated by a trusted authority. Note that the concept of seniority (expertise in a certain context/task, for instance) could be incorporated into role-based trust, as mentioned in Section 2.1.3.

To demonstrate our role-based approach, let's consider a simple system that recognizes the following four different roles listed in decreasing order[3], i.e. from the highest role to the lowest one: 1) authority, 2) expert, 3) seniority, and 4) ordinary. Each role level may also be associated with a trust value $T_r \in (0, 1)$ where higher level roles have larger T_r values. The rules for assigning and authenticating these roles can be structured as follows:

1. Agents representing authorities such as traffic patrols, law enforcement, state or municipal police etc. assume the authority role.

2. Agents specialized in road condition related issues such as media (TV, radio or newspaper) traffic reporters, government licensed and certified instructors of driving school etc. receive the expert role.

3. Agents familiar with the traffic or road conditions of the area in consideration, e.g. local people who commute to work on certain roads or highways or have many years of driving experience with a good driving record (e.g. taxi drivers), are given the seniority role.

4. All other agents are considered having the ordinary role.

All agents should possess certificates issued by a trusted certificate authority for authentication purpose. Note that we need a way for an agent to tell if another agent is indeed having the role that he is claiming to have. One possible solution to this problem is to make use of public-key certificates in an asymmetric cryptosystem as follows: Each agent should have a public key certificate, which can simply be a document containing the agent's name, his role and his public key. That document is signed by a trusted certificate authority (with the certificate authority's private key) to become the agent's public key certificate. Everyone can verify the authority's signature by using the authority's public key. Now, when agent A sends a message to agent B, A must sign the message with his private key. B then can verify (using A's public key) that the message was truly sent by A.

2.2.4 Experience-based Trust

We track experience-based trust for all agents in the system, which is updated over time, depending on the agent's satisfaction with the advice given, when asked. As mentioned in the previous section, our experience-based trust is cumulative in the sense that it updates agents' trust recursively. Thus, only the most recent trust values and the number of interactions between agents are needed to be stored in the system, to make the system scalable. We here formalize the computation of this trust. If we define the range of all personal experience trust values to be the interval $(-1, 1)$, where 1 represents absolute trust and -1 represents absolute distrust, then

[2]Note that an additional motive for modeling the trustworthiness of a variety of agents is to be able to learn about these agents for future interactions, for example in the calculations of experience-based trust and majority-opinion trust.

[3]Our experience-based trust may be helpful for role categorization. When agents have sufficient experience-based trust information about each other, they may report this information to a trusted authority (i.e. the transportation department of government). A mapping between agents' real-world profiles and their trustworthiness can then be derived for helping categorize their roles.

we can use the following scheme to update an agent's personal experience trust value[4], as suggested by [26]:

Let $T_A(B) \in (-1, 1)$ be the trust value indicating the extent to which agent A trusts (or distrusts) agent B according to A's personal experience in interacting with B. After A follows an advice of B, if the advice is evaluated as reliable, then the trust value $T_A(B)$ is increased by

$$T_A(B) \leftarrow \begin{cases} T_A(B) + \alpha(1 - T_A(B)) & \text{if } T_A(B) \geq 0, \\ T_A(B) + \alpha(1 + T_A(B)) & \text{if } T_A(B) < 0, \end{cases} \quad (1)$$

where $0 < \alpha < 1$ is a positive increment factor.

Otherwise, if B's advice is evaluated as unreliable, then $T_A(B)$ is decreased by

$$T_A(B) \leftarrow \begin{cases} T_A(B) + \beta(1 - T_A(B)) & \text{if } T_A(B) \geq 0, \\ T_A(B) + \beta(1 + T_A(B)) & \text{if } T_A(B) < 0, \end{cases} \quad (2)$$

where $-1 < \beta < 0$ is a negative decrement factor.

The absolute values of α and β are dependent on several factors because of the dynamics of the environment, such as the data sparsity situation mentioned in Section 2.1.2 and the event/task specific property mentioned in Section 2.1.3. For example, when interaction data is sparse, these values should be set to be larger, giving more weights to the available data. For life-critical events (i.e. collision avoidance), $|\alpha|$ and $|\beta|$ should be larger, in order to increase or decrease trust values of reporting agents more rapidly. Also note that we may set $|\beta| > |\alpha|$ by having $|\beta| = \mu|\alpha|$ and $\mu > 1$ to implement the common assumption that trust should be difficult to build up, but easy to tear down.

We also incorporate a forgetting factor λ ($0 < \lambda < 1$) in Equations 1 and 2, allowing A to assign less weight to older interactions with B. This is to cope with the possible changes of B's behavior over time. If we define t as the time difference between the current interaction and the previous one[5], the equations then become

$$T \leftarrow \begin{cases} \lambda^t(1 - \alpha)T + \alpha & \text{if } T \geq 0, \\ \lambda^{-t}(1 + \alpha)T + \alpha & \text{if } T < 0, \end{cases} \quad (3)$$

$$T \leftarrow \begin{cases} \lambda^t(1 - \beta)T + \beta & \text{if } T \geq 0, \\ \lambda^{-t}(1 + \beta)T + \beta & \text{if } T < 0, \end{cases} \quad (4)$$

where we substitute $T_A(B)$ by T for the purpose of clarity. The trust values A has of B will increase/decrease more slowly than those in Equations 1 and 2 because older interactions between them are discounted and have less impact on the current trust values.

The number of interactions between agent A and agent B, $N_A(B)$, should also be discounted accordingly. This can also be done recursively as follows:

$$N_A(B) = \lambda^t N_A(B) + 1 \quad (5)$$

Note that the experience-based formulae are also valuable to cope with agents who try to build up trust and then deceive. Once deception is detected, trust can be torn down quite quickly.

2.2.5 Majority Opinion and Confidence

Suppose agent A in VANET receives a set of m reports $\mathcal{R} = \{R_1, R_2, ..., R_m\}$ from a set of n other agents $\mathcal{B} = \{B_1, B_2, ..., B_n\}$ regarding an event. Agent A will consider more heavily the reports sent by agents that have higher level roles and larger experience-based trust values. When performing majority-based process, we also take into account the location closeness between the reporting agent and the reported event, and the closeness between the time when the event has taken place and that of receiving the report. We define C_t (time closeness), C_l (location closeness), T_e (experience-based trust) and T_r (role-based trust). Note that all these parameters belong to the interval $(0, 1)$ except that T_e needs to be scaled to fit within this interval.

For each agent B_i ($1 \leq i \leq n$) belonging to $\mathcal{B}(R_j) \subseteq \mathcal{B}$ that report a same report $R_j \in \mathcal{R}$ ($1 \leq j \leq m$), we aggregate the effect of its report according to the above factors. The aggregated effect $E(R_j)$ from reports sent by agents in $\mathcal{B}(R_j)$ can be formulated as follows:

$$E(R_j) = \sum_{B_i \in \mathcal{B}(R_j)} \frac{T_e(B_i)T_r(B_i)}{C_t(R_j)C_l(B_i)} \quad (6)$$

In this equation, experience-based trust and role-based trust are discounted based on the two factors of time closeness and location closeness. The summation is used to provide the aggregated effect of the reporting of the agents.

Note that location closeness $C_l(B_i)$ depends only on the location of agent B_i while time closeness $C_t(R_j)$ depends on the time of receiving the report R_j. $C_t(R_j)$ can also be written as $C_t(B_i)$ because we can assume that each report is sent by an unique agent in possibly different time.

To consider the effect of all the different reports, the majority opinion is then

$$M(R_j) = \arg\max_{R_j \in \mathcal{R}} E(R_j) \quad (7)$$

which is the report that has the maximum effect, among all reports.

A majority consensus can be reached if

$$\frac{M(R_j)}{\sum_{R_j \in \mathcal{R}} E(R_j)} \geq 1 - \varepsilon \quad (8)$$

where $\varepsilon \in (0, 1)$ is set by agent A to represent the maximum error rate that A can accept. A majority consensus can be reached if the percentage of the majority opinion (the maximum effect among different reports) over all possible opinions is above the threshold set by agent A.

If the majority consensus is reached, the majority opinion is associated with a confidence measure. This measure takes into account the number of interactions taken for modeling experience-based trust values of reporting agents and the maximum accepted error rate ε. We define $N(R_j)$ as the average of the discounted number of interactions used to estimate experience-based trust values of the agents sending the majority report R_j calculated using Equation 5. Based on the Chernoff Bound theorem [14], the confidence of the majority opinion can be calculated as follows:

$$\gamma(R_j) = 1 - 2e^{-2N(R_j)\varepsilon^2} \quad (9)$$

3. EXPERIMENTAL EVALUATION

In this section, we present preliminary evaluation of our trust model. We use SWANS (Scalable Wireless Ad-hoc Network Simulator, jist.ece.cornell.edu) with STRAW (STreet RAndom Waypoint) mobility model [3]. SWANS is entirely implemented in Java and can simulate networks with potentially thousands of nodes while using incredibly small amount of memory and processing power. STRAW allows to simulate real world traffic by using real

[4]A "commuter pool" might for instance offer significant experience

[5]The value of t may be scaled within the range of $[0, 1]$. This can be achieved by setting a threshold t_{max} of the maximum time for an agent to totally forget the experience happened at the time that is t_{max} prior to the current time.

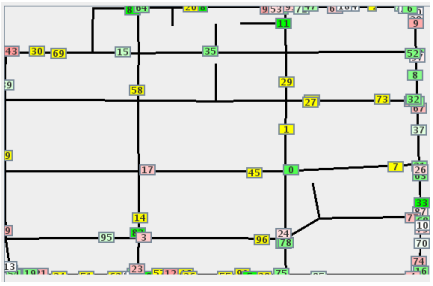

Figure 2: Simulating VANET using SWANS Simulator with STRAW Mobility Model

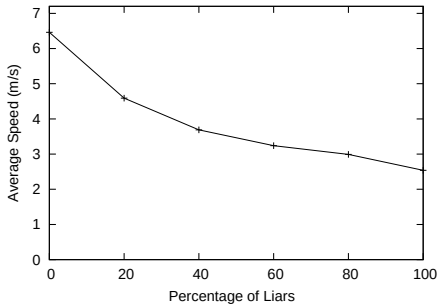

Figure 3: Average Speed of All Cars When There are Different Percentages of Liars

maps with vehicular nodes that follow rules such as speed limits, traffic signals, stop signs etc.

We use a map of north Boston, USA. Figure 2 shows a snapshot of one of our simulation runs. The bold lines are the extracted road segments from the map. The small rectangles labelled by integers represent vehicles running on the streets. For all our experiments we fix the total number of vehicles to 100 and run the simulation for a total duration of 900 seconds of simulation framework time. Note that in this paper we only experiment with the role-based and experienced-based dimensions of our trust model while leaving more comprehensive experimental evaluation for future work.

3.1 Performance Metric

One of the applications of V2V communication is to be able to route traffic effectively through the VANET and to avoid congestion or hot spots. Malicious agents in the network may send untruthful traffic information, to mislead other agents and cause traffic congestion. We measure the performance of our proposed trust model by observing to what extent it can cope with deceptive information sent by malicious agents. According to [3], we can measure congestion based on the average speed of vehicles. Lower average speed implies more traffic congestion. The performance of our model can then be measured as the increase in average speed of all agents by incorporating our model under the environment where malicious agents exist.

3.2 Results

We present experimental results to clearly show the value of different trust metrics integrated in our expanded trust management and to demonstrate that the combined one is the most effective.

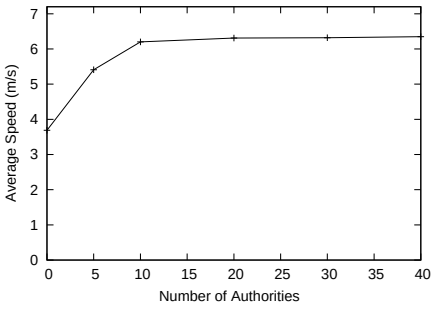

Figure 4: Average Speed of All Cars When There are Different Numbers of Authorities

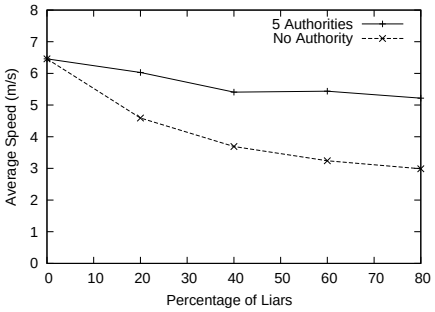

Figure 5: Average Speed of All Cars When There are Five Authorities

3.2.1 Effect of Liars on Average Speed

In our first experiment, we vary the percentage of malicious nodes in the environment and measure the change in average speed of the vehicles in the network (a measure advocated in [3]). We choose a lying strategy for the malicious nodes where they always lie about congestion on a particular road segment i.e., report congestion when there is no congestion and vice versa. We present the results in Figure 3. As expected, average speed of vehicles in the network decreases as the percentage of liars increases.

3.2.2 Countering Liars with Role-based Trust

Next we experiment with role-based trust where we introduce some agents in the environment with the role of authorities as mentioned in Section 2.2.3. In our simulation, authorities are assumed to be always trustworthy. In this experiment, we fix the number of malicious agents to be 40% and then vary the number of agents with the role of authority between 0 and 40. These results are presented in Figure 4. With an increase in the number of authorities in the environment, the overall average speed of the nodes increases, countering the effect of malicious agents. This shows the effectiveness of role-based trust in our model. From Figure 4, we can see that the average speed reaches a maximum with about 20 authorities. Figure 5 shows that even if we have a small number of agents with a role of authority in the system, we can still effectively cope with an increasing percentage of malicious nodes.

3.2.3 Countering Liars with Experience-based Trust

In this experiment, we employ only the experience-based dimension of trust. We vary the percentage of liars and measure the overall average speed of vehicles. As we can see from Figure 6, using experience-based trust results in an increase in the average speed of vehicles. This trend is consistent for all percentages of liars in

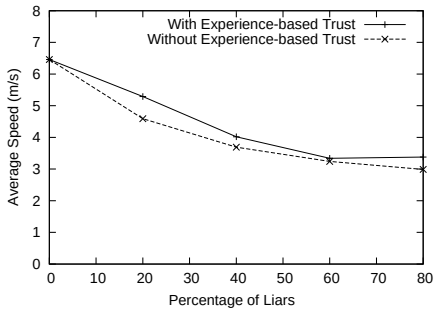

Figure 6: Average Speed of All Cars with Experience-based Trust

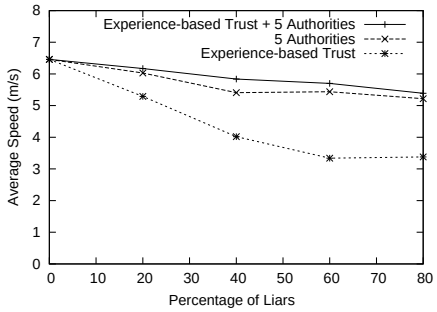

Figure 7: Average Speed of All Cars with Role-based and Experience-based Trust

the system which shows that experience-based trust is able to cope with the lying behavior of malicious agents.

Note that the performance of our trust model, namely the speed of vehicles, is averaged over the total duration of only 900 seconds of the simulation framework time. At the beginning of the simulation an agent does not yet have any experience with other agents. This explains the model's moderate performance during this early period.

3.2.4 Combining Role-based and Experience-based Trust

From Figures 5 and 6, we can see that even though experience-based trust results in an increase in the average speed of vehicles in the network with the presence of malicious agents, role-based trust does this job more effectively. In this experiment, we combine both dimensions together and measure the average speed. These results are presented in Figure 7. As we can see, by combining these two dimensions we can achieve an average speed which is higher than when using any one of these two dimensions individually. This shows that a trust model for agents in VANETs can greatly benefit by combining several dimensions of trust as proposed in this work.

3.2.5 Coping with Sparsity

This experiment is carried out to demonstrate the property of our model in coping with the data sparsity problem. In this experiment, we involve 50 nodes and run the simulation for 300 seconds of simulation framework time. We reduce the ratio of communication between nodes. The available data for modeling the trustworthiness of nodes is more sparse when the communication ratio is lower. As can be seen from Figure 8, the percentage of detecting malicious nodes decreases when the ratio of communication is reduced. By decreasing the value of β, the ability of detecting malicious nodes

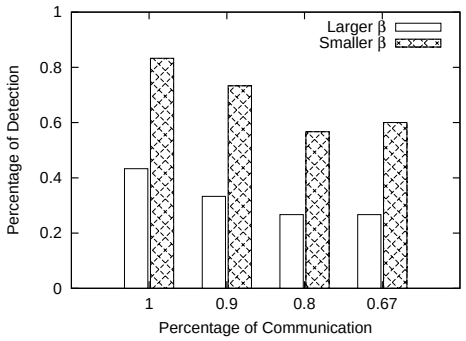

Figure 8: Coping with Sparsity

is increased dramatically[6]. This indicates that our model is able to cope with the data sparsity problem by changing the parameter β to adjust the weight of available data.

The role-based trust in our model is also able to cope with data sparsity. As shown in Figure 7, with only the experience-based trust, the performance difference of our model between more and fewer liars is large. This difference is reduced when the role-based dimension is also used. The role-based trust reduces the impact of more liars and therefore is able to begin to cope with the data sparsity problem.

4. RELATED WORK

Lin et al. [12] have investigated the benefits achieved by self-interested agents in vehicular network through simulations. They consider a scenario where agents can achieve road congestion information from other agents through Gossiping. Two different behaviors of self-interested agents are investigated 1) Agents want to maximize their own utility 2) Agents want to cause disorder in the network. Simulation results indicate that for both behaviors, self-interested agents have only limited success in achieving their goals, even if no counter measures are taken. However, the authors realize the need to take these preliminary results to more complex and potentially more damaging scenarios that may arise in VANETs. They also identify the need to establish trust in vehicular ad-hoc networks through distributed reputation mechanisms, motivating our work. In contrast to the traditional view of entity-level trust, Raya et al. [19] propose that data-centric trust may be more appropriate in the domain of Ephemeral Ad Hoc Networks such as VANETs. Data-centric trust establishment deals with evaluating the trustworthiness of the data reported by other entities rather than trust of the entities themselves. Even though there are some commonalities between our approach and theirs, for example, they also propose the use of task/event specific trust metrics as well as time and location closeness but we combine these metrics in a fundamentally different way taking the traditional view of entity-level trust instead of data-centric trust. One of the shortcomings of their work is that trust relationships in entities can never be formed, only ephemeral trust in data is established, and because this is based on a per event basis, it needs to be established again and again for every event. This will work so long as there is enough evidence either in support of or against a specific event, but in case of data sparsity we believe our model would perform better. We leave a detailed comparison between these two models for future work.

[6]The absolute value of β in Equation 2 reflects the weight placed on available data. Since $-1 < \beta < 0$, decreasing the value of β will increase its absolute value, and the weight of data will also be increased.

Dotzer [4] has suggested building a distributed reputation model that exploits a notion called opinion piggybacking where each forwarding agent (of the message regarding an event) appends its own opinion about the trustworthiness of the data. They provide an algorithm that allows an agent to generate an opinion about the data based on aggregated opinions appended to the message and various other trust metrics including direct trust, indirect trust, sender based reputation level and Geo-Situation oriented reputation level. This last trust metric allows their model to introduce some amount of dynamism in the calculation of trust by considering the relative location of the information reporting node and the receiving node. Additionally, the situation oriented reputation level allows a node to consider certain situational factors e.g. familiarity with the area, rural or metropolitan area etc. again introducing some dynamism in trust evaluation based on context. Our model has direct trust in the form of experience-based trust, indirect trust in the form of role-based trust. Furthermore, we also use location closeness in our model that is similar to Geo-Situation oriented reputation level in their model. However, we provide an algorithm to combine, for example, experience-based and role-based trust into a priority-based trust, at the same time taking the majority opinion into account. This way of combining these different metrics is a novel feature of our model and is tailored specifically for the domain of VANET. Additionally, our model does not rely on introducing opinion piggybacking in message passing and the associated algorithms to generate and aggregate opinions at each individual node.

Golle et al. [7] present a technique that aims to address the problem of detecting and correcting malicious data in VANETs. The key assumption of their approach is in maintaining a model of VANET at every node. This model contains all the knowledge that a particular node has about the VANET. Incoming information can then be evaluated against the agent's model of VANET. If all the data received agrees with the model with a high probability then the agent accepts the validity of the data. However, in the case of receiving data which is inconsistent with the model, the agent relies on a heuristic that tries to restore consistency by finding the simplest explanation possible and also ranks various explanations. The data that is consistent with the highest ranking explanation(s) is then accepted by the node. The major strength of this approach is that it provides strong security against adversaries that might even be highly trusted members in the network or might be colluding together to spread malicious data. The approach that we present in this paper is orthogonal to their approach. In particular, we do not aim to detect and correct malicious data in the network, instead we want to detect the entities (agents or cars) that are generating this malicious data, establishing trust or distrust in the entity itself. This allows an agent to avoid an interaction with a distrustful agent in future.

A number of researchers have proposed trust and reputation models with role-based approach and the notion of confidence [19, 24, 14]. In particular, [9] introduced FIRE, a framework that integrates direct trust and role-based trust, in which the direct trust model of [23] is proposed as the method for capturing this element of the overall calculation, with some adjustment to consider more carefully the decay of trust values over time. In contrast, our model incorporates role-based trust and experience-based trust, which are combined using a priority-based approach, together with majority-based trust to aggregately evaluate the trustworthiness of agents while taking into consideration the important properties specific to VANET environments.

5. CONCLUSION AND FUTURE WORK

The question of placing trust in the data received from other agents in VANETs can potentially become a question of life and death. The success of deploying VANETs, therefore, is contingent upon the success in establishing effective methods of trust establishment [12]. In this work we started by discussing some of the key challenges to modeling the trust of agents in VANET environments followed by identifying the areas where the existing trust models in the domain of multi-agent systems are lacking in their applicability to VANETs. We then presented our expanded trust model for agents in VANETs. Our model is a novel integration of several trust metrics, including role-based trust, experience-based trust, priority-based trust, and majority-based trust. It is also important to note that our expanded trust is decentralized, location/time specific, event/task specific, able to cope with the data sparsity problem, cumulative in order to be scalable, sensitive to privacy concerns, and able to support system-level security. Experimental results demonstrate that our approach works effectively for the domain of VANETs.

For future work, we plan to explore various extensions to our current model. One interesting topic to explore is how to make use of a "commuter pool" – a set of agents that travel the same route with some regularity, as mentioned in Section 2.1.2. This would provide a social network where trust may be built up and frequent encounters may occur. This scenario would heighten the value of experience-based trust as part of the model.

Considering effective modeling of location information could also form an important thread for future research, due to its role in the calculation of majority-based opinion. For example, to avoid spoofing of location information, independent methods for vehicle tracking may need to be incorporated. We may also explore how to integrate incentives for drivers to opt into honest location reporting (e.g. as a precondition to receiving information from other vehicles).

To cope with various malicious attacks in general is another interesting topic of research. Collusion is notoriously difficult to address, but individual vehicles that are misreporting may possibly be detected due to differences with other vehicles, through our majority opinion algorithm. The case where agents fail to report events is also an interesting one to explore, for future research. If location tracking information becomes more prevalent, failure to report a life critical event at that location may be independent reason to decrease trustworthiness; vehicles in special roles (such as police) would likely serve to confirm the presence of such a life critical event. Current models of trust and reputation in multiagent systems have focused more on evaluating the trustworthiness of information that has been received, rather than considering the lack of reporting. Perhaps some new ground in trust modeling would be introduced by this research.

For future work we also plan to expand our experimental evaluation to include more complex scenarios where we test the effectiveness of other components including *event/task* and *location/time* specific components. Approaches such as that of [22] or of [21] may be particularly valuable to consider, as they propose methods to also be context-sensitive when modeling multidimensional trust. Furthermore, it is also important to measure scalability of our trust model with an increasing number of agents in the system. In fact, increasing the number of vehicles in our simulations may also provide additional insights into how best to set the value of n in Step 1 of our algorithm. We could also experiment with different settings in our experimental evaluation, for instance allowing nodes to randomly lie about congestion on a road.

We could also consider a scenario where more than one agent (vehicle) in VANET forms a coalition with other agents to achieve a common goal. For instance, one such goal could be to cause

mayhem in the network which can be attributed to vandalism or terrorism [12]. The consequences can be very critical and might end up claiming many lives. Future experimentation could also include cases where life critical events such as accidents are at play. In these cases, some kind of authority should be involved and this can serve to keep the other vehicles on the road honest in their reporting. A false report would differ with that of the authority. These experiments would therefore provide greated insights into the value of our concept of role-based trust.

As a final thread for future research, we may investigate the approaches of other authors who are also concerned with the issues of scalability and privacy that we are interested in addressing within our model, in order to determine new directions, for example, a position-based clustering technique for communication between agents as proposed in [28]; preserving the privacy of an agent through the use of proxies in peer-to-peer data sharing as in [13] which suggests that proxies may provide valuable masking of the identity of an agent, as long as they are trusted.

6. REFERENCES

[1] Wikipedia on road traffic safety. *http://en.wikipedia.org/wiki/Road-traffic_safety*.

[2] I. Chisalita and N. Shahmehri. On the design of safety communication systems for vehicles. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 37(6):933–945, 2007.

[3] D. R. Choffnes and F. E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *Proceedings of VANET*, 2005.

[4] F. Dotzer. Vars: A vehicle ad-hoc network reputation system. In *Proceedings of WoWMoM*, 2005.

[5] S. Eichler, C. Schroth, and J. Eberspacher. Car-to-car communication.

[6] GM. Threat assessment algorithm. *http://204.68.195.151/people/injury/research/pub/acas/acas-fieldtest/*.

[7] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in vanets. In *Proceedings of VANET*, 2004.

[8] D. Huynh, N. Jennings, and N. Shadbolt. Developing an integrated trust and reputation model for open multi-agent systems. In *Proceedings of the Fifth International Conference on Autonomous Agents Workshop on Trust in Agent Societies*, 2004.

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

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

[11] C. Leckie and R. Kotagiri. Policies for sharing distributed probabilistic beliefs. In *Proceedings of ACSC*, pages 285–290, 2003.

[12] R. Lin, S. Kraus, and Y. Shavitt. On the benefit of cheating by self-interested agents in vehicular networks. In *Proceedings of International Autonomous Agents and Multi Agent Systems (AAMAS)*, 2007.

[13] Y. Lu, W. Wang, B. Bhargava, and D. Xu. Trust-based privacy preservation for peer-to-peer data sharing. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(3):498–502, 2006.

[14] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. In *Proceedings of the 35th Hawaii International Conference on System Science (HICSS)*, 2002.

[15] R. Mukherjee, B. Banerjee, and S. Sen. Learning mutual trust. *Trust in Cyber-societies, Springer-Verlag*, pages 145–158, 2001.

[16] S. Rahman and U. Hengartner. Secure vehicle crash reporting. In *Proceedings of MOBICOMM*, 2007.

[17] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[18] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15:39–68, 2007.

[19] M. Raya, P. Papadimitratos, V. Gligor, and J. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. *Technical Report, LCA-REPORT-2007-003*, 2007.

[20] K. Regan, P. Poupart, and R. Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *Proceedings of the Twenty-First Conference on Artificial Intelligence (AAAI)*, 2006.

[21] M. Rehak and M. Pechoucek. Trust modeling with context representation and generalized identities. In *Proceedings of the Eleventh International Workshop on Cooperative Information Agents (CIA 2007)*, 2007.

[22] A. Rettinger, M. Nickles, and V. Tresp. Learning initial trust among interacting agents. In *Proceedings of the Eleventh International Workshop on Cooperative Information Agents (CIA 2007)*, 2007.

[23] J. Sabater and C. Sierra. Regret: A reputation model for gregarious societies. In *Proceedings of the Fifth International Conference on Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies*, pages 61–69, 2001.

[24] W. Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Auton Agent Multi-Agent Sys*, 12:183–198, 2006.

[25] T. C. to Car Communication Consortium (C2CC). *http://www.car-to-car.org/*.

[26] T. Tran. A reliability modelling based strategy to avoid infinite harm from dishonest sellers in electronic marketplaces. *Journal of Business and Technology (JBT), Special Issue on Business Agents and the Semantic Web*, 1(1):69–76, 2005.

[27] Y. Wang and J. Vassileva. Bayesian network-based trust model. In *Proceedings of the 6th International Workshop on Trust, Privacy, Deception and Fraud in Agent Systems*, 2003.

[28] Z. Wang, L. Liu, M. Zhou, and N. Ansari. A position-based clustering technique for ad hoc intervehicle communication. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):201–208, 2008.

[29] B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *Proceedings of the 4th International Workshop on Cooperative Information Agents*, pages 154–165, 2000.

[30] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: The problem of unfair ratings. In *Proceedings of the Eighth International Conference on Electronic Commerce (ICEC'06)*, 2006.