A Survey on Trust Management for Intelligent Transportation System

Shuo Ma, Ouri Wolfson
Department of Computer Science
University of Illinois at Chicago
Chicago, U.S.A.
{sma,wolfson}@cs.uic.edu

Jie Lin
Department of Civil and Materials Engineering
University of Illinois at Chicago
Chicago, U.S.A.
janelin@uic.edu

ABSTRACT

Trust management is a fundamental and critical aspect of any serious application in ITS. However, only a few studies have addressed this important problem. In this paper, we present a survey on trust management for ITS. We first describe the properties of trust, trust metrics and potential attacks against trust management schemes. Existing related works are then reviewed based on the way in which trust management is implemented. Along with the review, we also identify some open research questions for future work, and consequently present a novel idea of trust management implementation.

Categories and Subject Descriptors

A.1 [**General Literature**]: Introductory and survey; C.2.0 [**Computer-Communication Networks**]: General;

General Terms

Security

Keywords

Trust management, Intelligent Transportation System (ITS), Entity trust, Data trust, Attack prevention

1. INTRODUCTION

In recent years, ITS (Intelligent Transportation System) has drawn increasing professional attention from researchers in academia and industry companies as well as official authorities, and has been considered the next life-changing technological revolution. The prospect of ITS promises a variety of applications, including safety applications, crowd-sourcing applications, entertainment applications, etc. Many prototypical applications [8, 9, 10] have been proposed. All of these proposals focus mainly on the implementation of application-specific functionalities, yet they all overlook a fundamental issue: namely, trust management.

Trust is a wide-ranging concept used in many disciplines like sociology, economics, psychology, computing, etc [1]. In this paper, we consider the semantics of "trust" only in the field of distributed systems and networking. Specifically, in traditional online e-commerce environments such as eBay, Amazon, the beta reputation system [13], etc., trust management, roughly speaking, refers to the management of the trustworthiness of relationships among entities. For the sake of simplicity, in a trust relationship in which entity *X* trusts entity *Y*, we refer to *X* as the *trustor* and *Y* as the *trustee*. In a trust management scheme for online communities, the direct consequence of a distrusted relationship is typically to prevent any meaningful interaction between the two involved entities.

In wireless networks like VANETs, ITS, etc., the concept of trust inherits its old interpretation from online e-commerce environments and also extends to the trust of an entity to data. For instance, in a crowd-sourcing application, entities need to make trust decisions based on received messages. We refer to the former interpretation of trust as *entity trust* and the latter interpretation as *data trust*.

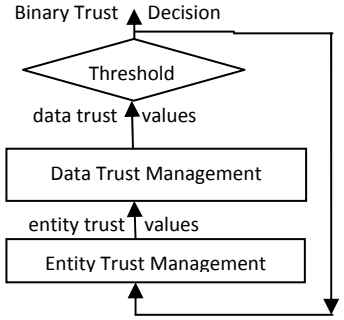

Figure 1: Potential dependence between entity and data trust management

One way to implement data trust management is to employ entity data trust management. As shown in Figure 1, data trust management can be layered or combined with entity trust management. Data trust management utilizes entity trust values generated by

IWCTS'11, Nov 1, 2011, Chicago, IL, U.S.A.

the entity trust management layer to output data trust values. Data trust values are converted to binary decisions via comparison with a trust threshold value. Those decisions are in turn fed back to the entity trust management layer, which updates entity trust values. To illustrate the above relations, consider a crowd-sourcing application. When an entity needs to evaluate the authenticity of a received message, it can use its trust value for the message's generator as the basis for the calculation of the trust value for the message. The resulting trust value is then converted to a trust decision. This decision will later be used to update the entity's trust value for the message's generator. Note that the dependence on entity trust management is not indispensable for data trust management implementations. Some proposed data management schemes [11, 12] only exploit the fact that information is often redundant in ITS to deal with data trust issues. However, all of the existing proposals are implemented in a completely distributed fashion. In contrast, we suggest a novel idea of implementing trust management in a semi-centralized fashion in part B, Section 3.

For data trust management, it is worthwhile to make a distinction between data generated by sensors and data generated by humans. Some ITS applications, e.g. traffic monitoring, may mainly rely on data generated by deployed sensors. In contrast, other ITS applications may involve lots of human interaction, and are heavily contingent upon human input. For instance, a crowd-sourcing application for public transportation information may require pedestrians to use mobile devices to report surprising temporary subway station closings or bus schedule changes. Likewise, a taxi ride-sharing application relies on manual ride requests from travelers and replies from taxi drivers. Both types of applications face inevitable data trust hazards. Data generated by humans are always risky to trust, due to the potential presence of malicious entities. Data generated by sensors become unreliable if sensors are hacked and controlled by malicious entities; however, the design and implementation of tamper-proof sensors are considered as security issues and hence are not discussed here. Accordingly, in this survey we focus on data trust management for data generated by humans only.

In Section 2, we first examine the properties of trust and trust metrics, and briefly describe common attacks against trust management schemes in ITS. Next, we overview the related work of trust management for ITS and show how those works are connected to the concepts we describe in Section 3. Finally, we summarize our discussion and conclude our work in Section 4.

2. Concepts

A. Properties of Trust

Though we have confined the concept of trust within a particular field, trust remains a complex, multiple-faceted concept. Thus in this section, we examine the unique properties of trust in ITS.

First, trust is *partially transitive*. On one hand, transitiveness implies that trust can be acquired either directly or indirectly. Direct trust is always earned via individual experience. Indirect trust is earned via referrals, opinions, etc. On another hand, partial transitiveness implies that indirect trust often comes with special constraints, e.g. a maximum referral hop limit. It is desirable for a trust management scheme to take both direct and indirect trust into consideration, though it may assign different weights to them.

Trust is both *static* and *dynamic*. Static trust means that the value of trust does not change over time. Identity-based trust is typical static trust. Here the identity refers to information regarding an entity's social role and status as well as its relationship with other entities. In instances of identity-based trust, one can consider the following example: Without having had any previous interactions with it, it is reasonable for a private passenger car to trust a police patrol car. Dynamic trust means that the value of trust changes over time. For example, interaction-based trust is dynamic.

Trust is *situation-dependent*. For example, an entity may adapt its threshold for a trusted decision, according to situation. In general, the trust threshold is set higher in situations where a trust decision matters more, e.g. in safety applications. The threshold may also be adapted to the security level of the system [17]. For instance, an entity would have a lower threshold in a system with sophisticated cryptography than in one without cryptography.

B. Trust Metrics

While properties of trust often tell how trust can be measured, trust metrics tell what to measure in order to evaluate trust. In ITS applications, a trust metric is often some characteristic of an entity, e.g. honesty in message generation. Or it is some capability of an entity, e.g. the capability to distinguish between truthful and false messages, the capability to provide reliable, timely and integral message delivery [18], etc. Here trust metrics are actually all entity trust metrics. This is because we only consider the authenticity of data, so there is no need for various data trust metrics.

Trust metrics inherit the properties of trust. To understand this, we show how an example trust metric can be utilized in trust management. Consider honesty in message generation as a specific trust metric. The value of this metric can be determined statically, i.e. initialized based on the role of an entity. For example, a policeman possesses a higher static trust value in honesty than a regular citizen. Though static, this trust value can also be dynamically changed through interactions; value may also be gained indirectly. For example, entity *X* may get the value of trust in honesty of entity *Y* from entity *Z*. Finally, the trust threshold related to this metric may be adapted to situations as

well. For example, suppose an entity decides to distrust any messages originated from entities with a trust value in honesty lower than a certain threshold. Such a threshold can be set higher in a dense network where information is abundant, and thus the entity can be more selective in trusting messages. In contrast, the trust threshold should be lower in a sparse network where information is rare and the entity must be more open to messages.

C. Potential Attacks

There are some common attacks [15] deliberately designed to sabotage trust management schemes. Those attacks include simple false information injection attacks, on-and-off attacks, Sybil attacks and collusion attacks. A simple false information injection attack happens when a malicious entity generates false information on purpose. An on-and-off attack happens when a malign entity behaves well or badly alternatively in order to dodge detection. A Sybil attack [16] happens when a malign entity uses a large number of fake identities to beat the redundancy check of the network. For instance, an entity may ask opinions for a message from multiple different entities. A Sybil entity with many pseudonyms, i.e. bogus entities, can fake all those opinions using different pseudonyms and consequently fool the entity asking for help. A collusion attack happens when a group of well-coordinated malign entities contrive a conspiracy. We will show how these attacks are handled when reviewing related works of trust management in the next section.

Note that we consider traditional security problems like access controls, cryptography, etc., as a separate class of issues from the entity and data trust management problems investigated here. The difference between those two types of problems is first described in [6], which refers to traditional security issues as *hard security* and to trust management as *soft security*. Thus, traditional information security hazards, such as modification of messages, denial of the generation of messages, etc., are not considered as attacks specially targeting trust management schemes. These traditional hazards are typically prevented by employing asymmetric key cryptography. For example, [5, 7, 11, 12] use digital signatures to prevent malicious entities from modifying messages without detection. However, it remains a separate important research direction to consider traditional security issues in an ITS environment. For instance, the issue of how to efficiently manage public/private key pairs without revealing the privacy of entities in an ITS environment needs to be studied. As an example, [2] suggests using anonymous key pairs to preserve sensitive information regarding the entity, such as owner, identity, routine, etc. [3] proposes an approach to deploy anonymous key pairs in VANETs.

3. TRUST MANAGEMENT FOR ITS

In this section, we provide our critical reviews of the existing works on trust management for ITS. The works are organized based on the way in which trust management is implemented.

A. Opinion Piggybacking

For the purpose of validating a received message, both [4] and [5] consider a technique named *opinion piggybacking*. Opinion piggybacking means that each entity forwarding a message appends its own opinion to the message and decides whether or not to trust the message based on the attached opinions.

Specifically, in [4] each such opinion of a forwarding entity is a triple tuple, including a continuous trust value o_{val} representing its trust value for the message, a discrete trust level $s \in \{1,2,3\}$ representing its trust value for the message's generator, and its ID. The paper provides forwarding entities with an algorithm to calculate the values of their own o_{val} and s for a message, by considering all of the previous opinions attached to the message, combined with locally stored trust values for the corresponding opinion providers. The trust decision threshold becomes dynamic when taking the spatial distance between locations of the message source entity and the deciding entity, as well as the deciding entity's familiarity of the area, into account. However, [4] provides little information regarding how trust values for entities are initialized and updated.

In contrast to [4], in [5] each opinion consists of a binary decision for the message (i.e. whether it is trusted or not) a confidence value for the decision, and the signature of the opinion generator. Specifically, the paper adapts a cluster-based routing protocol to propagate messages. The cluster-head entity calculates the trust value for the message by considering both the confidence of each attached opinion and its own trust value for the opinion's corresponding generator. The cluster-head then makes a decision based on the calculation result to determine whether or not to relay the message. Consequently, only trusted messages get disseminated among different clusters.

An entity updates its trust values for other entities according to the following rules. Entity X's trust for entity Y is positively enforced if Y's opinion on a message leads to a correct decision; otherwise X's trust value for Y is reduced. The paper does not elaborate on how an entity is able to verify the correctness of a trust decision. An entity can absolutely discover the authenticity of the event reported in the message by direct observation. For example, an entity which has trusted a message reporting the clearance of a road maintenance on its way to a destination would verify the message is correct when passing by the site. However, it is obvious that sometimes an entity has no way to witness an event reported by a once trusted message. For instance, an entity that has trusted a message reporting a jam on its future trajectory and thus decides to reroute is unable to verify the truth via direct observation. Thus, it is reasonable for an entity to confirm the authenticity of a reported event, if it receives a similar message from a highly trusted entity

too. The threshold for "high trustworthiness" can be customized by individual entities.

Chen et al. [5] propose some methods for preventing on-and-off attacks. It suggests to defend against such attacks by utilizing the "hard to win but easy to lose" principle. This principle imitates a practical norm in real social life: namely, that trust value for an entity is difficult to build up but easy to tear down. Consequently, entities have to behave very discreetly all the time in order to keep their earned credit. A similar kind of attack, i.e. *betrayal attacks*, wherein malicious entities first act normally to build up their trustworthiness but then abruptly start malign behaviors, can also be thwarted by following the same principle. Specifically, the scheme proposed in [5] employs a forgetting factor that allows trust values earned via interaction to decay over time. It also uses a larger value for the decrease factor than that of the increase factor. As a result, trust values for entities always gain slowly and lose fast over multiple interactions.

One unique problem with the approach in [5] is regarding the confidence value of the opinion. Although the confidence value helps model the uncertainty of the opinion, there is no good guidance for entities to accurately estimate such a value. If, unfortunately, the estimation goes wild, the poorly calculated confidence value will damage the entity's trustworthiness badly in the feedback stage. In addition, the scheme is bonded with a cluster-based routing scheme. Therefore its applicability narrows dramatically.

Both schemes described in [4, 5] suffer from several other problems in the ITS environment. First is that a forwarding entity of a message is likely to have no previous interaction with the message's generator and have no basis to provide an opinion to the message. One solution is to ask the message generator to embed its encrypted role into the message. Then forwarding entities can use static role-based trust value as the initial ground for opinion-giving.

Second is that both schemes only use a single trust value to measure entity trust. However, it is important for them to make a distinction between trust in honesty and trust in discernment. Trust in honesty measures the trustor's faith in the belief that the trustee will not produce falsified information, whereas trust in discernment measures the trustor's faith in the trustee's ability to make correct trust decisions about messages. The difference between these two metrics reflects a similar situation in our social network: someone may be so honest that s/he never tells a lie; however, s/he may be also so inexperienced that s/he will be easily fooled by a lie. Likewise, entity *X*, with a high trust value in honesty but a low trust value in discernment, may barely generate false information but keep making wrong decisions on messages, due to reasons such as lack of interaction or being surrounded by colluding entities. In such a case, another entity *Y* is expected to believe data originating from *X* but ignore *X*'s trust opinions for other messages. In this case, if two trust metrics are mixed and represented by a single value, *Y* is likely to either overestimate the credibility of *X*'s opinions or underestimate the reliability of data generated by *X*.

The last problem is most crucial. To show what the issue is, consider a scenario where a malicious entity *O* broadcasts false messages in one neighborhood for a while, flees to another far-away area, say 5 miles from the old neighborhood, and starts to broadcast false messages again. It is likely that in the new area, there are few entities which have previous interactions with *O*, and thus have no idea of *O*'s past bad behaviors. Consequently, the forwarding entities of falsified messages generated by *O* in the new area may make incorrect trust decisions in the beginning phase. Those entities may discover the evil of *O* after a period of time, but by then *O* may have gone far away again.

B. A Novel Trust Management Scheme

The last problem described above has its root in the fact that the proposed trust management schemes are fully decentralized: there is a lack of a central server that records the trustworthiness of entities. This inspires in us a novel idea of implementing trust management in a semi-centralized scheme.

Next we briefly describe how the scheme will work. The scheme assumes the existence of a central server, whose public key is globally known. Each entity registers with the server. The server stores reputation values of entities. Each entity can log into the server to file complaint or praise about another entity, whose corresponding reputation value is updated by the server according to certain rules. Each entity is required to log into the server every T_{ru} time in order to download a certificate encrypted by the server's private key. This certificate contains a statement associating the identity and the latest reputation value stored in the server of the entity. When an entity broadcasts a message, it is required to embed the certificate into the message. Consequently, any entity receiving the message can decrypt the certificate using the server's public key and get a rough idea about the trustworthiness of the message's generator via the embedded reputation. In addition, an entity is allowed to question the server about the reputation value of another entity at any time if it suspects that the reputation value in the received certificate is out of date. This means will efficiently prevent the aforementioned perpetrate-run-perpetrate type of attacks.

There are clearly some details that need to be determined for the above scheme. For example, what is the best value for T_{ru} such that a good balance is struck between the freshness of the reputation value and the efficiency of communication cost? Likewise, what are the rules of the server when updating reputation values, given reported complaints and praise? We leave those questions as future work and plan to implement this scheme in a forthcoming paper.

C. Opinion Inquiring

Akin to the opinion piggybacking approach, Minhas et al. [11] propose to validate messages via the consideration of other entities' opinions too. However, unlike in [4, 5] where opinions are bonded with the message forwarding protocol and consequently cause an entity to lose the option to choose opinion providers, the paper allows an entity actively to ask for opinions from entities it picks. When an entity receives a message announcing an event, it asks opinions from N other most trusted entities, based on the trust values output by the lower layer entity trust management, of which the details are briefly summarized below.

Similar to the approach in [5], the entity trust consists of role-based trust and interaction-based trust. A globally trusted certificate authority (CA) issues an encrypted certificate to each entity, which binds the role and the public key of the entity. By requiring communicating parties to exchange certificates before interactions, role masquerading is efficiently prevented. Interaction-based trust is learned by an entity via past interactions. The principle of the learning process is the same as that in [5]: "good" interactions get rewarded and "bad" interactions get punished. Each entity ranks the trustworthiness of other entities in a major order of role-based trust and a minor order of interaction-based trust.

Once it has all of the opinions for a received message, the entity uses an equation to calculate the trust value for the message. Within that calculation, each opinion is assigned different weight in relation to factors including: (i) local role-based trust for the opinion provider; (ii) local interaction-based trust for the opinion provider; (iii) temporal closeness between the time when the event takes place and the time when the opinion is generated; and (iv) spatial closeness between the location where the event takes place and the location where the opinion is generated. The output trust value is finally converted to a decision.

There is one unique problem with this opinion inquiry-based scheme. Since the selection of entities for which opinions are asked is based on trust value rather than spatial closeness, the scheme may suffer a high communication cost and time delay if, unfortunately, the selected entities are far away.

D. Passive Majority Consensus

Patwardhan et al. [7] utilize the fact that information is often redundant in ITS applications to validate a message. The paper assumes a network where anchored resources, such as parking meters and roadside sensors, perpetually provide trustworthy data to surrounding entities. A message can be accepted, i.e. validated, by an entity through either a majority consensus or direct communication with the anchored resource that produces the message. A majority consensus at an entity O will validate a message M if (i) at the time of consensus, O has received at least P other messages that report the same event as message M does, all from different entities (ii) message M along with the other P messages consist of the majority opinion regarding the event, where P is a system parameter.

The proposed scheme takes a passive approach in waiting for messages from other entities, in contrast to the scheme in [11], which proactively asks for opinions from other entities. The drawback of the passive approach is that an entity may wait for a long time or even forever to receive enough messages of the same event required by the majority consensus in a sparse network. Thus, it is natural to consider a neutral approach combining the merits of the active and passive approaches. For example, adapt the state of activity of the entity to the network density, and consider both trust value and geographical closeness as metrics when choosing entities for opinions. In addition, these two schemes also suffer from the same three problems of schemes in [4, 5] described before.

E. Position Verification

Golle et al. [12] also present a data trust management scheme for VANETs without the use of any entity trust metric. The authors assume that each entity maintains a *model of the VANET* against which any incoming message will be validated. At its core, a model of the VANET is a set of observations about the VANET already known to an entity. All messages consistent with the entity's model of the VANET are validated; otherwise, the entity attempts to eliminate inconsistency by ranking all of the possible explanations and picking the simplest explanation.

The paper does not provide a general algorithm for validating data against the model of the VANET, but uses examples to demonstrate the idea. For instance, the paper shows how the approach is used to prevent Sybil attacks. Specifically, drivers are supposed to be capable of broadcasting position statements pertaining to themselves and others. For example, driver A may broadcast a message stating, "I am at location L_1 and I spot a police car at location L_2". The paper assumes that broadcasted position statements are immediately available to entities network-wide. Now suppose that in its model of the VANET, entity A has identified that neighbor B and C are indeed distinct entities. Further suppose that both B and C state that identity X and identity Y, which are far away from A, are located at the same position. However, at the same time, X and Y themselves state that they are at different positions. Clearly there is a conflict among the statements received by A. There are two possible explanations which can resolve the conflict: namely (i) B and C collude to lie about X and Y or (ii) X or Y is faked by a Sybil entity. A makes the choice by utilizing the so-called *adversarial parsimony* principle, which essentially picks the explanation involving fewest malicious entities. Thus, in this case, the second explanation, i.e. the potential Sybil attack, is detected by entity A.

The problem of this approach is that it is not practical for each vehicle to build and maintain a model of the VANET in real time, since the paper assumes that any

broadcasted statement made by an entity is instantly universally available to all other entities, i.e. ignoring the propagation time of messages. Besides, since Sybil attacks require the lack of a central trusted authority, such as a CA, which provides identity authentication services, the above approach is not necessary for ITS, as entities may get certificates from trusted infrastructures, e.g. gas stations, parking lots or dedicated roadside facilities [14] to authenticate themselves. However, implementation level details for the approach, such as which infrastructures are considered to be trusted, how certifications are managed, etc., needed to be thoroughly thought out.

F. Collusion Attacks Prevention

Few attempts have been made to prevent collusion attacks. Some existing trust management schemes for ITS are considered to have a certain defensive ability against collusion attacks of a particular form. For instance, the scheme proposed in [12] can prevent a special kind of collusion attack, i.e. position spoofing by a group of malicious entities. However, because of the nature of the adversarial parsimony principle, the scheme only works when even the simplest explanation includes a collusion attack. Trust management systems described in [4, 5] are also considered to be capable of absorbing colluded false message injection attacks to a certain degree. Nevertheless, the strength of such a capability is contingent upon factors like network density, complexity of the collusion, etc. In short, the principles of defending against collusion attacks in general as well as concrete methods for preventing application-specific collusion attacks still await further study.

4. CONCLUSION

Trust management is a crucial aspect for ITS applications, and yet remains an open problem. Existing works on trust management for ITS aim at different kinds of sub-problems. Unfortunately, there is a lack of classification and comparison of these works against a uniform backdrop. For this purpose, we present a survey on trust management for ITS. Specifically, our survey describes the properties of trust, trust metrics and potential attacks against trust management schemes. Existing related works are carefully reviewed and compared. We also contribute a novel idea of implementing trust management in a semi-centralized fashion. Our work is an important step toward building an efficient, comprehensive and reliable trust management scheme for ITS.

5. REFERENCES

[1] Jøsang, A., Keser, C. and Dimitrakos, T. Can We Manage Trust? *Proc. of 3rd International Conference on Trust Management*, Versailles, Springer-Verlag, 05

[2] Raya, M. and Hubaux, J.-P. Securing vehicular ad hoc networks. *J. Comput. Secur.*, IOS Press, 2007, 15

[3] Lai, C., Chang, H. and Lu, C. C. A secure anonymous key mechanism for privacy protection in *ITST'09*, 2009, 635 -640

[4] Dotzer, F., Fischer, L. and Magiera, P. VARS: a vehicle ad-hoc network reputation system. *WoWMoM 2005*.

[5] Chen, C., Zhang, J., Cohen, R. and Ho, P.-H. A Trust Modeling Framework for Message Propagation and Evaluation in VANETs. *ITCS, 2010*.

[6] L. Rasmusson and S. Janssen. Simulated Social Control for Secure Internet Commerce. Proceedings of the *New Security Paradigms Workshop*.ACM, 1996.

[7] Patwardhan, A., Joshi, A., Finin, T. and Yesha, Y. A Data Intensive Reputation Management Scheme for Vehicular Ad Hoc Networks. Proc. of 3rd Int'l *Conference on Mobile and Ubiquitous Systems: Networking Services*, 2006

[8] Lalos, P., Korres, A., Datsikas, C.K., Tombras, G.S. and Peppas, K. A Framework for Dynamic Car and Taxi Pools with the Use of Positioning Systems. *COMPUTATIONWORLD '09.*

[9] Tao, C.-C. Dynamic Taxi-Sharing Service Using Intelligent Transportation System Technologies *Wireless Communications, Networking and Mobile Computing*, 2007. International Conference on, 2007

[10] Lee, U., Zhou and B., Gerla, M. Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. *Wireless Communications*, IEEE , vol.13, no.5, pp.52-57, 2006.

[11] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen. Towards expanded trust management for agents in vehicular ad-hoc networks. International Journal of Computational Intelligence: Theory and Practice (IJCITP), vol. 5, no. 1, 2010.

[12] Golle, P., Greene, D. and Staddon, J. Detecting and correcting malicious data in VANETs. Proc. of 1st *ACM international workshop on Vehicular ad hoc networks*, ACM, 2004, 29-37

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

[14] Wex, P., Breuer, J., Held, A.; Leinmuller, T. and Delgrossi, L. Trust Issues for Vehicular Ad Hoc Networks. *VTC Spring* 2008.

[15] J.-H. Cho and A. Swami. Towards trust-based cognitive networks: A survey of trust management for mobile ad hoc networks. In Proceedings of the 14th *International Command and Control Research and Technology Symposium*, Washington, DC, 2009.

[16] J. Douceur. The sybil attack. In Proceedings of the *First International Workshop on Peer-To-Peer Systems* (IPTPS), 2002.

[17] Gerlach, M. Trust for Vehicular Applications Proc. of the 8th Int'l *Symposium on Autonomous Decentralized Systems*, IEEE Computer Society, 07.

[18] Z. Liu, A. W. Joy, and R. A. Thompson. A Dynamic Trust Model for Mobile Ad Hoc Networks. Proc. 10th IEEE Int'l Workshop on Future Trends of Distributed Computing Systems, Sushou, China, 26-28 May 2004, pp. 80-85.