Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 5254141, 10 pages
http://dx.doi.org/10.1155/2016/5254141

Research Article

Anomaly Detection for Internet of Vehicles: A Trust Management Scheme with Affinity Propagation

Shu Yang, Zhihan Liu, Jinglin Li, Shangguang Wang, and Fangchun Yang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Jinglin Li; jlli@bupt.edu.cn

Received 31 December 2015; Accepted 1 March 2016

Academic Editor: Seung Yang

Anomaly detection is critical for intelligent vehicle (IV) collaboration. Forming clusters/platoons, IVs can work together to accomplish complex jobs that they are unable to perform individually. To improve security and efficiency of Internet of Vehicles, IVs' anomaly detection has been extensively studied and a number of trust-based approaches have been proposed. However, most of these proposals either pay little attention to leader-based detection algorithm or ignore the utility of networked Roadside-Units (RSUs). In this paper, we introduce a trust-based anomaly detection scheme for IVs, where some malicious or incapable vehicles are existing on roads. The proposed scheme works by allowing IVs to detect abnormal vehicles, communicate with each other, and finally converge to some trustworthy cluster heads (CHs). Periodically, the CHs take responsibility for intracluster trust management. Moreover, the scheme is enhanced with a distributed supervising mechanism and a central reputation arbitrator to assure robustness and fairness in detecting process. The simulation results show that our scheme can achieve a low detection failure rate below 1%, demonstrating its ability to detect and filter the abnormal vehicles.

1. Introduction

Internet of Vehicles (IoV) is an open converged network system supporting human-vehicles-environment cooperation [1]. Fusing multiple advanced terms, such as VANET [2], autonomous driving [3], cloud computing [4], and multiagent system (MAS) [5], this hybrid concept plays a fundamental role towards a cooperative and effective intelligent transport system. An anomaly detection scheme is desirable in an environment filled up with uncertainty. Primarily, the security problem is motivated by the question [6] "How can I trust the information content I receive?" This issue is then decomposed into two subterms: "Is the communication channel via which I receive messages from a sender secure?" and "How can I trust the sender of the messages I receive?" The decomposition allows us to tell the difference between *computational trust* and *behavioral trust*. Being a complementary part of *computational trust* (such as encryption and tamper-proofing), the model of *behavioral trust* admits information's imperfection in an open system; therefore, individuals need extra trust-related information in decision-making [7]. Extra trust-related information could be extracted from history reputation or could be elicited from interaction experience between two individuals. Being capable of providing a measurement of trustworthiness, behavioral-based trust management enables intelligent vehicles to improve collaborations by reducing false or malicious behaviors. Anomaly detection technology is the key method to build behavioral trust.

This paper aims to detect anomaly vehicles in autonomous driving environment. As commercial IVs are drawing near, we have to face the facts that vehicles are more and more intelligent. Meanwhile, cyber vehicles are unprecedented and vulnerable when supported by an uncertain and dynamic network [8]. Malicious attacks and information tampering, along with system failures, will directly threaten human lives and properties. Anomaly vehicles include malicious vehicles and incapable vehicles. Malicious vehicles are entities with intentions to make damage in driving environment. Incapable vehicles are not intentionally to give negative influence; however, they may disturb order due to their limited capability. For example, an incapable intelligent vehicle could not

behave properly in a rigid and accurate-ordered automatic driving platoon but may behave well in a normal driving pattern. On the other hand, a malicious intelligent vehicle should be forbidden in any situation. To highlight our motivations, we present the following two illustrative scenarios. *Scenario 1*: in cluster/platoon-based driving, IVs frequently communicate with each other to maintain lateral/longitudinal control. Vehicles with incapability or malicious intention may join the cluster or platoon. Their malicious or false behaviors are very likely to temper/disturb collaboration. In this safety-oriented case, local vehicles should be able to maintain robust intracluster trust to wipe out unqualified vehicles. *Scenario 2*: in efficiency-oriented case, where IVs need to collaborate in a broad area, they exchange message to presence traffic conditions, request parking plot information through VANET, and even negotiate routes to prevent traffic congestions. None of these three functions would be efficient without trustworthy collaboration. The above two scenarios suggest that a trust management scheme with anomaly detection is urgently need.

Solutions on IoV's anomaly detection still face many challenges raised by mobility including dynamic vehicle groups, real-time constraints, and intrinsic dynamic property of trust itself, which makes single or static trust measurement ineffective. Considering the mobile nature of vehicles, topology is changing so rapidly that preestablished trust relationships are likely to be invalid. As a result, two nodes need to build up trust in a timely fashion. Moreover, trust is not constant but changing along with different driving situations. An accurate trust should capture the context of interaction and history reputation. For example, a car with good reputation may not be trustworthy when it is over speed. Trust management system therefore calls for the ability to synthesize multiple resources, either from roads or from cloud. The essence of Internet of Vehicles is to obtain more safety and efficiency by integrating multiple infrastructures, networks, and vehicle intelligence. In accordance with this idea, we propose a hybrid approach called Cluster-Based Anomaly Detection (CAD). Figure 1 describes the framework of CAD. CAD is composed of two big components, namely, cluster-based trust component and central reputation component. Cluster-based trust component builds time-fashioned trust to reflect dynamic situation while central reputation component is to evaluate one's trust from a long-term perspective. These two components interact by evidence uploading and reputation providing. Cluster-based trust component has two major functions, namely, trust-based AP clustering and mutual supervision to maintain the robustness of dynamic trust.

The major contribution of this paper lies in the following two aspects:

(i) We identify cluster-based trust and reputation as two major components of anomaly detection. To exploit cluster-based trust, we propose a cluster-based trust evaluation algorithm, which modifies *Affinity Propagation Clustering* to generate the most trustworthy cluster head based on evaluation and communication. The algorithm runs in a distributed manner and shows robustness to malicious/incapable vehicles.

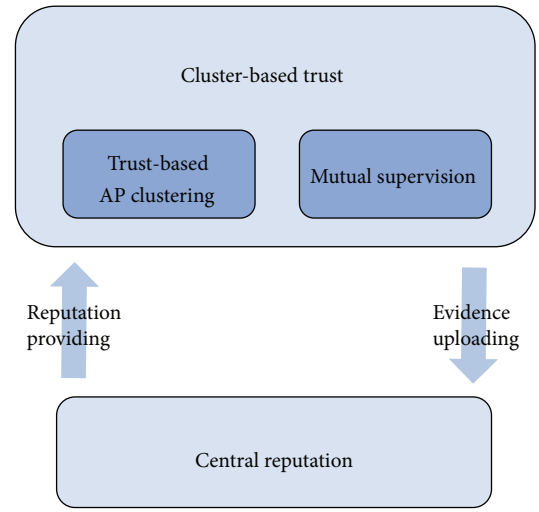

Figure 1: Framework of CAD.

(ii) We adopt a sparse RSU-enhanced reputation provision scheme. Central Arbitrator (CA) collects evidences from sparse RSUs. Then, a reputation system is established to evaluate global and history reputation from accumulated data.

2. Related Work

Trust issues stem from secure and social psychology fields and have been growing theoretically in organization management. More recently, as network technology is constantly changing the way people interact, former stable and well-structured organizations are likely to transform into another paradigm featured by agile structures and ad hoc groups. IoV, for example, is a typical agile structure that calls for collaboration among agents. Ramchurn et al. [9] pointed out that "trust pervades multiagent interaction at all levels," generally including (1) individual-level trust, whereby an agent has some beliefs about the honesty or reciprocative nature of its interaction partners, and (2) system-level trust, whereby the agents in the system are forced to be trustworthy by the rules of encounter that regulate the system. Although various schemes have been investigated, the author noticed that trust at these two levels has been dealt with separately in most times. This insight inspired us to develop a hybrid framework which takes both levels of trust into consideration.

Most existing systems in VANETs use distributed approach. Raya et al. [10] argue that the trust should be attributed to data per se in ephemeral ad hoc networks and proposed a framework for data-centric trust establishment. Their scheme shows high resilience to attackers and could converge to stable right decision. However, Raya's trust mechanism may make no contribution to reduce attackers in system level; since there is no punishment for cheating, attackers are seldom suppressed. Chen et al. [11] present a decentralized framework combined with message propagation and trust evaluation in VANET. Specifically, trust

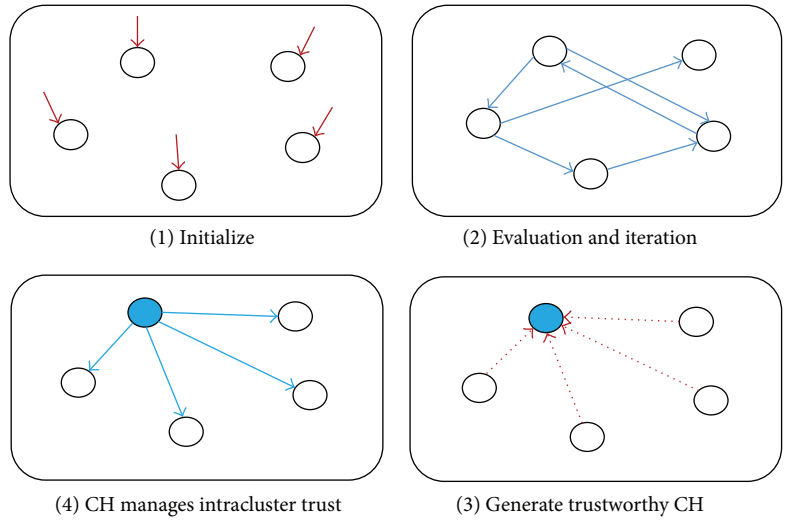

Figure 2: Four steps of Peer Detecting-based trust establishment.

measurement consists of role-based trust and experience-based trust. It is a good attempt to synthesize static priori trust (role-based trust) with dynamic situational trust (experience-based trust). Nonetheless, they did not take historical reputation into consideration. Rostamzadeh et al. [12] focus on trustworthy information dissemination by assigning trust value to each road segment. The dissemination task is to find a path which consists of a series of safe road segments. Their work is featured by good scalability and thus potential in many applications. DTM^2 [13] is a distributed trust model inspired by Job Market model. With the help of third party hardware, system could incent good behaviors and punish malicious behaviors by changing each vehicle's signal value. To conclude, the decentralized approach is developed under the assumption that there is no centralized third party to evaluate and maintain the trust value.

Recently, the RSU deployment is promoted by intelligent transport system group. Centralized trust management is not an ambiguous goal with the help of RSUs. Centralized approach is able to evaluate trust value from a global and historical view. Therefore, many works have preliminarily emerged centralized trend as a complementary of distributed system. Wang et al. [14] proposed a vertical handoff method, which improves availability of network access. Their method therefore makes contributions to building centralized trust management system. Machado and Venkatasubramanian [15] aim to aggregate advantages of both centralized and distributed trust computation. The authors categorize the messages exchanged in VANET into *alerts* and *reports*; alerts are time-critical in response to an incident while reports are evidence to evaluate quality of alerts. RSUs play Central Authority (CA) who keep track of messages and accordingly maintain a global reputation for each vehicle. Their central grading system could efficiently distinguish dishonest nodes in real-life scenarios. Huang et al. [16] utilize identity-based cryptography to integrate entity-based trust and social trust in proxy server. The email interactions among individuals are mined to obtain social trust. Trust measurement should be requested and acquired from this server. One disadvantage of this system, as the author mentioned, is that service may experience long delay due to network latency and the management entities to mine the email source. Such latency problem bothers centralized reputation system. The author then proposes a situation-aware trust architecture for VANETs [17]. A predictive trust setup system is designed to reduce on-the-scene trust setup latency. They also envision that the roadside infrastructure deserves more attention and research.

3. Trust Establishment by Peer Detecting

In this section, we illustrate the establishment of cluster-based trust. To establish trust among IVs, the key is to generate the trustworthy CH. Cluster and its head are generated after several rounds of iteration. The generated CH is an authoritative node managing intracluster trust. One of the cluster algorithms which works by passing messages between nodes is *Affinity Propagation* (AP). To start, measures of similarities are calculated for each pair; real-valued messages are then exchanged between pairs of nodes until high quality exemplars and corresponding clusters gradually emerge. The schematic is shown in Figure 2.

AP works by passing messages between nodes, which is naturally more suitable for trust establishment than other clustering algorithms because of the following characteristics: (1) transitivity: in trust theory, if $node_A$ has no direct trust with $node_C$, it could still build an indirect trust relation via $node_B$ to $node_C$; likewise, in our AP, $vehicle_A$ makes a judgement about $vehicle_C$ with the help of indirect judgement from other nodes; the primitive AP clustering algorithm therefore well-reflects transitivity, making it fit into trust establishment; (2) asymmetry: trust is not symmetric; that $node_A$ trusts $node_B$ does not guarantee $node_B$ trusts $node_A$. AP has the ability to cluster by asymmetric "distance measurement"; (3) distributed manner: AP runs in a completely distributed manner, increasing robustness to attacks; (4) moreover, it

achieves a much lower average squared error than normal clustering method [18].

The AP algorithm works iteratively. The similarity $s(i, j)$ is sent from $node_i$ to $node_j$ to measure "distance" between a pair. The responsibility $r(i, j)$ is sent from $node_j$ to $node_i$ to tell how eager i wants j to be CH. The availability $a(i, j)$ is sent from $node_i$ to $node_j$ to tell how eager j wants to be i's CH. The self-responsibility $r(i, i)$ and self-availability $a(i, i)$ both represent accumulated evidence reflecting if i is suitable to be CH. The updating process for responsibility and availability in every iteration procedure is illustrated below. More detailed works are [18, 19], which have lain the foundation of our work.

Primitive AP Iteration Process is as follows:

$$r(i,j) \longleftarrow s(i,j) - \max_{k \text{ s.t.} k \neq j} \{a(i,k) + s(i,k)\},$$

$$a(i,j) \longleftarrow \min\left\{0, r(j,j) + \sum_{\forall k \neq i,j} \max\{0, r(k,j)\}\right\}, \quad (1)$$

$$a(j,j) \longleftarrow \sum_{k \text{ s.t.} k \neq j} \max\{0, r(k,j)\}.$$

To make real-valued message converge, messages are damped by λ, $Message_{new} = \lambda Message_{old} + (1 - \lambda) Message_{new}$, where λ is a weighing factor that ranges from 0 to 1. When messages converged, a CH is generated:

$$\mathrm{CH}_i = \max_j \{a(i,j) + r(i,j)\}. \quad (2)$$

3.1. UntrustDegree. Our proposed scheme uses the fundamental idea of Affinity Propagation from a trust perspective. In general, AP could detect anomaly vehicles in a group. We design an *UntrustDegree* function as "distance measurement" for AP algorithm to find "the most trustworthy node," that is, to find the node which minimizes overall UntrustDegree. The function $UntrustDegree(i, j)$ is automatically calculated by IV. An IV can observe other vehicles' behaviors and give an UntrustDegree according to its knowledge:

$$UntrustDegree(i,j) = F_i\left(Identity, \overrightarrow{Situation}, \overrightarrow{Behavior}_j\right) \in [0,1]. \quad (3)$$

Identity is one item from set $Id = \{bus, taxi, police, private, \ldots\}$ denoting real identity of one car and could be represented by a unique digital number. $\overrightarrow{Situation}$ is a vector predefined as some basic values which gives environmental context (e.g., the weather). $\overrightarrow{Behavior}_j$ is a vector recording basic actions that IV_j has done recently. With the help of behavior detection technologies [20] or interactive gaming [21], we reasonably assume that IVs are intelligent enough to evaluate each other. The value, $UntrustDegree(i, j) \in [0, 1]$, is primarily positive but set negative, namely, $-UntrustDegree(i, j) \in [-1, 0]$, to fit AP algorithm.

The self UntrustDegree, $UntrustDegree(i, i)$, is initialized to the same value. It should be noted that a higher self-trust degree makes it more likely to become the cluster head. In our final model (discussed in Section 5.2), valid self-trust is set at a value which balances *IV's evaluation* and *historical reputation*. *Historical reputation* can only be legally announced by CA. When a group of IVs pass by a RSU, RSU will proactively download/broadcast reputations to IVs.

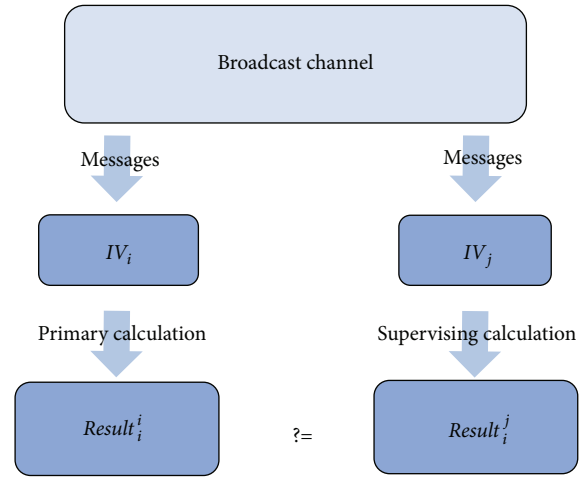

Figure 3: Mutual supervisor model.

3.2. Mutual Supervisor Model. Each IV_i receives responsibility $r(i, j)$ from the neighborhood. Also, IV_i broadcasts $a(i, j)$ to the neighborhood to claim how suitable it is to become a CH. However, a malicious/incapable node can cheat/mistake in this message passing process by broadcasting a false $a(i, j)$. For example, if IV_i broadcasts very high $a(i, j)$ to other nodes, it is more likely to be elected CH according to the AP algorithm. We need a mechanism to prevent nodes broadcast false availability or responsibility.

We proposed a supervisor model to alleviate cheating/mistaking in this process. The core of mutual supervisor model is to match IV_i with a supervisor IV_j. Among moving companions of one vehicle, a supervisor is another *IV* which can receive almost the same broadcast information by sharing the same wireless channel. A supervisor therefore listens to the supervisee related message to validate availability/responsibility by repeating the calculation of suspicious IV_i. The result calculated by IV_i itself is $Result_i^i$. The supervisor IV_j's calculation result for IV_i is $Result_i^j$. If the two results $Result_i^i$ and $Result_i^j$ have large difference, then this means IV_i is very likely to have cheated in message passing process. The integral mechanism of supervisor model is illustrated in Figure 3.

To assure a stable and honest supervisor, we apply Algorithm 1. From this algorithm, we see that IV_i has possibility to supervise another IV_j only when (1) IV_i does not tend to believe IV_j and (2) IV_i and IV_j have small relative mobility. That is, they are stable driving companions. This mechanism builds up mutual supervision relationship between two adversary nodes so that supervisor and supervisee are not likely to collude. More important, it can identify cheating nodes in message passing process.

Input: a supervisor IV_i, nearby node' states (position, speed)
Output: *pair(supervisor, supervisee)*
(1) For *Node* IV_j *in DSRC(Dedicated Short Range Communication) range*
(2) $M_{i,j} = MobMetr((Pos_i, Pos_j), (Speed_i, Speed_j))$ *//calculate mobility metric*
(3) If IV_j *has no supervisor* and $UntrustDegree(i, j) \leq Threshold$
(4) Then *Add* IV_j *in Supervisee Candidate List*
(5) End For
(6) For IV_j *in Supervisee Candidate List*
(7) If $M_{i,j} < M_{\min}$ Then $k = j$ *//find the most stable supervisee*
(8) End For
(9) Return *matched pair*(IV_i, IV_k) *//i supervises k*

Algorithm 1: Supervisor Matching algorithm.

Table 1: Neighbor and supervision field.

Neighbor field		Supervision field	
$(x, y)_j$	Position of IV_j	$a'(k, j)$	IV_k's last availability received from IV_j
$(v_x, v_y)_j$	Speed of IV_j		
$UntrustDegree\,(i, j)$	UntrustDegree from IV_i to IV_j	$a'(j, k)$	IV_k's last availability sent to IV_j
$a(i, j)$	Last availability received from IV_j		
$a(j, i)$	Last availability sent to IV_j	$r'(k, j)$	IV_k's last responsibility received from IV_j
$r(i, j)$	Last responsibility received from IV_j		
$r(j, i)$	Last responsibility sent to IV_j	$r'(j, k)$	IV_k's last responsibility sent to IV_j
$CH_{cnvg,j}$	Cluster head converge flag for IV_j		
$SUPVE_j$	IV_j's supervisee		

The input of Algorithm 1 is IV's state tuples (position, speed). For IV_i, running this algorithm will work out a supervisee. For each IV_j in DSRC range, IV_i calculates mobility metric $M_{i,j}$ (lines (1)-(2)); the smaller the metric is, the more similar the two motions are. Thus, a small metric indicates a stable driving companion. If any IV_j has no supervisor and $UntrustDegree(i, j) \leq Threshold$ (this indicates that i does not tend to trust j), IV_i adds IV_j in Supervisee Candidate List (Line (3)-(4)). After that, IV_i chooses the most stable candidate (with the smallest mobility metric) to be the supervisee (lines (6)–(8)). Finally, a pair (IV_i, IV_j) is returned (line (9)).

3.3. Generating CH by Message Passing. We try to use a distributed algorithm to reach a consensus among large amounts of opinions. Each IV_i maintains a neighbor list N_i. As Table 1 shows, the list consists of N_i^j for each neighbor IV_j. Additionally, IV_i also maintains a supervision field for a supervisee IV_k.

Generating CH needs several iterations which are periodically triggered by time. Besides, broadcasting and supervising also need a synchronous clock. *Hello beacons* are broadcast and received to maintain local awareness.

Broadcast and Receive Hello Beacons Process is as follows:

(1) For every T_{hello}, each IV_j broadcast *hello beacon* is

$$\left\langle j, (x, y)_j, \left(v_x, v_y\right)_j, CH_j, SUPVE_j \right\rangle. \tag{4}$$

(2) Each receiving neighbor IV_i calculates $UntrustDegree(i, j)$ if they are traveling in the same direction.

(3) IV_i adds/updates N_i^j in its neighbor list:

$$\left\langle j, (x, y)_j, \left(v_x, v_y\right)_j, UntrustDegree\,(i, j), CH_j, SUPVE_j \right\rangle. \tag{5}$$

Availability and responsibility messages should be broadcast periodically. We define this period as $T_{message}$. Each IV_i will calculate $a(j, i)$ and $r(j, i)$ for each neighbor IV_j. This value is damped with the previous value stored in the neighbor list. IV_j then broadcasts $a(j, i)$ and $r(j, i)$ of all neighbors IV_j.

According to mutual supervisor model, the process of calculating $a(j, i)$ and $r(j, i)$ should be supervised. Each IV automatically chooses a supervisee by *Supervisor Matching algorithm*. Supervisor checks supervisee's calculation result and releases *alert* on condition that supervisee's message is suspicious. The process enhanced by mutual supervisor model is illustrated below.

Supervising and Message Passing Process. For every $T_{message}$, each IV_i will do the following:

(1) It will find a matching supervisee IV_k which is prepared for the next T_{round}'s iteration. If found in this $T_{message}$, it is claimed by $SUPVE_j$. If failed, it will try next $T_{message}$.

(2) If it hears an *alert* about IV_n, each IV_i will ignore IV_n's messages in this T_{round}.

(3) It will calculate responsibility $r(i, j)$ for each neighbor IV_j.

(4) It will update with damping factor and store: $r(i, j) = (1 - \lambda)r(i, j)_{new} + \lambda r(i, j)_{old}$.

(5) It will calculate availability $a(j, i)$ for each neighbor IV_j.

(6) It will update with damping factor and store: $a(j, i) = (1 - \lambda)a(j, i)_{new} + \lambda a(j, i)_{old}$.

(7) It will determine if itself is converged to CH: if $r(i, i) + a(i, i) > 0$, then set $CH_{cnvg,j}$.

(8) It will broadcast Responsibility and Availability array, $r(i, j)$ and $a(j, i)$.

(9) It will supervise IV_k: IV_i updates and calculates $r'(j, k)$ and $a'(j, k)$ for IV_k.

(10) IV_i listens to IV_k's messages: $r(k, j)$ and $a(j, k)$; if $|r(k, j) - r'(k, j)| > Threshold$ or $|a(k, j) - a'(k, j)| > Threshold$, then it will broadcast *alert* about IV_k.

$T_{message}$ must be small enough to allow algorithm converged within a T_{round}. We have injected a supervision mechanism into clustering process. Any node that broadcasts false availability and responsibility would very likely be discovered. The punishment to malicious nodes is twofold: first, its message would be ignored by neighbors through *alert*; second, a malicious behavior would be reported to CA.

In any $round_r$, there is a CH_{r-1} generated from $round_{r-1}$. CH_{r-1} will claim its role and broadcast *Final Message*, which represents CH's final evaluation to each cluster member VI_i:

$$FinalMessage = \{UntrustDegree\,(CH_{r-1}, i)\}. \quad (6)$$

FinalMessage is trustworthy since it is sent from CH, which is elected as "the most trustworthy node" by all group members. Built upon *FinalMessage*, intracluster trust management is relatively reliable to support IVs' collaborations.

4. Degrading Anomaly by Evidence Evaluation

Reputation-based method has been widely used in web service [22, 23] and cloud computing [24] to enhance system reliability and robustness. We believe this method could also improve system performance in Internet of Vehicles. In this scenario, IVs will observe and evaluate qualities of each other. Moreover, they form *evidences* and report them to CA. CA is supported by strong storage and computational resources, thus being capable of computing reputation from a global view. A global reputation is valuable for on-the-road IVs to choose potential collaborators. More importantly, reputation can be increased or degraded, as a system-level enforcement, to incent good behaviors as well as to punish bad ones.

IVs leverage "store-upload" mechanism in delivering evidences to a CA. Since RSUs are sparsely deployed, each IV would store evidences in its storage firstly and then upload them when moving into a RSU's service range. Evidence evaluation lies in the core of reputation. CA is able to make a conclusion on certain behavior by evaluating and merging different pieces of evidences from different individuals. Note that not all evidences are consistent, and not all evidences are trustworthy. For instance, in order to disturb reputation system, a malicious node may report false evidences.

To mathematically model evidence evaluation, assume CA has to decide among several basic behaviors $\beta_i \in \Omega$, based on K pieces of evidences $\{e_k^j\}$ to IV_j which are uploaded from different k IVs. Let B^j denote the final judgement on behavior type of IV_j. The following three methods are leveraged to get a consensus evaluation, with the ability to filter false evidences.

(A) Majority Voting. The final evaluation accords with the majority. Given counts of each type of observed behaviors, $count_i$, the behavior type of VI_j is defined by

$$B^j = \beta_{\text{argmax}(count_i)}. \quad (7)$$

(B) Weighted Voting. For each behavior, this method sums up all the votes value supporting this behavior. The votes are weighed by corresponding trust level r_k^i. Then, the type with the highest value is final evaluation:

$$B^j = \beta_{\text{argmax}((1/count_i)\sum r_k^i)}. \quad (8)$$

(C) Bayesian Inference. Among the data fusion techniques, Bayesian Inference (BI) is the most popular one used for trust building and managing. To use BI, the a priori probability of each action β_i is firstly assigned. A posterior probability of each action β_i is calculated given a set of evidences $e = \{e_1^j, e_2^j, e_3^j, \ldots, e_k^j\}$ using Bayes' theorem. For IV_j,

$$P\left[\beta_i \mid e\right] = \frac{P\left[\beta_i\right] \times \prod_{k=1}^{K} P\left[e_k^j \mid \beta_i\right]}{\sum_{l=1}^{I}\left(P\left[\beta_l\right] \times \prod_{k=1}^{K} P\left[e_k^j \mid \beta_l\right]\right)}. \quad (9)$$

Final consensus is the actions type with the maximum posterior probability:

$$B^j = \beta_{\text{argmax}(P[\beta_i|e])}. \quad (10)$$

Besides evidence evaluation, reputation evolution rule is another critical issue. An effective reputation system requires appropriate reputation evolving rules. We will discuss rules in Section 5.1.

5. Performance and Analysis

To evaluate performance of our scheme, we ran an extensive simulation in TransModeler with real map and high fidelity data. We use a map of urban area of San Antonio, USA. We feed real macroscopic traffic data, which are measured in critical roads and sections, to reconstruct real traffic scenario. We believe that macroscopic data could reflect traffic dynamic to a high extent. We do not simulate the wireless medium in this case since it is orthogonal to our evaluation. All simulations were performed with approximately 400 vehicles on a 6 miles' expressway. Five RSUs are sparsely deployed along the expressway as Figure 4. The DSRC range is set at

Table 2: Three basic behaviors.

β_i	Basic behavior type	UntrustDegree	Reputation change	Example
β_1	Life-critical events	0.9	Exponential degrade	2^7 (128) → 2^6 (64)
β_2	Efficiency reduction events	0.5	Linear degrade	128 → 120
β_3	Normality	0.1	No change	128 → 128

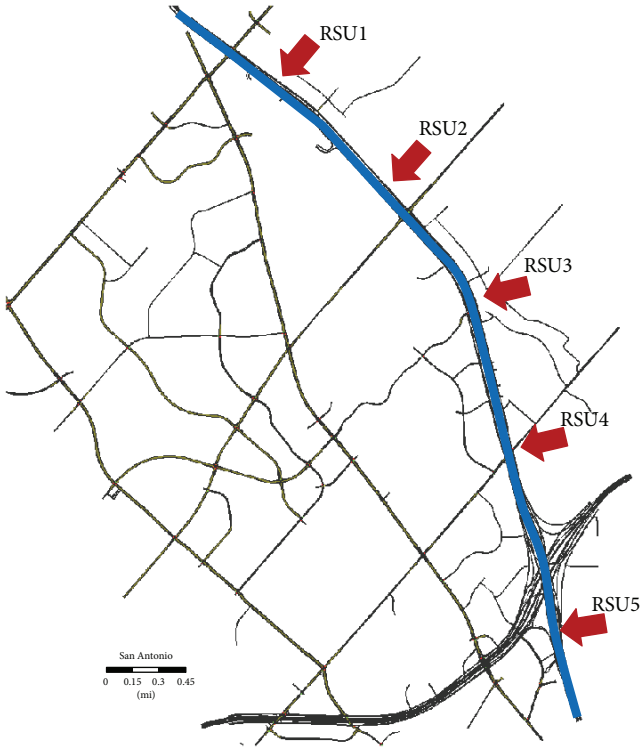

Figure 4: RSUs' deployment.

Table 3: Five behavior patterns.

Scenario number	Anomaly nodes behavior pattern		
	β_1	β_2	β_3
1	100%	0	0
2	0	100%	0
3	50%	50%	0
4	30%	50%	20%
5	20%	30%	50%

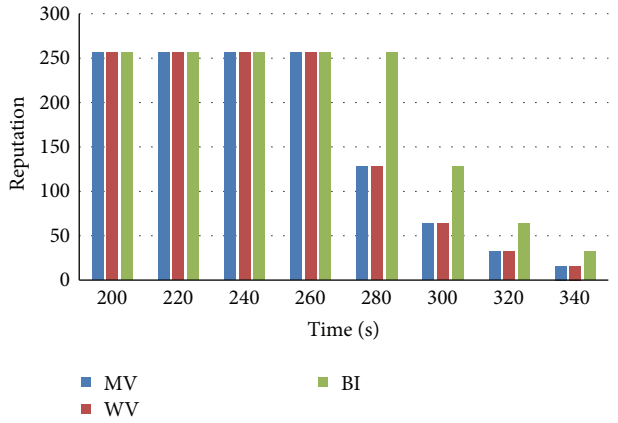

Figure 5: Effects of three merging techniques.

300 m. Each simulation ran for 600 s; however, only the last 400 s were used for performance metric calculations.

As noted earlier, an IV will be observed and evaluated by neighbor IVs. We use the example with three *Basic Behaviors* in Table 2. Each behavior causes different interactive trust. According to reputation evolution rules, one behavior deserves change in reputation.

To depict complex malicious/inappropriate behaviors, which are often mixed with different basic behaviors, we simulate several *behavior patterns* in Table 3. An anomaly node produces one behavior in every T_{round}. These patterns are simplified to make simulation feasible. We believe they could still well-reflect validity of our designed scheme.

5.1. The Effect of AP Algorithm. Ideally, AP clustering would generate a CH for every on-the-road vehicle. However, a small portion of vehicles, N_{left}, could be left alone when iterations are finished. There are two major reasons for these nodes: (1) the node could not find a converged CH candidate in its neighborhood and (2) the node itself is the CH but is the only member of cluster. Beside N_{left}, there are N_{in} nodes which form M normal clusters. In anomaly node-free simulation, several results are shown in Table 2. *Covered ratio* is a parameter describing how much the clustering results could cover the whole participants:

$$CoverRatio = \frac{N_{in}}{N_{in} + N_{left}}. \quad (11)$$

In anomaly simulation, several results are shown in Table 3. The simulations are ran several times so Figures 5, 6, 7, and 8 are averaged. A trade-off between *Covered Ratio* and *Cluster Member Number* could be found through Table 4. The higher the *Covered Ratio,* the lower the *Cluster Member Number*.

According to reference [18], *Damping Factor* is critical for convergence. Different *Damping Factors* result in different cluster outcomes. In general, a bigger *Damping Factor* leads to a relatively higher *Covered Ratio* and a lower *Cluster Member Number*. We recommend to set $\lambda \in [0.6, 0.8]$ so that algorithm tends to come out as an approximate but stable solution.

Another important parameter not mentioned in [18] is *Iteration Cycle*. Mathematically, the convergence of AP clustering is only influenced by *Damping Factor*, because

Table 4: The results of primary AP clustering.

Damping factor	Iteration cycle	DSRC range	Average cover-ratio	Average normal cluster number	Average member number	State
0.6	6	300 m	85.8%	26.5	13.2	Underdamping
0.7	5	300 m	38.9%	9.3	17.8	Underiteration
0.7	6	200 m	91.5%	46.7	8.5	Ok
0.7	6	300 m	94.3%	39.9	10.0	Ok
0.7	7	300 m	95.5%	85.5	4.8	Overiteration
0.8	6	300 m	74.3%	18.5	22.5	Overdamping

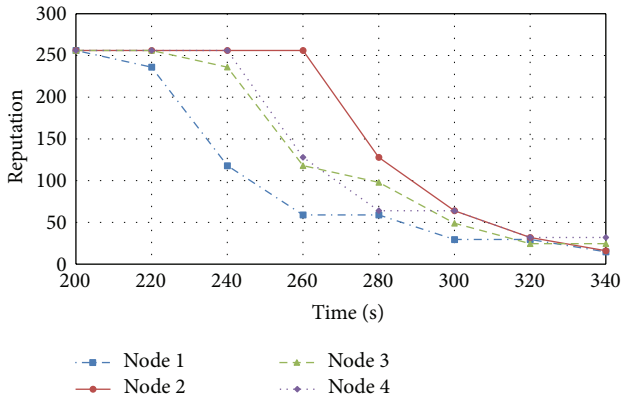

Figure 6: Reputation evolution of four anomaly nodes.

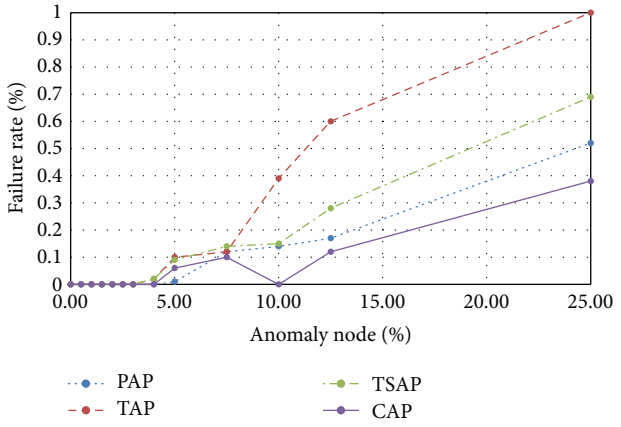

Figure 7: Percentage-Failure Rate curve.

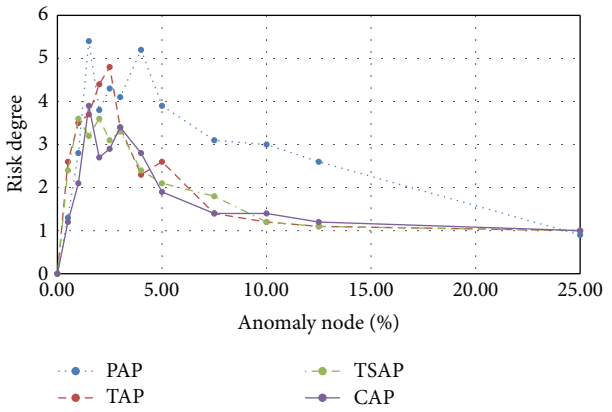

Figure 8: Percentage-Risk Degree curve.

the authors implicitly assume AP clustering could always have enough time for iteration. However, we have modified and applied this algorithm for anomaly detection where the communicating topology is constantly changing. Thus, the communication environment could not always provide plenty of time for clustering. So *Iteration Cycle* should be regarded as a critical parameter. If it is too short, clustering process will not be able to produce enough CHs to cover most nodes. On the other hand, if the cycle is too long, over-iteration will generate too many CHs. To conclude, modified AP clustering is oriented to real application other than a pure math problem and several parameters should be meticulously adjusted for real deployment, among which the most important ones are *Damping Factor* and *Iteration Cycle*.

We use two metrics to measure effectiveness of modified AP algorithm.

(1) Direct Influence. We define one *Failure* as an anomaly node elected to be CH; *Failure Rate* is to measure direct influence of one anomaly node, also called unsuccessful anomaly detection rate:

$$Failure\ Rate = \frac{Number_{anomaly\text{-}CH}}{Number_{anomaly\text{-}node}}. \quad (12)$$

(2) Indirect Influence. We define *Risk Degree* to feature how much potential influence an anomaly VI_i has when it is in one cluster:

$$Risk\ Degree = \left(1 - UntrustDegree\left(CH, i\right)\right) \times Number_{cluster\ members}. \quad (13)$$

If an anomaly node becomes CH, UntrustDegree is 0; otherwise, UntrustDegree is referred to CH's *Final Message*, which expresses CH's opinion of each node. *Risk Degree* could feature the indirect influence of an unqualified node according to its role (CH or member) and UntrustDegree.

Table 5: Prior distribution of behaviors and observed behaviors.

Observed	Behavior		
	β_1	β_2	β_3
β_1	0.7	0.15	0.1
β_2	0.2	0.7	0.1
β_3	0.1	0.15	0.8

For example, if an unqualified is admitted into a 20-member cluster, and CH's final message claims its UntrustDegree is 0.5, then its *Risk Degree* is 10. If it is admitted into a 5-member cluster, Risk Degree is 2.5. The later risk is much smaller because the anomaly node has fewer potential partners and thus may have fewer threats.

5.2. Comparison of Four Models. Our performance evaluation is based on four models: Primary AP model (PAP), Tempering AP model (TAP), Tempering&Supervising AP model (TSAP), and Converged AP model (CAP). PAP is directly derived from AP clustering algorithm. TAP models the clustering scenario where anomaly nodes could temper/disturb message passing process. In short, TAP considers tempering/disturb behaviors over PAP. To alleviate influence of tempering/disturbing, TSAP model injects *Mutual Supervision Model* into PAP to identify anomaly nodes. Finally, CAP is a converged model which enhanced TSAP with historical reputation.

As a converged model, CAP combines historical reputation with real-time cluster-based trust. CA collects uploaded evidences from on-road vehicles and uses three techniques to fuse evidences: (1) Majority Voting, (2) Weighted Voting, and (3) Bayesian Inference. For Bayesian Inference, the prior distribution of behaviors and observed results are defined in Table 5.

We assume that anomaly nodes use a random reporting strategy, which means they generate evidences randomly regardless of what other nodes really have done. Normal nodes will always report true evidences. Figure 5 describes the effects of different evidence merging techniques. In this simulation, three techniques are almost equally effective. However, MV and WV are more suitable for data merging since they have less computation overhead. Figure 6 shows four anomaly nodes' reputation evolves in system. Anomaly nodes would be distinguished and punished by CA.

In a process of iteration, IVs with larger values of Self-UntrustDegree are more likely to be chosen as CH. These values are "preferences." In PAP/TAP/TSAP, IV_n preference is set as median of $UntrustDegree(i, n)$. However, in CAP where historical reputation is considered by algorithm, IV_n's preference is calculated by

$$Preference_n = PunishFactor_n \times Median\left(UntrustDegree\left(i, n\right)\right). \quad (14)$$

When IV_n's reputation is low, $PunishFactor_n \in [1, 2]$ is big; preference therefore becomes small (preference is a negative real number), indicating IV_n is not suitable to be CH.

Figure 7 shows the comparison of four models. We set unqualified node percentage as variable. We simulate with different percentages ranges from $[0, 0.25]$ because too high percentage is not realistic.

Generally, when anomaly nodes percentage is low (≤5%), *Failure Rate* is 0%. As percentage goes up, *Failure Rate* also goes higher. TAP is a model with tempering/disturbing and no supervision mechanism, so it performs worse than PAP (no tempering/disturbing) and TSAP (tempering/disturbing, supervision model). In contrast, CAP is a converged model (tempering/disturbing, supervision model, and reputation) with a strong defense to anomaly nodes, so it shows the highest robustness among four models. Furthermore, either model could limit failure rate below 1% even when anomaly nodes percentage is up to 25%.

Risk Degree features how much potential influence an anomaly node has when it is in one cluster. Figure 8 shows the *Anomaly Percentage-Risk Degree* curve for four models. Risk Degree is firstly low when *Anomaly Node Percentage* is low. However, it suddenly goes to peak when *Percentage* slightly increases. Finally, it stably declines with increasing *percentage*. The explanation for this curve is as follows: (1) when *Percentage* is very low (≤1%), tempering/disturbing is few, and anomaly nodes therefore are easily distinguished by normal nodes. As a result, anomaly nodes are very likely to be left alone. That is, they are excluded from big clusters by AP algorithm. So the overall *Risk Degree* is low. (2) When *Percentage* goes higher but not that high (≤5%), this percentage still indicates a "safe environment"; IVs tend to form "big clusters." However, with more anomaly nodes percentage, more anomaly nodes have chances to join big clusters by more tempering/disturbing. According to formula (13), even one anomaly node in a big cluster would cause a big risk degree. (3) When *Percentage* increases over 5%, our algorithms tend to be conservative and form "small clusters," which have fewer cluster members. Fewer members render lower *Risk Degree*. According to Figure 8, CAP could limit Risk Degree under 4, demonstrating that our trust management is effective on risk control.

6. Conclusion and Future Work

Our system aims to build a trustworthy platform to detect abnormal vehicles. To this end, we modified *Affinity Propagation* to elect a most trustworthy node, called cluster head, among vehicles. CH maintains trust management during a period until a new CH is elected. We also considered that AP is executed in a distributed manner thus easily tempered by malicious nodes. So we presented a mutual supervision model to tackle tempering behaviors. Lastly, we blend another component, CA, into our system. CA consisted of servers and sparse RSUs and is able to provide historical reputation for better decision-making. Overall, this trust management system could detect and filter anomaly nodes.

In the future, great efforts are needed on both the in-vehicular system and RSUs to strengthen our secure system. These efforts include deploying mobile and local CA using cloud computing techniques, improving intelligence of

mutual trust evaluation, and reducing overhead of detection process.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the National High-Tech Research and Development Program (863) of China under Grant no. 2012AA111601 and 2015 Construction of key discipline under no. 700200253.

References

[1] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, 2014.

[2] Q. Yuan, Z. Liu, J. Li, J. Zhang, and F. Yang, "A traffic congestion detection and information dissemination scheme for urban expressways using vehicular networks," *Transportation Research Part C: Emerging Technologies*, vol. 47, no. 2, pp. 114–127, 2014.

[3] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds," in *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 241–246, Seoul, Republic of Korea, March 2014.

[4] Y. Leng and L. Zhao, "Novel design of intelligent internet-of-vehicles management system based on cloud-computing and Internet-of-Things," in *Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT '11)*, vol. 6, pp. 3190–3193, Harbin, China, August 2011.

[5] D. Weyns, T. Holvoet, and A. Helleboogh, "Anticipatory vehicle routing using delegate multi-agent systems," in *Proceedings of the 10th IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 87–93, IEEE, Seattle, Wash, USA, October 2007.

[6] V. Gligor and J. M. Wing, "Towards a theory of trust in networks of humans and computers," in *Security Protocols XIX*, Lecture Notes in Computer Science, pp. 223–242, Springer, Berlin, Germany, 2011.

[7] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proceedings of the 17th IEEE Symposium on Security and Privacy*, pp. 164–173, IEEE, Oakland, Calif, USA, May 1996.

[8] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and Sensor Networks (SASN '05)*, pp. 11–21, ACM, November 2005.

[9] S. D. Ramchurn, D. Huynh, and N. R. Jennings, "Trust in multi-agent systems," *The Knowledge Engineering Review*, vol. 19, no. 1, pp. 1–25, 2004.

[10] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1912–1920, Phoenix, Ariz, USA, April 2008.

[11] C. Chen, J. Zhang, R. Cohen, and P.-H. Ho, "A trust modeling framework for message propagation and evaluation in VANETs," in *Proceedings of the 2nd International Conference on Information Technology Convergence and Services (ITCS '10)*, IEEE, August 2010.

[12] K. Rostamzadeh, H. Nicanfar, N. Torabi, S. Gopalakrishnan, and V. C. M. Leung, "A context-aware trust-based information dissemination framework for vehicular networks," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 121–132, 2015.

[13] N. Haddadou and A. Rachedi, "DTM2: adapting job market signaling for distributed trust management in vehicular ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '13)*, pp. 1827–1832, IEEE Press, Budapest, Hungary, June 2013.

[14] S. Wang, C. Fan, C.-H. Hsu, Q. Sun, and F. Yang, "A vertical Handoff method via self-selection decision tree for internet of vehicles," *IEEE Systems Journal*, vol. 99, pp. 1–10, 2014.

[15] R. G. Machado and K. Venkatasubramanian, "Short paper: establishing trust in a vehicular network," in *Proceedings of the IEEE Vehicular Networking Conference (VNC '13)*, pp. 194–197, IEEE, December 2013.

[16] D. Huang, Z. Zhou, X. Hong, and M. Gerla, "Establishing email-based social network trust for vehicular networks," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC '10)*, IEEE, Las Vegas, NVev, USA, January 2010.

[17] D. Huang, X. Hong, and M. Gerla, "Situation-aware trust architecture for vehicular networks," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 128–135, 2010.

[18] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[19] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, IEEE, Honolulu, Hawaii, USA, December 2009.

[20] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[21] T. Lei, S. Wang, J. Li, I. You, and F. Yang, "Detecting and preventing selfish behaviour in mobile ad hoc network," *The Journal of Supercomputing*, 2015.

[22] S. Wang, Z. Zheng, Z. Wu, M. R. Lyu, and F. Yang, "Reputation measurement and malicious feedback rating prevention in web service recommendation systems," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 755–767, 2015.

[23] S. Wang, L. Huang, C.-H. Hsu, and F. Yang, "Collaboration reputation for trustworthy Web service selection in social networks," *Journal of Computer and System Sciences*, vol. 82, no. 1, pp. 130–143, 2016.

[24] K. Hwang, S. Kulkarni, and Y. Hu, "Cloud security with virtualized defense and reputation-based trust mangement," in *Proceedings of the 8th IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC '09)*, pp. 717–722, IEEE, Chengdu, China, December 2009.