A Trust-Based Model for Security Cooperating in Vehicular Cloud Computing

doi: 10.1155/2016/9083608

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 9083608, 22 pages
http://dx.doi.org/10.1155/2016/9083608

Research Article

A Trust-Based Model for Security Cooperating in Vehicular Cloud Computing

Zhipeng Tang,[1] Anfeng Liu,[1] Zhetao Li,[2] Young-june Choi,[3] Hiroo Sekiya,[4] and Jie Li[5]

[1]*School of Information Science and Engineering, Central South University, Changsha 410083, China*
[2]*College of Information Engineering, Xiangtan University, Xiangtan 411105, China*
[3]*Department of Software, Ajou University, Suwon 443749, Republic of Korea*
[4]*Graduate School of Advanced Integration Science, Chiba University, Chiba 263-8522, Japan*
[5]*Department of Computer Science, University of Tsukuba, Tsukuba 305-8573, Japan*

Correspondence should be addressed to Zhetao Li; liztchina@gmail.com

Received 22 June 2016; Accepted 7 September 2016

Academic Editor: Miao Wang

VCC is a computing paradigm which consists of vehicles cooperating with each other to realize a lot of practical applications, such as delivering packages. Security cooperation is a fundamental research topic in Vehicular Cloud Computing (VCC). Because of the existence of malicious vehicles, the security cooperation has become a challenging issue in VCC. In this paper, a trust-based model for security cooperating, named DBTEC, is proposed to promote vehicles' security cooperation in VCC. DBTEC combines the indirect trust estimation in Public board and the direct trust estimation in Private board to compute the trust value of vehicles when choosing cooperative partners; a trustworthy cooperation path generating scheme is proposed to ensure the safety of cooperation and increase the cooperation completion rates in VCC. Extensive experiments show that our scheme improves the overall cooperation completion rates by 6~7%.

1. Introduction

Many new applications have been raised on the vehicular technology by V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) communications [1–5]. Recently, several researches related to the combination of cloud computing and vehicular networks [4, 5] are proposed. A Platform as a Service (PaaS) model provides cloud services for mobile vehicles [6–8]. Hussain et al. describe architectures of Vehicular Clouds (VC), namely, Vehicles using Clouds (VuC) and Hybrid Clouds (HC), in which vehicles play roles of cloud service providers and clients, respectively [9]. Vehicular Cloud Computing (VCC) is one of the most promising paradigms [1, 4, 9–11]. VCC, which consists of vehicles cooperating the resources of computing, has a significant impact on applications [9, 11]. However, VCC is different from the traditional cloud infrastructure and requires a sophisticated security and privacy protection approach because the legitimate users and attackers have the same privileges [1, 4, 12–19].

One of the promising applications in VCC is performing tasks by vehicles' cooperation. This application, which is more difficult than the existing ones in depth and breadth, has important significance: in the traditional Delay Tolerant Network (DTN) and Peer-to-Peer Network, it can only disseminate information. But, in VCC, not only can this application disseminate information, but also it can do more practical work, such as delivering packages, luggage, and credentials [1, 4, 10, 11].

Taxi network is a typical scenario of VCC. Each taxi in this scenario is regarded as a vehicle which can share information by communicating in a point-to-point manner and accessing internal broadcast by communication devices. From the perspective of traditional view, taxis can be modeled as mobile nodes in DTN. However, more applications can be achieved when modeled in VCC. In particular, when performing a task, vehicle can apply for cooperating with several vehicles, which will improve service quality and reduce resource consumption. Listed below are several concrete examples.

(a) Vehicle A has received a user request and promised to pick up a passenger in street S at timestamp T, but traffic jam has made it impossible for vehicle A to finish this task. In this situation, vehicle A can request vehicle B to perform this task. There are two preconditions when selecting vehicle B. First, vehicle B has ability to fulfil this task; second, vehicle B should be trustworthy. (b) Vehicle A has received a user request and promised to perform a task which cannot be finished by itself individually, such as picking up a tourist group. In this situation, vehicle A should select n reliable vehicles and send cooperation request to them for performing this task together. (c) Vehicle A has received a user request and promised to deliver an important package to person P in street S. In real scenario, this task has to be performed by cooperation of several vehicles. For instance, first, vehicle A delivers this package to vehicle B; then vehicle B delivers it to vehicle C; finally, vehicle C delivers it to person P. This process forms a cooperation path. In order to guarantee the safety of the package, how to select trustworthy vehicles in cooperation path is a challenging problem.

The examples listed above can be summarized as the following application scenario: vehicle A has received a user request for performing certain tasks. These tasks not only include the traditional applications in DTN [20–22], such as relaying information, but also can be extended to physical request, such as delivering objects. However, for some reason, vehicle A cannot fulfil the task by itself. In order to finish this task, it sends request for cooperation to n vehicles which are willing to offer help. In the cases when vehicles which received the cooperation request still cannot fulfil the tasks by themselves, they will further send this cooperation request to other vehicles recursively to request from them to offer help to finish the remaining tasks, which forms a nontrivial cooperation path.

Figure 1 illustrates a concrete example of the summarized application scenario: when vehicle A receives a user request of delivering a package to person P in street S as soon as possible. If vehicle A can finish this whole task by itself, it will provide services to user directly, which forms a trivial cooperation path whose length is 1 hop, namely, a hop from user to vehicle A. If vehicle A cannot deliver the package to person P directly, vehicle A will finish what it can do and then send messages to other vehicles to ask if they are willing to cooperate to perform the rest of the task. Vehicles that give positive response form a set Υ. Vehicle A will select several trustworthy vehicles in set Υ and send cooperation request to them. Assume vehicle B has received cooperation request from vehicle A. If vehicle B can finish the whole task by itself, it will provide services to requestor vehicle A directly. If vehicle B still cannot finish the task by itself, it will do what it can do and further recursively send the cooperation request to other vehicles just like vehicle A. This recursive process forms a nontrivial cooperation path. Every cooperation path corresponds to a solution to user request.

There are several challenges in this application scenario. (a) The first challenge is lack of trust information. How to choose trustworthy vehicles is a vital problem in this application scenario. However, there are thousands of vehicles in a metropolis. It is unrealistic for a vehicle to have

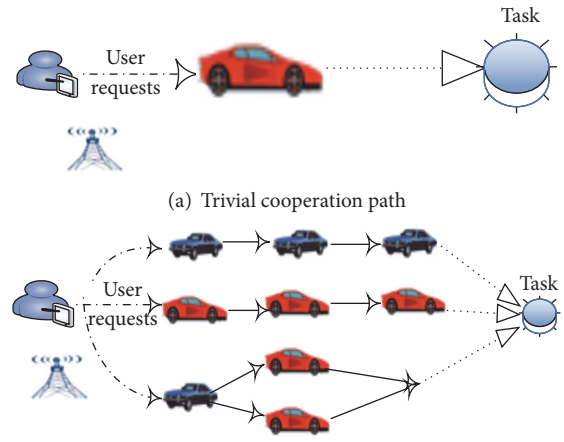

Figure 1: Cooperation path.

trust information of all vehicles in the metropolis. In fact, for a certain vehicle, the reliability of most of vehicles is unknown. When a vehicle needs cooperation to perform a task, such as delivering a package, the phenomenon of lack of trust information makes choosing trustworthy cooperative vehicles difficult. (b) The second challenge is ensuring the safety and success of tasks. In traditional communication network, such as DTN, we can encrypt information to ensure the safety and privacy. Even if the encrypted information is destroyed by attackers, we still can retransmit this information to ensure the task's reliability [20–26]. Things are different in VCC; physical objects can also be delivered in this paradigm. Irreversible loss will be made if the physical objects are ruined by malicious vehicles.

In this paper, a trust-based model is proposed to promote the secure cooperation in VCC. Listed below are the contributions of this paper.

(1) A double board based trust estimation and correction (DBTEC) scheme is proposed to predict the reliability of vehicles and guide the selection of trustworthy cooperative vehicles in a more effective manner. In traditional scheme, vehicles use information acquired in direct interactions with other vehicles to update the trust information of other vehicles. But in DBTEC scheme, Public board is introduced to enrich the method of acquiring trust information. Every vehicle stores the service quality and trust information of other vehicles, which are acquired in the direct interactions with other vehicles, in their own storage, called Private board. In addition, they use Public board, which stores public estimated service quality for other vehicles reported by all vehicles in cloud to update and correct the trust information stored in Private board. The method of updating and correcting trust information from Public board, called trust value estimation model, is based on the following inference: the information acquired from direct interaction is trustworthy; vehicles can use this information as touchstone to confirm if a certain vehicle is trustworthy. Then, based on the public estimated service quality related to the trustworthy vehicle in Public board, vehicles can update and correct the trust information

of other vehicles in Private board and use the revised trust information to guide their future selection of cooperative partners.

(2) A new method of constructing cooperation path is proposed in this paper. In traditional scheme, the cooperation path is fixed once it is constructed. This static method is not suitable for VCC. In this paper, we propose a dynamic cooperation path construction scheme. In the proposed scheme, every vehicle dynamically searches and selects cooperative vehicles and constructs new node in cooperation path by analyzing the feedback of detections. The new vehicles will recursively repeat this process until finishing the task.

(3) Extensive theoretical analysis and simulation have been made to prove the effectiveness of this paper from aspects of security and reliability.

The rest of this paper is organized as follows. In Section 2, related works are reviewed. In Section 3, the system model, threat model, and problem statement are described. In Section 4, the DBTEC schemes are proposed. Section 5 gives the analysis of experimental results. We conclude this paper in Section 6.

2. Related Work

Extensive researches have been done on the topic of trust computing and inference [27–30] and they have been applied to various networks, such as Peer-to-Peer (P2P) file-sharing networks [31, 32], service network [1, 9, 18], wireless sensor networks (WSNs) [30], crowd sensing network [3], and social networks [28]. The aim of trust computing and inference is to select cooperative partner using computed trust value information [29]. Kamvar et al. [31] proposed trust computing and inference scheme in the Peer-to-Peer (P2P) file-sharing networks based on historical uploads, which is called EigenTrust. Inferring trust information through historical behaviors is a common method used in networks. In EigenTrust scheme, to encourage legitimate and trustworthy behaviors and improve the network's overall performance, some privilege is given to trustworthy objects. The main difficulties of EigenTrust are that, when applying it to distributed network, it is difficult to share trust information with others. This proposal mainly concentrates on P2P file-sharing networks. However, in a dynamic environment, such as vehicular ad hoc networks (VANETs), this proposal is not feasible.

Haddadou et al. give a dynamic solution based on reputation model for vehicles in [27], which differs from the solution in [31]. The basic idea of [27] is to add a category criterion to drivers.

However, the amount of trust information acquired in direct interactions is limited. In a large network, the number of nodes can be up to thousands. So the trust information acquired from direct interactions is sparse in that network. Judging the reliability of vehicles only using direct interactions will lead to cold start problem. There are several definitions of cold start. The main idea of cold start is that when a new object enters the network, because of the deficiency of trust information acquired from interactions, it is hard to judge if a vehicle is malicious, which makes choosing a right cooperative partners difficult [28]. To overcome the cold start problem, researchers introduce the concepts of direct trust information and indirect trust information. Direct trust information is acquired in the direct interactions between two objects. Indirect trust information is the trust information inferred from other objects' recommendation trust information. For example, object A has no direct interactions with object C, but object A has directly interacted with object B. Assuming that object A's trust value to object B is $P_{A\to B}$ and object B recommends object C to object A with trust value $R_{A\to B}$, object A can infer that the trust value to object C is $R_{A\to C} = P_{A\to B} \times R_{A\to B}$ from the recommendation trust information. Combining direct trust information and indirect trust information enhances the computation of trust value [28], but how to effectively compute the trust value is a complicated issue, which needs an extensive research.

The traditional application in VCC is disseminating information. For example, Rostamzadeh et al. propose a safe and reliable trust-based framework for disseminating information in vehicular networks [29]. With the advance of crowd sensing network, Internet of Vehicles, and Internet of Everything, delivering physical objects is becoming an emergent application in society. The safety and reliability of delivering physical objects are important requirement in this application, which becomes a key issue in research. This paper discusses this issue in detail.

3. The System Model and Problem Statement

3.1. System Model. Suppose that there are n registered vehicles. $\phi = \{V_1, V_2, \ldots, V_n\}$ is the set of vehicles. All vehicles will move randomly in a limited area.

There are two kinds of service requests in VCC: user requests and cooperation requests. The major difference between them is that user requests are generated by users, but cooperation requests are generated by vehicles. The following paragraphs describe these two kinds of requests.

Typical instances of user requests include delivering packages, picking up passengers, or tourist group with minimized costs. Vehicles can accept user requests and provide services to requestors for some payment. Once vehicles' accepted user requests cannot be finished by themselves, they will select several trustworthy vehicles which are willing to provide services and send cooperation request to them.

Once those vehicles receive cooperation requests, they will cooperate to provide services together. These vehicles still may not be able to fulfil the tasks by themselves and further send cooperation requests to other trustworthy vehicles recursively. This recursive process will form a nontrivial cooperation path (see Figure 1(b)).

All cooperation path forms set $\mathfrak{C} = \{\mathfrak{c}_1, \mathfrak{c}_2, \ldots, \mathfrak{c}_s\}$ in which $\mathfrak{c}_i$ is a trivial/nontrivial cooperation path. $|\mathfrak{C}| = s$ is the number of cooperation paths in VCC. The length of cooperation path $\mathfrak{c}_i$ is $|\mathfrak{c}_i|$, which is equal to the number of cooperation requests generated to finish a task. A cooperation path can be subdivided to many subcooperation paths, whose starting nodes are one of the nodes in the paths and ending nodes are the original paths' ending nodes; this concept will be used in Section 4.4.

The quality of service (QoS) can be modeled as a value between 0 and 1 called service quality. Different vehicles can provide different quality of service. For vehicle V_i, its service quality is Q_i. The set of service quality of all vehicles is $Q = \{Q_1, Q_2, \ldots, Q_n\}$.

Vehicles in Φ can be categorized into two types: normal vehicles and malicious vehicles. Malicious vehicles will use various means to strive for the opportunities of providing services, such as reporting mendacious trust value or service quality and colluding with other malicious vehicles. Once malicious vehicles get the opportunities, they will screw the service requests up in various manners, such as colluding with other malicious vehicles to provide low-quality services or destroy packages, to disrupt the network, and to benefit themselves. Assume that the first h vehicles in Φ are malicious, which consist of set $M = \{V_1, V_2, \ldots, V_h\}$. The remaining vehicles are normal, which consist of set $N = \{V_{h+1}, V_{h+2}, \ldots, V_n\}$. Obviously, $M \cup N = \phi$.

In order to prevent malicious vehicles from disrupting the network, normal vehicles should avoid sending cooperation requests to them. They store the estimated trust value and estimated service quality for other vehicles in storage, called Private board, and use this information to guide the selection of trustworthy vehicles when sending cooperation requests. As will be illustrated in Section 4.3, the Private board of vehicle V_i can be modeled by two sets:

$$E^i = \left\{E_1^i, E_2^i, \ldots, E_n^i\right\},$$
$$R^i = \left\{R_1^i, R_2^i, \ldots, R_n^i\right\}, \quad (1)$$

where E_j^i is the estimated service quality of V_j recorded by V_i and R_j^i is the estimated trust value of V_j recorded by V_i. Several timestamps (T^i, T_i^{col}, and T_i^{row}) are also recorded to trace the recording time.

Besides Private board, In DBTEC schemes, all vehicles can access a public cloud storage space, called Public board, anywhere and selectively report their estimated service quality for other vehicles to it. Vehicles can use the information in Public board to update the estimated trust value stored in Private board. As will be illustrated in Section 4.3, the Public board can be modeled by two $n \times n$ matrices, E' and T', where E' records the public estimated service quality reported by vehicles and T' records the reporting timestamps.

Note that estimated service quality is selectively reported to Public board, which means some service quality information may not be updated to Public board. Several reasons may result in this phenomenon: privacy protection, avoiding revenge, and network interruption.

3.2. Threat Model. There are h malicious vehicles in VCC, which consist of set $M = \{V_1, V_2, \ldots, V_h\}$; four possible malicious behaviors are listed below. These malicious behaviors can be combined to form sophisticated malicious models, such as providing unstable services. In Section 5.3, five malicious models are introduced to analyze the performance of DBTEC schemes.

(1) Report False Self-Estimated Service Quality to Public Board When Registering. High-quality service is wanted by users. Normal vehicles and users tend to choose vehicles providing high-quality services as cooperative partners. Normal vehicle V_i reports true service quality Q_i it can provide, which is called self-estimated service quality, to Public board when registering. Malicious vehicles can deceive normal vehicles by reporting mendacious self-estimated service quality to Public board; this deception method is effective especially in the stage of cold start, in which trust information is deficient.

(2) Slander Normal Vehicles. As described above, normal vehicles and users tend to choose vehicles providing high-quality services as cooperative partners. Slander normal vehicles by reporting estimated service quality lower than normal level to Public board will reduce the probability that normal vehicles get opportunities of providing services, which increase the malicious vehicles' opportunity indirectly.

Generally speaking, this can be regarded as a kind of collusion attack since all malicious vehicles can benefit from cooperatively slandering normal vehicles and acquire much more opportunities to provide services.

(3) Collude with Malicious Vehicles by Praising Malicious Partners. This is a stronger collusion attack than the previous one since it has direct impacts on confusing normal vehicles. It praises malicious vehicles by reporting estimated service quality above their normal level to Public board, which can directly increase malicious vehicles' opportunities of providing services.

(4) Malicious Vehicles Camouflage Themselves as Normal Vehicles by Acting Like Them in Most of Time. Malicious vehicles can pretend to be normal vehicles by behaving just like them and provide unstable services. In this malicious scenario, the malicious vehicles behave normally generally. However, sometimes they will act some malicious behavior to benefit themselves. Because of the camouflage, this attack is hard to find.

3.3. Problem Statements. The application scenario considered in this paper is as follows: in Vehicular Cloud Computing (VCC), vehicles will receive user's service requests and provide services to them. In the process of providing services to users, if vehicles can finish the task, they will provide services directly to users, which forms a trivial cooperation path whose length is 1 hop (see Figure 1(a)). But vehicles may not be able to finish the task by themselves for some reason. In this case, vehicles will choose several trustworthy vehicles which are willing to offer help and send cooperation request to them. Vehicles which receive cooperation request still may not be able to completely finish the task by themselves. They will recursively send cooperation requests to other vehicles until the task is finished. The recursive process of completing tasks forms a nontrivial cooperation path (see Figure 1(b)). A cooperation path corresponds to a solution to user request in this application scenario. Specifically, the cooperation path is trivial in the case when vehicle which receives the user

request can finish the user request directly and no further cooperation happened. The key challenge of this application scenario is how to select vehicles, which can guarantee the success of the service and maximize the quality of service for cooperation.

In the process of cooperation, vehicles may wrongly choose malicious vehicles for cooperation, which will lead to the failure of the cooperation. We refer to selecting a vehicle to cooperate as a choice. A wrong choice means selecting a malicious vehicle for cooperation. A right choice means selecting a normal vehicle for cooperation. If there exists a wrong choice in a cooperation path, we say this cooperation path is failed. There are three aims in the application scenario to overcome the challenge described above.

(1) Minimize Failure Rate of Cooperation. Assume that all cooperation paths in VCC form set $\mathfrak{C} = \mathfrak{C}_Y \cup \mathfrak{C}_N$, where $\mathfrak{C}_Y$ is the set of successful cooperation paths and $\mathfrak{C}_N$ is the set of failed cooperation paths. So the number of cooperation paths is $|\mathfrak{C}|$ and the number of failed cooperation paths is $|\mathfrak{C}_N|$. The failure rate of cooperation is defined as $\theta_{\mathfrak{C}}$ and we should minimize it:

$$\theta_{\mathfrak{C}} = \frac{|\mathfrak{C}_N|}{|\mathfrak{C}|}. \tag{2}$$

(2) Minimize the Failure Rate of Choices. Similarly, assume that all choices in VCC form set $\mathfrak{c} = \mathfrak{c}_Y \cup \mathfrak{c}_N$, where $\mathfrak{c}_Y$ is the set of right choices and $\mathfrak{c}_N$ is the set of wrong choices. So the number of choices is $|\mathfrak{c}|$ and the number of wrong choices is $|\mathfrak{c}_N|$. The failure rate of choices is defined as $\theta_{\mathfrak{c}}$ and we should also minimize it:

$$\theta_{\mathfrak{c}} = \frac{|\mathfrak{c}_N|}{|\mathfrak{c}|}. \tag{3}$$

(3) Maximize Quality of Service of All Cooperation Paths. We define the quality of service of a cooperation path as the minimum service quality provided by vehicles in the cooperation path. This definition is reasonable because of the Cannikin law. Assume that the quality of service of cooperation path i is $\mathbb{Q}_i$, where $i \in \{1, 2, \ldots, |\mathfrak{C}|\}$. Therefore the total quality of services is $\sum_{i=1}^{|\mathfrak{C}|} \mathbb{Q}_i$. We should maximize the average service quality of cooperation path:

$$\frac{\sum_{i=1}^{|\mathfrak{C}|} \mathbb{Q}_i}{|\mathfrak{C}|}. \tag{4}$$

In general, we can combine the above three optimization requirements and try to find a scheme which satisfies the following three formulas together in this application scenario of VCC:

$$\begin{aligned} &\text{minimize} \quad \theta_{\mathfrak{c}} = \frac{|\mathfrak{c}_N|}{|\mathfrak{c}|}, \\ &\text{minimize} \quad \theta_{\mathfrak{c}} = \frac{|\mathfrak{c}_N|}{|\mathfrak{c}|}, \\ &\text{maximize} \quad \frac{\sum_{i=1}^{|\mathfrak{C}|} \mathbb{Q}_i}{|\mathfrak{C}|}. \end{aligned} \tag{5}$$

Notations describes some important notations used throughout this paper.

4. DBTEC Schemes

4.1. Overview. The main contribution of DBTEC schemes is to combine Public board with Private board to guide the selection of cooperative vehicles. In traditional scheme, vehicles can only acquire trust information from direct interactions with other vehicles [28]. Unlike the traditional scheme, in DBTEC schemes, vehicle A not only uses the information acquired in direct interaction but also uses trustworthy vehicles' public estimated service quality stored in Public board. When vehicle A needs to cooperate, it will choose a cooperative partner. DBTEC schemes use the information stored in Public board, which is a public information storage stored in cloud, to update the estimated trust value stored in vehicle A's Private board and then uses updated Private board to further guide the selection of proper cooperative partners. In DBTEC schemes, vehicles with high estimated trust value in vehicle A's Private board are called trustworthy. The trustworthy vehicles' public estimated service quality stored in Public board is just like the touchstone used to test whether an unfamiliar vehicle is malicious. When vehicle A trusts vehicle B, DBTEC will check public estimated service quality for all vehicles reported by vehicle B in Public board and all estimated service quality for vehicle B reported by all vehicles to find the inconsistency and use the inconsistency to find malicious vehicles.

Compared with traditional scheme, DBTEC scheme has major advantages. One of them is overcoming the problem that trust information is deficient in the stage of cold start. In the stage of cold start, vehicles do not have enough trust information to guide the selection of cooperative partners, which will significantly increase the probability that malicious vehicles get the opportunity of providing services. DBTEC scheme uses the trust information we already have as a touchstone to check the consistency and inconsistency in Public board and further updates the Private board's trust information. This process is just like diffusion of trust information; vehicles will get a lot of indirect information from the process, which will overcome the problem that trust information is deficient and increase the accuracy rate of selecting right vehicles.

Described below are the DBTEC schemes from high level.

As described in system model, there are n vehicles in VCC. Different vehicles can provide different quality of service. For vehicle V_i, its service quality is Q_i. Vehicles can be categorized into two types: malicious vehicles and normal vehicles. There are h malicious vehicles in VCC. Public board is a public cloud storage which can be accessed by vehicles anywhere. All vehicles can report their estimated service quality for other vehicles to Public board. All vehicles store a Private board in which they keep their own estimated trust value and estimated service quality information for other vehicles.

For a normal vehicle V_i, when vehicle V_i receives a user request, it will check if it can be done by itself; if not, it will search for vehicles which are willing to perform this task,

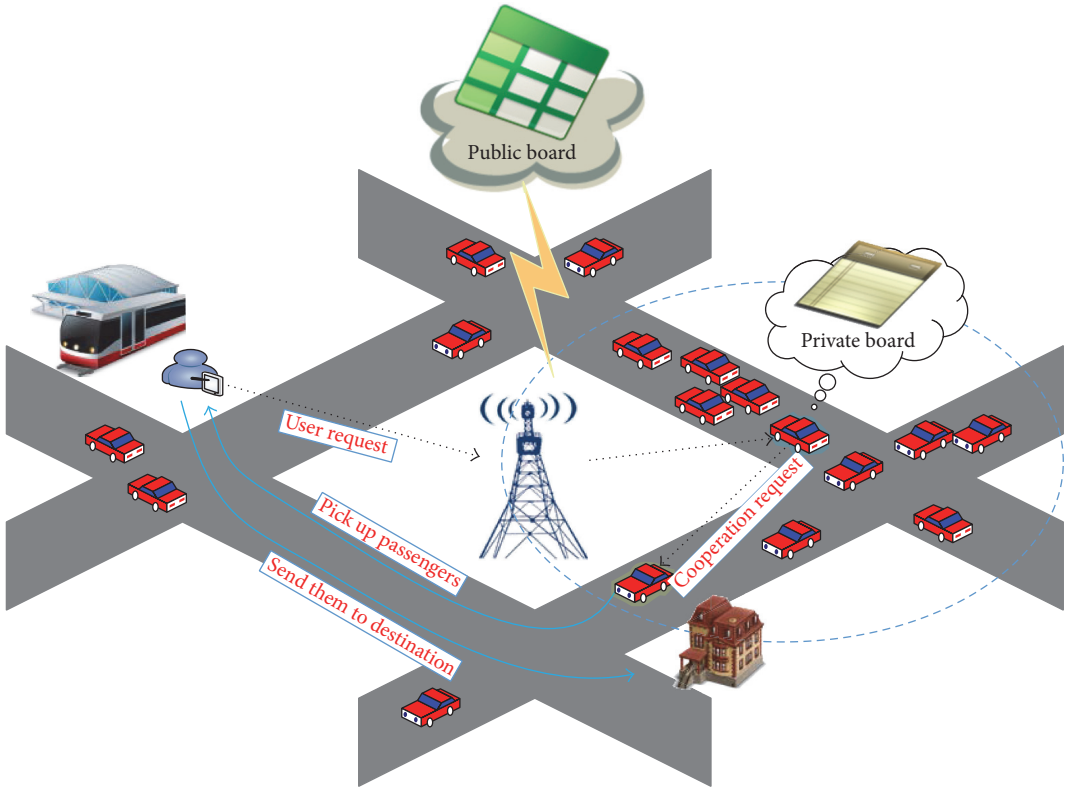

Figure 2: The process of cooperation in DBTEC.

use trust value estimation model to update the information stored in Private board using the information stored in Public board, choose several cooperative vehicles using the synthesized score which is computed from the estimated trust information and service quality information stored in Private board, and send cooperation requests to them. After the cooperative vehicles provide services to vehicle V_i, vehicle V_i will rate the service quality of cooperative vehicles, update the estimated service quality information and trust information in Private board, and selectively report the estimated service quality to Public board.

For a malicious vehicle V_j, it will use various methods to strive for the opportunities of providing services. Once malicious vehicles get these opportunities, they will screw the service requests up in various manners to disrupt the network and benefit themselves. Common malicious behaviors include reporting false self-estimated service quality, slandering normal vehicles, praising malicious partners, and providing unstable services (see Section 3.2).

Figure 2 illustrates a concrete example of the process of cooperation in DBTEC. A passenger in train station wants to hire a taxi; he sends user request by mobile phone to vehicle A with blue shadow. Unfortunately, when vehicle A is driving to train station, it encounters a traffic jam in a street. Obviously, it cannot finish the task by itself in time. It seeks neighboring vehicles which are willing to offer help and combines trust information stored in Private board with information stored in Public board to predict the reliability of these neighboring vehicles. Then it sends cooperation request to a trustworthy vehicle, namely, the vehicle with green shadow. The trustworthy vehicle drives to train station to pick up the passenger and send him to destination.

In the following subsections, we describe the Public board model, Private board model, behavior of normal vehicles, trust value estimation model, and cooperation path generating model, respectively.

4.2. Public Board Model. All vehicles can access Public board, which is stored in cloud storage, anywhere. Public board stores the public estimated service quality for vehicle V_j reported by vehicle V_i, called $E_j'^i$, and the timestamp at which it was reported, called $T_j'^i$. Public board can be expressed by two matrices, E' and T':

$$E' = \begin{bmatrix} E_1'^1 & E_2'^1 & \cdots & E_n'^1 \\ E_1'^2 & E_2'^2 & \cdots & E_n'^2 \\ \cdots & \cdots & \cdots & \cdots \\ E_1'^n & E_2'^n & \cdots & E_n'^n \end{bmatrix},$$
$$T' = \begin{bmatrix} T_1'^1 & T_2'^1 & \cdots & T_n'^1 \\ T_1'^2 & T_2'^2 & \cdots & T_n'^2 \\ \cdots & \cdots & \cdots & \cdots \\ T_1'^n & T_2'^n & \cdots & T_n'^n \end{bmatrix}. \quad (6)$$

Initialize:
Initialize all scalars in E' with 0.5;
Initialize all scalars in T' with 0.
(1) **While** true
(2) **If** V_i reports its estimated service quality value ϱ for V_j
(3) $E_j'^i \coloneqq \varrho$
(4) Set $T_j'^i$ to current timestamp
(5) **End If**
(6) **If** V_i inquires public estimated service quality value of $E_j'^i$
(7) Send $(E_j'^i, T_j'^i)$ to V_i
(8) **End If**
(9) **End While**

Algorithm 1: The algorithm running in Public board.

The scale of $E_j'^i$ is between 0 and 1. $E_j'^i = 0.5$ is a neutral service quality estimation. When $E_j'^i < 0.5$, the service quality is worse than normal level. Conversely, when $E_j'^i > 0.5$, the service quality is better than normal level. $T_j'^i$ is a value larger than or equal to 0. $T_j'^i = 0$ means no updating of estimated service quality has been committed and $E_j'^i$ is still the initial value.

Vehicle can register to join VCC. For vehicle V_i, it sends a registering request to Public board and report its own service quality Q_i, called self-estimated service quality, to it (service quality can be fake if the vehicle is malicious) when registering; Public board will store this service quality in $E_i'^i$ and update the corresponding $T_i'^i$ to 0. After reporting its own service quality to $E_i'^i$, Public board will pack all vehicles' self-estimated service quality together and send it to vehicle V_i; vehicle V_i will use this information as initial service quality to update Private board.

After initialization, Public board will handle the following two events:

(1) If receiving estimated service quality for vehicle V_j reported by vehicle V_i, Public board will update the service quality stored in $E_j'^i$ and timestamp $T_j'^i$.

(2) If a certain vehicle inquires about the estimated service quality for vehicle V_j reported by vehicle V_i, Public board will pack the service quality $E_j'^i$ and the corresponding timestamp $T_j'^i$ as a tuple $(E_j'^i, T_j'^i)$ and send it to the vehicle.

The pseudocode of Public board is presented in Algorithm 1.

4.3. Private Board Model. All vehicles store Private board in their own storage to guide the selection of trustworthy cooperative vehicles.

For vehicle V_i, assume that its estimated service quality for vehicle V_j is E_j^i and its estimated trust value for vehicle V_j is R_j^i. Both of them will be stored in its Private board. In other words, vehicle V_i's Private board records

$$E^i = \left\{E_1^i, E_2^i, \ldots, E_n^i\right\},$$
$$R^i = \left\{R_1^i, R_2^i, \ldots, R_n^i\right\}. \tag{7}$$

The meaning of E_j^i is similar to $E_j'^i$ as described in Section 4.2. The scale of trust value R_j^i is between 0 and 1. $R_j^i = 0.5$ is a neutral trust value estimation. When $R_j^i < 0.5$, vehicle V_i thinks vehicle V_j is malicious. Conversely, when $R_j^i > 0.5$, vehicle V_i thinks vehicle V_j is normal.

Besides E^i and R^i, several timestamps are stored in vehicle V_i's Private board: the timestamp T_j^i at which vehicle V_i updates E_j^i and the timestamp at which vehicle V_i updates R_k^i because of trusting in vehicle V_t (this timestamp can be divided into two subtimestamps: the subtimestamp $T_{i,t\to k}^{\text{col}}$ at which vehicle V_i updates R_k^i from column perspective because of trusting in vehicle V_t and the subtimestamp $T_{i,t\to k}^{\text{row}}$ at which vehicle V_i updates R_k^i from row perspective because of trusting in vehicle V_t). In other words, besides E^i and R^i, vehicle V_i records

$$T^i = \left\{T_1^i, T_2^i, \ldots, T_n^i\right\},$$
$$T_i^{\text{col}} = \begin{bmatrix} T_{i,1\to1}^{\text{col}} & T_{i,1\to2}^{\text{col}} & \cdots & T_{i,1\to n}^{\text{col}} \\ T_{i,2\to1}^{\text{col}} & T_{i,2\to2}^{\text{col}} & \cdots & T_{i,2\to n}^{\text{col}} \\ \cdots & \cdots & \cdots & \cdots \\ T_{i,n\to1}^{\text{col}} & T_{i,n\to2}^{\text{col}} & \cdots & T_{i,n\to n}^{\text{col}} \end{bmatrix},$$
$$T_i^{\text{row}} = \begin{bmatrix} T_{i,1\to1}^{\text{row}} & T_{i,1\to2}^{\text{row}} & \cdots & T_{i,1\to n}^{\text{row}} \\ T_{i,2\to1}^{\text{row}} & T_{i,2\to2}^{\text{row}} & \cdots & T_{i,2\to n}^{\text{row}} \\ \cdots & \cdots & \cdots & \cdots \\ T_{i,n\to1}^{\text{row}} & T_{i,n\to2}^{\text{row}} & \cdots & T_{i,n\to n}^{\text{row}} \end{bmatrix}. \tag{8}$$

These timestamps are stored to prevent updating Private board using the same information repeatedly.

4.4. Normal Vehicles. Every normal vehicle stores its own Private board and can access Public board anywhere.

When entering VCC, all normal vehicles will send a registering request to Public board, report their own self-estimated service quality to Public board, and then wait for the package sent by Public board which stores all vehicles' self-estimated service quality to initialize its Private board.

After registering, for normal vehicle V_i, it will receive a user request at some times. If vehicle V_i can finish the user's task, it will provide services to users directly, which forms a trivial cooperation path whose length is 1 hop (see Figure 1(a)). If vehicle V_i cannot finish the task by itself for some reason, it will send messages to other vehicles to ask if they can cooperate to perform this task. Vehicles which give positive responses form set Υ. Vehicle V_i then uses trust value estimation models to update its own Private board using the information from Public board and uses the updated Private board to compute synthesized score. Vehicle V_i will then send cooperation request to vehicles in Υ with large synthesized score. Vehicles receiving cooperation request will provide service to vehicle V_i. In the process of providing service, they may send cooperation request recursively to more vehicles. The recursive process will form a nontrivial cooperation path (See Figure 1(b)).

Note every cooperation path corresponds to a solution to a service request. The service quality of a cooperation path is defined as the minimum service quality provided by vehicles in cooperation path. Every cooperation path consists of many subcooperation paths whose starting node is an intermediate node in original path and ending node is the ending node of the original task.

After vehicles which received cooperation request finish the cooperation request from vehicle V_i, vehicle V_i will estimate the quality of this service, update its own Private board, and report the updated item to Public board selectively (they may not report it for self-protection or privacy-protection).

Concretely speaking, vehicle V_i sends cooperation request to vehicle V_j, vehicle V_j may finish this task by itself or by further cooperation with other vehicles, and the service quality provided by vehicle V_j for this cooperation request is $\overline{Q}$, which is the minimum service quality in subcooperation path starting from vehicle V_j (i.e., the original cooperation paths' starting node is vehicle V_i). Vehicle V_i will update R_j^i to $\overline{R}$ according to Formula (11), set E_j^i to $\overline{Q}$, and update T_j^i to current timestamp simultaneously.

$$\text{diff} = \left| E_j^i - \overline{Q} \right|, \tag{9}$$

$$f(x) = \begin{cases} 0, & x \le 0, \\ x, & 0 < x < 1, \\ 1, & x \ge 1, \end{cases} \tag{10}$$

$$\overline{R} = \begin{cases} f\left(R_j^i + \sigma_1 * \text{diff}\right), & \text{diff} \le \mu, \\ f\left(R_j^i - \sigma_2 * \text{diff}\right), & \text{diff} > \mu. \end{cases} \tag{11}$$

In Formula (11), μ is the threshold to check if the service quality E_j^i and $\overline{Q}$ are close enough; σ_1 and σ_2 are parameters used to control the extent of change in R_j^i. diff $> \mu$; namely, the difference between E_j^i and $\overline{Q}$ is large, which means the difference between estimated service quality of this cooperation and service quality of last cooperation (or the initial service quality of vehicle V_j) is large and therefore the service quality of vehicle V_j, namely, Q_j, is not stable (or vehicle V_j's initial service quality is false); we should decrease R_j^i according to parameter σ_2. Conversely, diff $\le \mu$ means the service quality of vehicle V_j, namely, Q_j, is stable (or vehicle V_j's initial service quality is true); we should increase R_j^i according to parameter σ_1.

The pseudocode of normal vehicles is presented in Algorithm 2.

4.5. Trust Value Estimation Model. Trust value estimation model can update the information of Private board based on Public board to increase the precision of trust value estimation and guide the selection of cooperative vehicles. In particular, this model will take great effects when trust information is deficient, such as cold start stage.

Trust value estimation model is based on this observation: when vehicle V_i trusts vehicle V_t, the following statements are true:

(1) The estimated service quality for vehicle V_t stored in vehicle V_i's Private board, namely, E_t^i, is true.

(2) Public board's all estimated service quality reported by vehicle V_t is true.

Vehicle V_i uses this observation to update other vehicles' trust value and estimated service quality in Private board and prevent malicious vehicles from taking part in cooperation. The following two rules describe the method.

Rule 1. For a certain vehicle V_t with high trust value R_t^i in vehicle V_i's Private board, if the difference between $E_t'^k$ and E_t^i is large, where $k \in \{k \mid 1 \le k \le n \text{ and } k \ne i \text{ and } k \ne t\}$, decrease R_k^i.

According to observation 1, the estimated service quality for vehicle V_t stored in vehicle V_i's Private board, namely, E_t^i, is true. If the difference between $E_t'^k$ and E_t^i is large, it is likely that vehicle V_k reports a false service quality to Public board, which is a malicious behavior; vehicle V_i should decrease its trust value.

Rule 2. For a certain vehicle V_t with high trust value R_t^i in vehicle V_i's Private board, if $T_r'^t \ne 0$, in other words, $E_r'^t$ has been updated, where $r \in \{r \mid 1 \le r \le n \text{ and } r \ne i \text{ and } r \ne t\}$, vehicle V_i updates the Private board according to two cases.

Case 1. In the case where $T_r^i \ne 0$, in other words, E_r^i has been updated, if the difference between $E_r'^t$ and E_r^i is large, decrease R_r^i. If the difference between $E_r'^t$ and E_r^i is small, then check R_r^i: if R_r^i is small, then further decrease R_r^i; if R_r^i is large, then further increase R_r^i.

Initialize:
Initialize all scalars in E^i and R^i with 0.5
Initialize all scalars in T^i, T_i^{col} and T_i^{row} with 0
Register itself by reporting Q_i to Public board
Waiting for self-estimated service quality of other vehicles sent by Public board
(1) **While** true
(2) Move randomly in the area
(3) **If** V_i receives a service request
(4) **If** V_i can finish the request
(5) Provide service to requestor directly
(6) **Else**
(7) Search vehicles willing to offer help
(8) Update Private board using trust value estimation model
(9) Compute synthesized score of vehicles in Υ
(10) Assume c vehicles are needed to complete the task
(11) Set $\gamma' = \{V_{k_1}, V_{k_2}, \ldots, V_{k_c}\}$ as vehicles in Υ with first c largest synthesized score
(12) Send cooperation request to vehicles in γ'
(13) **For** vehicle V_k **in** γ'
(14) Receive service from vehicle V_k with quality ϱ
(15) $E_k^i \coloneqq \varrho$
(16) Set T_k^i to current time stamp
(17) Report service quality ϱ to Public board
(18) diff $\coloneqq |E_k^i - \overline{Q}|$
(19) **If** diff $\leq \mu$
(20) $R_k^i \coloneqq f(R_k^i + \sigma_1 * \text{diff})$
(21) **Else**
(22) $R_k^i \coloneqq f(R_k^i - \sigma_2 * \text{diff})$
(23) **End If**
(24) **End For**
(25) **End If**
(26) **End If**
(27) **End While**

Algorithm 2: The algorithm running in normal car V_i.

When the difference between $E_r'^t$ and E_r^i is large, according to observation 2, $E_r'^t$ is close to real value, and therefore E_r^i may deviate from the real value, which means vehicle V_r provides different service quality to different vehicles maliciously. Vehicle V_i should decrease R_r^i.

When the difference between $E_r'^t$ and E_r^i is small, there are two cases to analyze: both vehicle V_i and vehicle V_t believe vehicle V_r is malicious or both vehicle V_i and vehicle V_t believe vehicle V_r is normal. We can use the estimated trust value for vehicle V_r stored in vehicle V_i's Private board, namely, R_r^i, to distinguish the two cases. When R_r^i is small, vehicle V_i can infer that both vehicle V_i and vehicle V_t think vehicle V_r is malicious; vehicle V_i further decreases R_r^i. Conversely, when R_r^i is large, vehicle V_i can infer that both vehicle V_i and vehicle V_t think object V_r is normal; vehicle V_i further increases R_r^i.

Case 2. In the case where $T_r^i = 0$, in other words, E_r^i has not been updated, if R_t^i is large enough, make vehicle V_i accept a virtual cooperation from vehicle V_r whose service quality is $E_r'^t$: set E_r^i as $E_r'^t$ and update the trust value of vehicle V_r according to this virtual cooperation using Formula (11).

According to observation 2, vehicle V_i can update the estimated service quality for vehicle V_r in Private board using $E_r'^t$, which is a trustworthy value when vehicle V_t is trustworthy.

We will detail this model in the next two subsections. Section 4.5.1 will give a concrete scheme; DBTEC scheme with this section's trust value estimation model is called DBTEC-1. Section 4.5.2 will improve the scheme to solve cold start problem; DBTEC scheme with this section's improved trust value estimation model is called DBTEC-2.

4.5.1. DBTEC-1. We detail the trust value estimation model and propose a temporary detailed scheme called DBTEC-1 in this section, which will be further improved in the next subsection.

Below we detail the two rules listed above.

Rule 1. For vehicle V_t with trust value $R_t^i \geq \gamma$ in vehicle V_i's Private board, if $T_t'^k > T_{i,t\to k}^{\text{col}} \geq 0$ and $\text{diff}' = |E_t^i - E_t'^k| > \mu$,

where $k \in \{k \mid 1 \le k \le n \text{ and } k \ne i \text{ and } k \ne t\}$, vehicle V_i decreases R_k^i to $\overline{R}'$ according to Formula (12) and updates $T_{i,t\to k}^{\text{col}}$ to current timestamp.

$$\overline{R}' = f\left(R_k^i - \omega_2 * \text{diff}'\right). \tag{12}$$

Condition $T_t'^k > T_{i,t\to k}^{\text{col}} \ge 0$ guarantees that $E_t'^k$ has been updated since initialization and vehicle V_i will not update trust value using the same information repeatedly. $T_{i,t\to k}^{\text{col}}$ is the subtimestamp at which vehicle V_i updates R_k^i from column perspective because of trusting in vehicle V_t. In other words, this subtimestamp records the timestamp. Rule 1 is used to update trust value's last time. $T_t'^k \le T_{i,t\to k}^{\text{col}}$ means no updating has been committed to $E_t'^k$ since using Rule 1 last time. Repeat updating using the same information will lead to error. Formula (12) means vehicle V_i will decrease R_k^i according to parameter ω_2 if $\text{diff}' > \mu$.

Rule 2. For a certain vehicle V_t with trust value $R_t^i \ge \gamma$ in vehicle V_i's Private board, if $T_r'^t > T_{i,t\to r}^{\text{row}} \ge 0$, where $r \in \{r \mid 1 \le r \le n \text{ and } r \ne i \text{ and } r \ne t\}$, vehicle V_i updates the Private board according to two cases.

Case 1. In the case where $T_r^i \ne 0$, in other words, E_r^i has been updated, suppose that the difference between $E_r'^t$ and E_r^i is $\text{diff}'' = |E_r^i - E_r'^t|$. If $\text{diff}'' > \mu$, vehicle V_i decreases R_r^i to $\overline{R}''$ according to Formula (13) and updates $T_{i,t\to r}^{\text{row}}$ to current timestamp:

$$\overline{R}'' = f\left(R_r^i - \omega_2 * \text{diff}''\right); \tag{13}$$

if $\text{diff}'' \le \mu$, vehicle V_i updates R_r^i to $\overline{R}'''$ using Formula (15) and updates $T_{i,t\to r}^{\text{row}}$ to current timestamp:

$$\text{dev} = R_r^i - 0.5, \tag{14}$$

$$\overline{R}''' = \begin{cases} f\left(R_r^i + \omega_1 |\text{dev}|\right), & \text{dev} > 0, \\ f\left(R_r^i - \omega_2 |\text{dev}|\right), & \text{dev} \le 0. \end{cases} \tag{15}$$

$\overline{R}''$ means vehicle V_i will decrease R_r^i according to parameter ω_2 if the difference between $E_r'^t$ and E_r^i is large ($\text{diff}'' > \mu$). $\overline{R}'''$ means that if vehicle V_r is likely to be a normal vehicle, we further increase R_r^i according to parameter ω_1. The more trustworthy vehicle V_r is, the larger amount of increment R_r^i has. Conversely, if vehicle V_r is likely to be a malicious vehicle, we further decrease R_r^i according to parameter ω_2. The less trustworthy vehicle V_r is, the larger amount of decrement R_r^i has.

Case 2. In the case where $T_r^i = 0$, in other words, E_r^i has not been updated, make vehicle V_i accept a virtual cooperation from vehicle V_r whose service quality is $E_r'^t$: set E_r^i as $E_r'^t$, update the trust value of vehicle V_r according to this virtual cooperation using Formula (11), and update $T_{i,t\to r}^{\text{row}}$ to current timestamp.

```
(1) For V_t in Φ/{V_i}
(2)   If E_t^i ≥ γ
(3)     For V_k in {V_k | 1 ≤ k ≤ n and k ≠ i and k ≠ t}
(4)       If T_t'^k > T_{i,t→k}^col ≥ 0
(5)         diff' := |E_t^i − E_t'^k|
(6)         If diff' > μ
(7)           R_k^i := f(R_k^i − ω_2 * diff')
(8)           Set T_{i,t→k}^col to current time stamp
(9)         End If
(10)      End If
(11)    End For
(12)    For V_r in {V_r | 1 ≤ r ≤ n and r ≠ i and r ≠ t}
(13)      If T_r'^t > T_{i,t→r}^row ≥ 0
(14)        If T_r^i ≠ 0
(15)          diff'' := |E_r^i − E_r'^t|
(16)          If diff'' > μ
(17)            R_r^i := f(R_r^i − ω_2 * diff'')
(18)            Set T_{i,t→r}^row to current time stamp
(19)          Else
(20)            dev := R_r^i − 0.5
(21)            If dev > 0
(22)              R_r^i := f(R_r^i + ω_1 * |dev|)
(23)            Else
(24)              R_r^i := f(R_r^i − ω_2 * |dev|)
(25)            End If
(26)            Set T_{i,t→r}^row to current time stamp
(27)          End If
(28)        Else
(29)          diff := |E_r^i − E_r'^t|
(30)          If diff ≤ μ
(31)            R_r^i := f(R_r^i + σ_1 * diff)
(32)          Else
(33)            R_r^i := f(R_r^i − σ_2 * diff)
(34)          End If
(35)          E_r^i := E_r'^t
(36)          Set T_{i,t→r}^row to current time stamp
(37)        End If
(38)      End If
(39)    End For
(40)  End If
(41) End For
```

Algorithm 3: DBTEC-1's trust value estimation model in V_i.

The pseudocode of this scheme is presented in Algorithm 3.

4.5.2. DBTEC-2. We further improve the performance of DBTEC-1 in this subsection.

In the beginning stage of VCC, deficiency of information will make many trust value estimation schemes invalid. This phenomenon is called cold start.

DBTEC-1 scheme cannot guide the selection of cooperative partner well when in stage of cold start because it can only take effect when vehicles' trust value becomes larger than threshold γ. We can improve the original scheme based on this observation.

Unlike DBTEC-1, we no longer use an absolute threshold as a starting condition of trust value estimation scheme. We can adjust the influence of trust value estimation scheme according to the trust value of vehicles. The more trustworthy the vehicle is, the more influence it will have in trust value estimation scheme.

This improvement can significantly enhance the performance of DBTEC-1, especially in the stage of cold start, since it can address the trust information deficiency problem in cold start stage, in which most of the failed cooperation happened. DBTEC scheme with this improved trust value estimation model is called DBTEC-2, which is an improved version of DBTEC-1.

Below we introduce the improvement in detail.

According to the description above, two new parameters, α_1 and α_2, are introduced into the scheme to control the influence of trust value estimation model. α_1 and α_2 can be computed according to the following two formulas:

$$\alpha_1 = \begin{cases} \omega_1, & E_t^i \geq \gamma, \\ \dfrac{\left(R_t^i - 0.5\right)\omega_1}{\gamma - 0.5}, & E_t^i < \gamma, \end{cases} \tag{16}$$

$$\alpha_2 = \begin{cases} \omega_2, & E_t^i \geq \gamma, \\ \dfrac{\left(R_t^i - 0.5\right)\omega_2}{\gamma - 0.5}, & E_t^i < \gamma. \end{cases} \tag{17}$$

When $R_t^i \geq \gamma$, vehicle V_t is completely trustworthy and therefore the influence degree α_1 is corresponding to ω_1, which is the biggest influence degree. When $R_t^i < \gamma$, Formula (16) maps trust value $0.5 \sim R_t^i$ to $0 \sim \omega_1$. The larger R_t^i is, the larger influence degree α_1 is. α_2 has similar conclusion. When $R_t^i \geq \gamma$, vehicle V_t is completely trustworthy and therefore the influence degree α_2 is corresponding to ω_2, which is the biggest influence degree. When $R_t^i < \gamma$, Formula (17) maps trust value $0.5 \sim R_t^i$ to $0 \sim \omega_2$.

Below are the rules of the improved scheme.

Rule 1. For a certain vehicle V_t with trust value $R_t^i \geq 0.5$ in vehicle V_i's Private board, compute α_1 and α_2. If $T_t'^k > T_{i,t\to k}^{\text{col}} \geq 0$ and $\text{diff}' = |E_t^i - E_t'^k| > \mu$, where $k \in \{k \mid 1 \leq k \leq n \text{ and } k \neq i \text{ and } k \neq t\}$, vehicle V_i decreases R_k^i to $\overline{R}'$ according to Formula (18) and updates $T_{i,t\to k}^{\text{col}}$ to current timestamp.

$$\overline{R}' = f\left(R_k^i - \alpha_2 * \text{diff}'\right). \tag{18}$$

Rule 2. For a certain vehicle V_t with trust value $R_t^i \geq \gamma$ in vehicle V_i's Private board, compute α_1 and α_2. If $T_r'^t > T_{i,t\to r}^{\text{row}} \geq 0$, where $r \in \{r \mid 1 \leq r \leq n \text{ and } r \neq i \text{ and } r \neq t\}$, vehicle V_i updates the Private board according to two cases.

Case 1. In the case where $T_r^i \neq 0$, in other words, E_r^i has been updated, suppose that the difference between $E_r'^t$ and E_r^i is $\text{diff}'' = |E_r^i - E_r'^t|$. If $\text{diff}'' > \mu$, vehicle V_i decreases R_r^i to $\overline{R}''$ according to Formula (19) and updates $T_{i,t\to r}^{\text{row}}$ to current timestamp:

$$\overline{R}'' = f\left(R_r^i - \alpha_2 * \text{diff}''\right); \tag{19}$$

if $\text{diff}'' \leq \mu$, vehicle V_i updates R_r^i to $\overline{R}'''$ using Formula (21) and updates $T_{i,t\to r}^{\text{row}}$ to current timestamp:

$$\text{dev} = R_r^i - 0.5, \tag{20}$$

$$\overline{R}''' = \begin{cases} f\left(R_r^i + \alpha_1 |\text{dev}|\right), & \text{dev} > 0, \\ f\left(R_r^i - \alpha_2 |\text{dev}|\right), & \text{dev} \leq 0. \end{cases} \tag{21}$$

Case 2. In the case where $T_r^i = 0$, in other words, E_r^i has not been updated, make vehicle V_i accept a virtual cooperation from vehicle V_r whose service quality is $E_r'^t$: set E_r^i as $E_r'^t$, update the trust value of vehicle V_r according to this virtual cooperation using Formula (11), and update $T_{i,t\to r}^{\text{row}}$ to current timestamp.

The pseudocode of this scheme is presented in Algorithm 4.

4.6. Cooperation Path Generating Model. When vehicle V_i receives a user request, such as delivering a package to person P in a certain place as soon as possible, if vehicle V_i can finish this task by itself individually, it will provide services to user directly, which forms a trivial cooperation path whose length is 1 hop, that is, a hop from user to vehicle V_i. If vehicle V_i cannot finish this task by itself, for example, vehicle V_i cannot deliver the package to person P directly, vehicle V_i will finish what it can do and then send messages to other vehicles to ask if they are willing to cooperate to perform the rest of the task. Vehicles give positive response form set Υ. Vehicle V_i will select several vehicles in set Υ according to computed synthesized scores and send cooperation request to them. Assume that vehicle V_j has received cooperation request from vehicle V_i. If vehicle V_j can finish the task, it will provide services to vehicle V_i. If vehicle V_j still cannot perform the task by itself, it will do what it can do and further recursively send the cooperation request to other vehicles just like vehicle V_i. This recursive process forms a nontrivial cooperation path (see Figure 1). The cooperation path starting from vehicle V_j can be viewed as a subcooperation path of the cooperation path starting from V_i as described in Section 4.4. Note that a cooperation path is corresponding to a solution to a user request.

The key problem in constructing cooperation path is how to guarantee the quality and the safety of service provided by the cooperation path. Users always want to receive service with high quality under the condition that the safety of this service can be guaranteed. For example, users who want to deliver a package to person P expect the package to be sent to person P as soon as possible without any damage.

To overcome the key problem, three factors have to be considered: the trust value, the service quality, and the near completion degree of cooperative vehicles.

```
(1) For V_t in Φ/{V_i}
(2)   If E_t^i ≥ 0.5
(3)     Compute α_1 and α_2
(4)     For V_k in {V_k | 1 ≤ k ≤ n and k ≠ i and k ≠ t}
(5)       If T_t^{'k} > T_{i,t→k}^{col} ≥ 0
(6)         diff' := |E_t^i − E_t^{'k}|
(7)         If diff' > μ
(8)           R_k^i := f(R_k^i − α_2 * diff')
(9)           Set T_{i,t→k}^{col} to current time stamp
(10)        End If
(11)      End If
(12)    End For
(13)    For V_r in {V_r | 1 ≤ r ≤ n and r ≠ i and r ≠ t}
(14)      If T_r^{'t} > T_{i,t→r}^{row} ≥ 0
(15)        If T_r^i ≠ 0
(16)          diff'' := |E_r^i − E_r^{'t}|
(17)          If diff'' > μ
(18)            R_r^i := f(R_r^i − α_2 * diff'')
(19)            Set T_{i,t→r}^{row} to current time stamp
(20)          Else
(21)            dev := R_r^i − 0.5
(22)            If dev > 0
(23)              R_r^i := f(R_r^i + α_1 * |dev|)
(24)            Else
(25)              R_r^i := f(R_r^i − α_2 * |dev|)
(26)            End If
(27)            Set T_{i,t→r}^{row} to current time stamp
(28)          End If
(29)        Else
(30)          If E_t^i ≥ γ
(31)            diff := |E_r^i − E_r^{'t}|
(32)            If diff ≤ μ
(33)              R_r^i := f(R_r^i + σ_1 * diff)
(34)            Else
(35)              R_r^i := f(R_r^i − σ_2 * diff)
(36)            End If
(37)            E_r^i := E_r^{'t}
(38)            Set T_{i,t→r}^{row} to current time stamp
(39)          End If
(40)        End If
(41)      End If
(42)    End For
(43)  End If
(44) End For
```

Algorithm 4: DBTEC-2's trust value estimation model in V_i.

The estimated trust value and the service quality are known to vehicles. The near completion degree of a vehicle means to what extent can this vehicle perform the task. When vehicle V_i sends messages to vehicle V_j and asks if it is willing to cooperate, vehicle V_j will report the near completion degree of itself on this task to vehicle V_i if it is willing to cooperate. The near completion degree is different from the service quality, which means the quality of service provided by vehicles in the process of providing service. For example, when the task is delivering a package, the near completion degree of vehicle V_j is how far vehicle V_j can deliver this package (it may pass the package on to another cooperative partner after delivering the packages to the farthest place it can reach). Obviously, in order to reduce the length of cooperation path as much as possible, vehicles should greedily choose the vehicle which can do more parts of the task as their cooperative partner. Because the greedy strategy will indirectly minimize the failure rate of cooperation, there is a certain probability of choosing malicious vehicles when sending cooperation request. Therefore the longer the cooperation path is, the more choices to be made are and the larger the probability that malicious vehicles get opportunities to provide services is.

Assume that the estimated service quality for vehicle V_j stored in vehicle V_i's Private board is E_j^i, the estimated trust value for vehicle V_j stored in vehicle V_i's Private board is R_j^i, and vehicle V_j's normalized near completion degree on task ξ is C_j^ξ. We can synthesize the three factors and use them to compute the synthesized score which can guide the selection of cooperative vehicles. Vehicle V_i's synthesized score to vehicle V_j is

$$P_j^i = \lambda_1 E_j^i + \lambda_2 R_j^i + \lambda_3 C_j^\xi, \tag{22}$$

where $\lambda_1 > 0$, $\lambda_2 > 0$, $\lambda_3 > 0$, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

When selecting cooperative vehicles, vehicle A will compute synthesized score of vehicles in set Υ and choose vehicles with large synthesized score to cooperate with.

The algorithm of generating cooperation path is presented in Algorithm 5.

5. Performance Analysis and Experimental Results

5.1. Overview. In this section, we will prove the effectiveness of DBTEC schemes by theoretical analysis and extensive experiments. In Section 5.2, the time complexity of DBTEC schemes is given to illustrate the theoretical performance of DBTEC schemes. In Section 5.3, the performance of DBTEC schemes is analyzed by experiments and simulations.

All simulation programs are implemented by C++ with Visual Studio 2013. The proportion of malicious vehicles to all vehicles is 40%~70%; the time interval between two consecutive timestamp is defined as 15 minutes. In a timestamp, the probability of receiving user requests for every vehicle is 20% in experiments of the average estimated trust value and the success ratio of each stage and 70% in experiments of the total success ratio of cooperation requests.

Five threat models are analyzed in Section 5.3. They are reporting false self-estimated service quality, pretending to be normal vehicles, slandering normal vehicles, praising malicious partners, and providing unstable services. They all have been described in Section 3.2.

Three major indexes are computed in each threat model. They are the average estimated trust value of malicious and normal vehicles, the total success ratio of cooperation requests, and the success ratio of each stage.

Initialize:
vehicle A receives a service request ξ
(1) **If** V_i can finish task ξ
(2) V_i can finish task ξ
(3) **Return** V_i's service quality
(4) **Else**
(5) Subdivide task ξ into $\xi_0, \xi_1, \xi_2, \ldots, \xi_k$
(6) V_i finishes the subtask ξ_0
(7) Search vehicles willing to perform the rest of tasks which form set γ and receive their C_j^{ξ} $(j = 1, 2, \ldots, \gamma)$
(8) Compute synthesized score of vehicles in γ
(9) Select vehicles with the first k largest synthesized score which forms set γ_k
(10) **For** V_t **in** γ_k
(11) **Call** Algorithm 5 with ξ_k
(12) **End For**
(13) **End If**

Algorithm 5: Cooperation path generating model.

The performances of three schemes are analyzed in each index. They are traditional scheme, DBTEC-1, and DBTEC-2. In traditional scheme, Public board is deprecated and vehicles can only acquire trust information by direct interaction [28]. More concretely speaking, traditional scheme can be regarded as a reduction version of DBTEC scheme without Public board model and trust value estimation model. DBTEC-1 and DBTEC-2 are proposed in Section 4.5. DBTEC-2 is an improved version of DBTEC-1, which can address the trust information problem in cold start stage as illustrated in Section 4.5.2.

5.2. Complexity Analysis. Assume that we use the number of vehicles involved in VCC, that is, h, as the measure to model the input size of the scheme, which is a natural choice in this scenario. It is easy to analyze the time complexity of DBTEC schemes.

In DBTEC-1 scheme, the whole structure of the pseudocode is formed by two-tier nested loops. The maximum running number of outer loops is $O(h)$. In worst cases, the maximum running number of inner loops is also $O(h)$. Therefore, the worst-case time complexity of DBTEC-1 is $O(h^2)$, which can be immediately computed by vehicular chips.

DBTEC-2 scheme is very similar to DBTEC-1 except that a little extra computation is introduced to compute α_1 and α_2, which is $O(1)$. Therefore, the worst-case time complexity of DBTEC-2 is also (h^2), which can be immediately computed by vehicular chips.

5.3. Performance in Various Threat Models

5.3.1. Reporting False Self-Estimated Service Quality. This is a comparatively simple treat model. In this model, malicious vehicles will report a mendacious self-estimated service quality to Public board when registering. When seeking cooperative partners, vehicles tend to send cooperation requests to vehicles with high service quality. Malicious vehicles expect to deceive them using the mendacious self-estimated service quality. After registering, malicious vehicles will move in the limited area and wait for cooperation requests, but they will not accept any user requests in these threat models.

We first analyze the influence of DBTEC-1 and DBTEC-2 on average estimated trust value in this threat model.

Figure 3 illustrates the increment of average estimated trust value of normal vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis of this figure is timestamp and the vertical axis is the estimated trust value of normal vehicles. As illustrated by Figure 3, in traditional scheme, the increment of average estimated trust value of normal vehicles is slow, but when using DBTEC scheme the speed of increment increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of increment more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

Figure 4 illustrates the decrement of average estimated trust value of malicious vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis of this figure is timestamp and the vertical axis is the estimated trust value of malicious vehicles. As illustrated by Figure 4, in traditional scheme, the decrement of average estimated trust value of malicious vehicles is slow, but when using DBTEC schemes the speed of decrement increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of decrement more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

In general, Figures 3 and 4 illustrate that DBTEC has significantly positive influence on estimating trust value of vehicles. DBTEC-2 performs much better than DBTEC-1, especially in the stage of cold start.

Then, we analyze the influence of DBTEC-1 and DBTEC-2 on total success ratio of cooperation requests in this threat

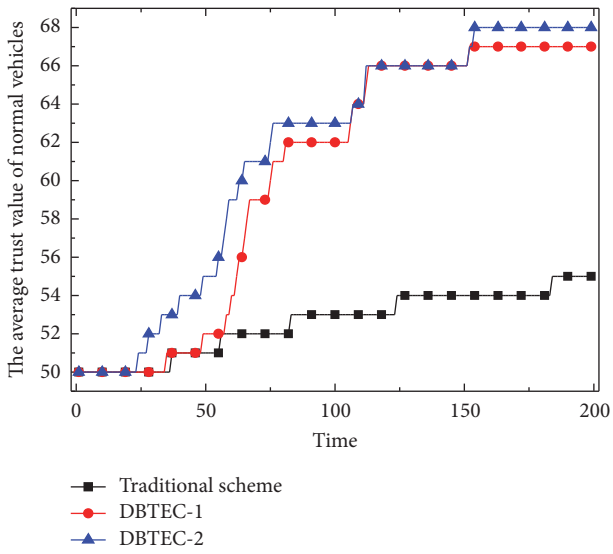

Figure 3: Time-average trust value of normal vehicles.

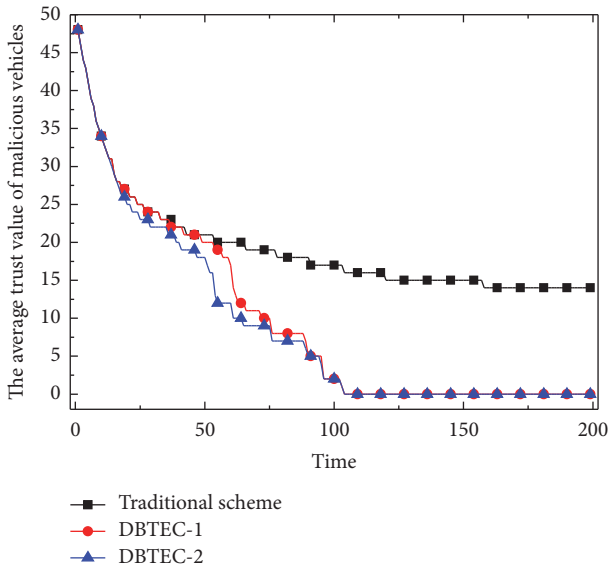

Figure 4: Time-average trust value of malicious vehicles.

model. The success ratio is defined as the proportion of cooperation requests sent to normal vehicles to total cooperation requests in a certain timespan. The total success ratio is defined as the success ratio from initialization to current timestamp. Figure 5 illustrates the increment of total success ratio as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis of this figure is timestamp and the vertical axis is the total success ratio from initialization to current timestamp. As illustrated by Figure 5, the positive influence of DBTEC-1 on total success ratio is comparatively small when compared with the positive influence of DBTEC-2. Particularly in the stage of cold start, the total success ratio of DBTEC-1 is nearly equal to the total success ratio of traditional scheme. DBTEC-2 outperforms

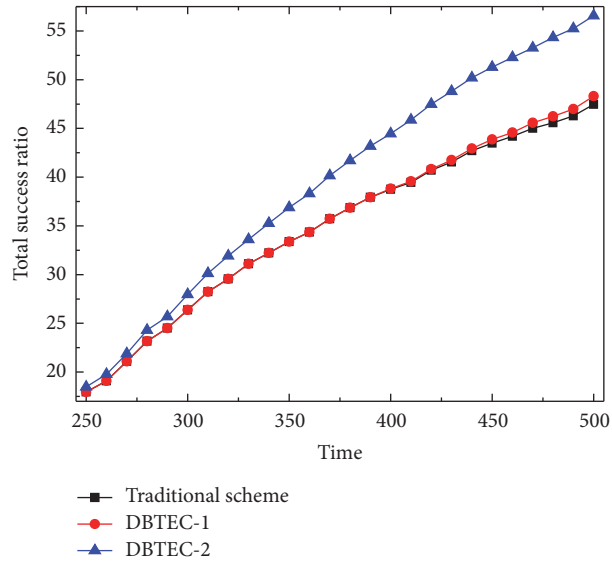

Figure 5: Time-total success ratio.

DBTEC-1 and traditional scheme significantly in the total success ratio.

This phenomenon can be explained by the following observation: total success ratio is dominated by the cooperation requests sent in the stage of cold start. In the stage of cold start, the deficiency of trust information leads to a lot of mistaken selections of cooperative partners, which dominate the change of total success ratio. The more mistakes made in this stage are, the less total success ratio is. As time goes on, vehicles' direct interaction accumulated a lot of trust information which can guide the selection of cooperative partners effectively. The changes of total success ratio in traditional scheme and in DBTEC scheme tend to be similar in this stage. The reason why DBTEC-1's positive influence is small, especially in the stage of cold start, is that DBTEC-1 will take effects when some vehicles' trust information is larger than γ. In other words, DBTEC-1 may not take effects in the stage of cold start and the behavior of DBTEC-1 in that stage is very similar to the behavior of traditional scheme. When this condition is satisfied, traditional scheme has accumulated a lot of trust information to guide its cooperative partners' selection and DBTEC-1 can only take effects in limited cases. However, DBTEC-2 can provide trust information even in the stage of cold start, which sharply reduces mistaken selections. That is why it is far better than DBTEC-1 and traditional scheme.

In general, DBTEC is better than traditional scheme in the total success ratio, DBTEC-2 increases much more total success ratio compared to DBTEC-1 and traditional scheme, and DBTEC-1 increases the total success ratio in a small amount.

Finally, we analyze the influence of DBTEC-1 and DBTEC-2 on the success ratio of each stage in this threat model. We set every 50 timestamps as a stage in the experiment. The success ratio of each stage is the success ratio in timespan of 50 timestamps. Figure 6 illustrates the increment

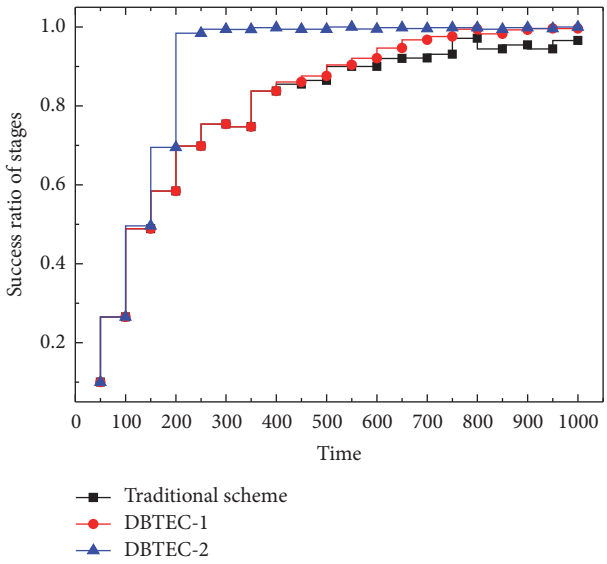

Figure 6: Time-success ratio of stages.

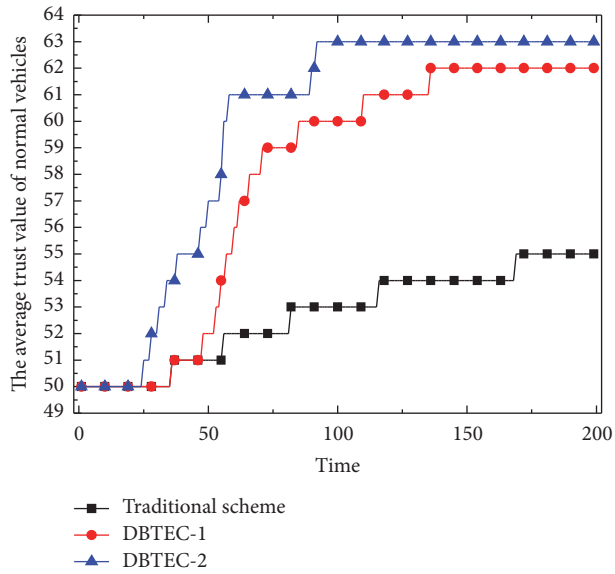

Figure 7: Time-average trust value of normal vehicles.

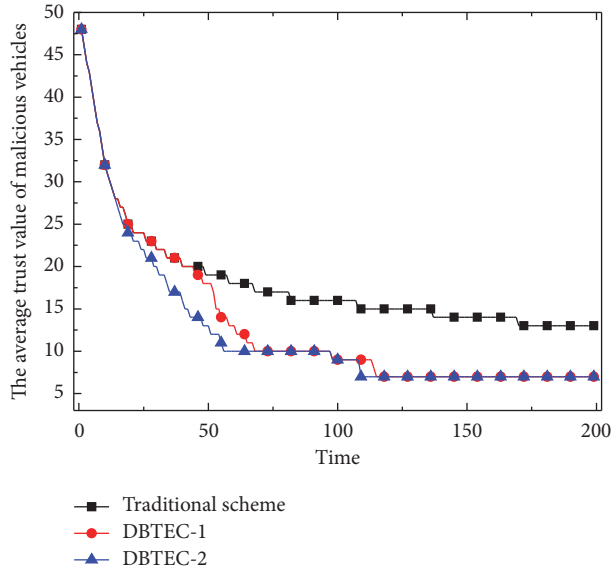

Figure 8: Time-average trust value of malicious vehicles.

of the success ratio of each stage, namely, each 50 timestamps. The horizontal axis of this figure is timestamp and the vertical axis is the success ratio of the stage the timestamp belongs to. As illustrated by Figure 6, the success ratio of stages of both traditional scheme and DBTEC schemes increases as time goes on. Both DBTEC-1 and DBTEC-2's increasing speed outperforms traditional scheme. DBTEC-2's increasing speed outperforms DBTEC-1 in every stage significantly; this phenomenon is significant especially in the stage of cold start. In general, DBTEC' success ratio of each stage is larger than traditional scheme in each stage. DBTEC-2 increases larger success ratio of each stage compared to DBTEC-1, especially in the stage of cold start.

5.3.2. Pretending to Be Normal Vehicles. This threat model is more complicated than the previous one. In this model, malicious vehicles not only will report mendacious self-estimated service quality to Public board when registering but also will pretend to be normal vehicles and perform the same as them. Experimental results proof, in this threat model, that DBTEC will distinguish malicious vehicles and normal ones better than the former threat model; DBTEC-2 has better effects than DBTEC-1.

We first analyze the influence of DBTEC-1 and DBTEC-2 on average estimated trust value in this threat model.

Figure 7 illustrates the increment of average estimated trust value of normal vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 3. As illustrated by Figure 7, similar results will be obtained as in Figure 3. In traditional scheme, the increment of average estimated trust value of normal vehicles is slow, but when using DBTEC schemes the speed of increment increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of increment more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

Figure 8 illustrates the decrement of average estimated trust value of malicious vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 4. As illustrated by Figure 8, similar results will be obtained as in Figure 4; in traditional scheme, the decrement of average estimated trust value of malicious vehicles is slow, but when using DBTEC schemes the speed of decrement increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed

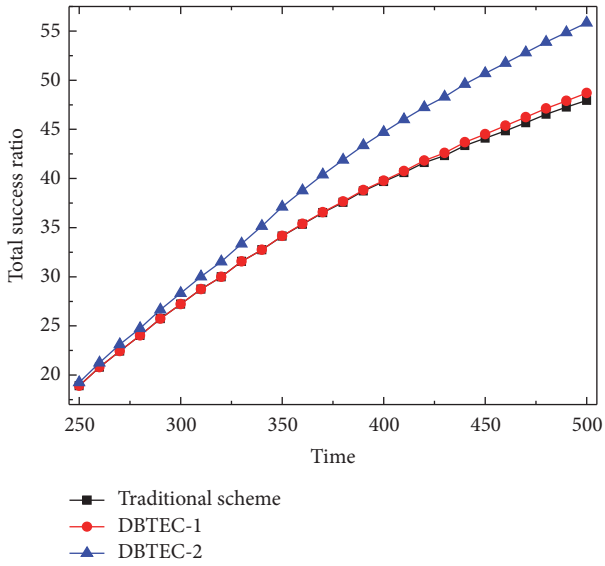

Figure 9: Time-total success ratio.

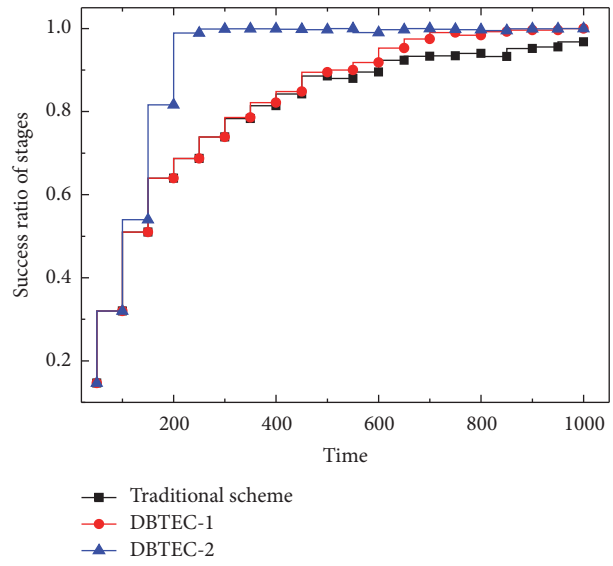

Figure 10: Time-success ratio of stages.

of decrement more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

In general, Figures 7 and 8 illustrate that DBTEC has significantly positive influence on estimating trust value of vehicles. DBTEC-2 performs much better than DBTEC-1, especially in the stage of cold start. DBTEC schemes can take effects even in more complicated situations.

Then, we analyze the influence of DBTEC-1 and DBTEC-2 on total success ratio of cooperation requests in this threat model. Figure 9 illustrates the increment of total success ratio as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis are the same as Figure 5. As illustrated by Figure 9, similar results will be obtained as in Figure 5; the positive influence of DBTEC-1 on total success ratio is comparatively small when compared with the positive influence of DBTEC-2. Particularly in the stage of cold start, the total success ratio of DBTEC-1 is nearly equal to the total success ratio of traditional scheme. DBTEC-2 outperforms DBTEC-1 and traditional scheme significantly in the total success ratio. In general, Figure 9 illustrates that DBTEC is better than traditional scheme in the total success ratio, DBTEC-2 increases much more total success ratio compared to DBTEC-1 and traditional scheme, and DBTEC-1 increases the total success ratio in a small amount.

Finally, we analyze the influence of DBTEC-1 and DBTEC-2 on the success ratio of each stage in this threat model. The experimental method is the same as Figure 6. Figure 10 illustrates the increment of the success ratio of each stage, namely, each 50 timestamps. The horizontal axis and the vertical axis are the same as Figure 6. As illustrated by Figure 10, similar results will be obtained as in Figure 6. The success ratio of stages of both traditional scheme and DBTEC schemes increases as time goes on. Both DBTEC-1 and DBTEC-2's increasing speed outperforms traditional scheme. DBTEC-2's increasing speed outperforms DBTEC-1 in every stage significantly; this phenomenon is significant especially in the stage of cold start. In general, Figure 10 illustrates that DBTEC' success ratio of each stage is larger than traditional scheme in each stage. DBTEC-2 increases larger success ratio of each stage compared to DBTEC-1, especially in the stage of cold start.

5.3.3. Slandering Normal Vehicles. In this model, malicious vehicles not only will report mendacious self-estimated service quality to Public board when registering but also will report low estimated service quality of normal vehicles to slander them even if these normal vehicles never provide services to them. By slandering normal vehicles, malicious vehicles' opportunities of providing services increase indirectly. This can be regarded as a collusion attack as illustrated in Section 3.2. Experimental results proof that DBTEC will also distinguish malicious vehicles and normal ones in this threat model; DBTEC-2 has better effects than DBTEC-1.

We first analyze the influence of DBTEC-1 and DBTEC-2 on average estimated trust value in this threat model.

Figure 11 illustrates the increment of average estimated trust value of normal vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 3. As illustrated by Figure 11, similar results will be obtained as in Figure 3. In traditional scheme, the increment of average estimated trust value of normal vehicles is slow, but when using DBTEC schemes the speed of increment increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of increment more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

Figure 12 illustrates the decrement of average estimated trust value of malicious vehicles as time goes on using

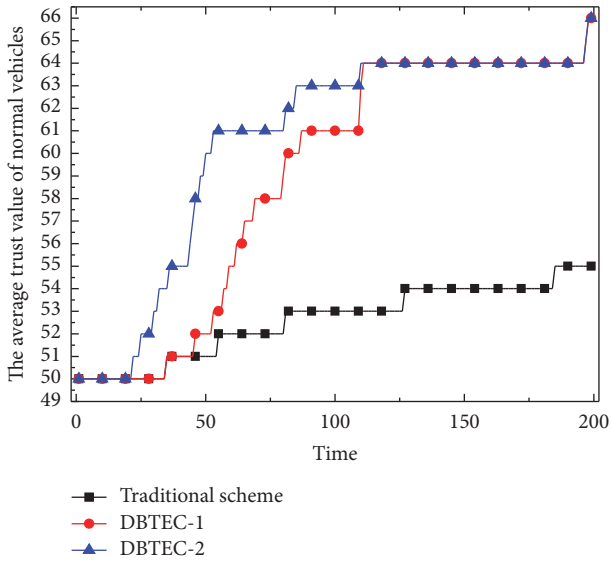

Figure 11: Time-average trust value of normal vehicles.

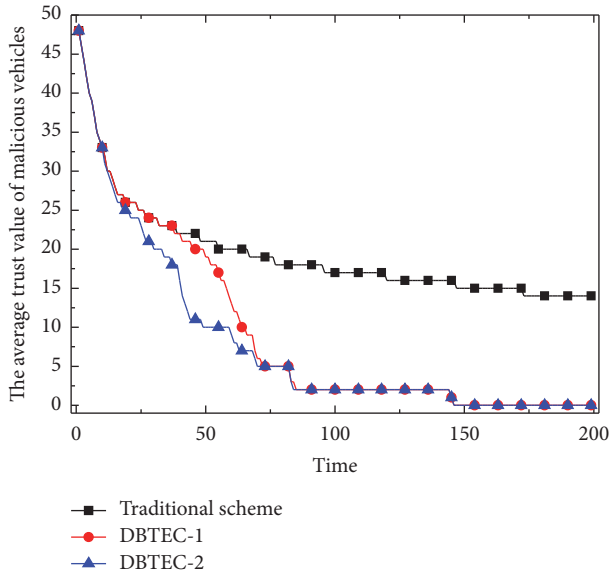

Figure 12: Time-average trust value of malicious vehicles.

traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 4. As illustrated by Figure 12, similar results will be obtained as in Figure 4; in traditional scheme, the decrement of average estimated trust value of malicious vehicles is slow, but when using DBTEC schemes the speed of decrement increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of decrement more significantly than DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

In general, Figures 11 and 12 illustrate that DBTEC has significantly positive influence on estimating trust value of

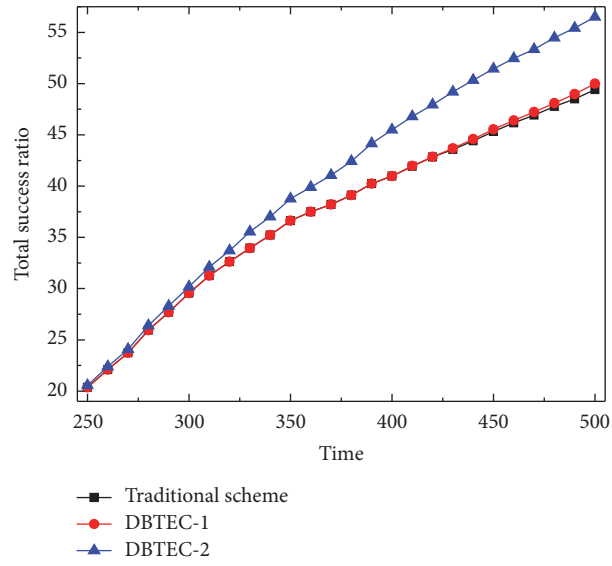

Figure 13: Time-total success ratio.

vehicles. DBTEC-2 performs much better than DBTEC-1, especially in the stage of cold start. DBTEC schemes can take effects even in complicated situations.

Then, we analyze the influence of DBTEC-1 and DBTEC-2 on total success ratio of cooperation requests in this threat model. Figure 13 illustrates the increment of total success ratio as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis are the same as Figure 5. As illustrated by Figure 13, similar results will be obtained as in Figure 5; the positive influence of DBTEC-1 on total success ratio is comparatively small when compared with the positive influence of DBTEC-2. Particularly in the stage of cold start, the total success ratio of DBTEC-1 is nearly equal to the total success ratio of traditional scheme. DBTEC-2 outperforms DBTEC-1 and traditional scheme significantly in the total success ratio. In general, Figure 13 illustrates that DBTEC is better than traditional scheme in the total success ratio, DBTEC-2 increases much more total success ratio compared to DBTEC-1 and traditional scheme, and DBTEC-1 increases the total success ratio in a small amount.

Finally, we analyze the influence of DBTEC-1 and DBTEC-2 on the success ratio of each stage in this threat model. The experimental method is the same as Figure 6. Figure 14 illustrates the increment of the success ratio of each stage, namely, each 50 timestamps. The horizontal axis and the vertical axis are the same as Figure 6. As illustrated by Figure 14, similar results will be obtained as in Figure 6. The success ratio of stages of both traditional scheme and DBTEC schemes increases as time goes on. Both DBTEC-1 and DBTEC-2's increasing speed outperforms traditional scheme. DBTEC-2's increasing speed outperforms DBTEC-1 in every stage significantly; this phenomenon is significant especially in the stage of cold start. In general, Figure 14 illustrates that DBTEC' success ratio of each stage is larger compared to traditional scheme in each stage. DBTEC-2 increases larger

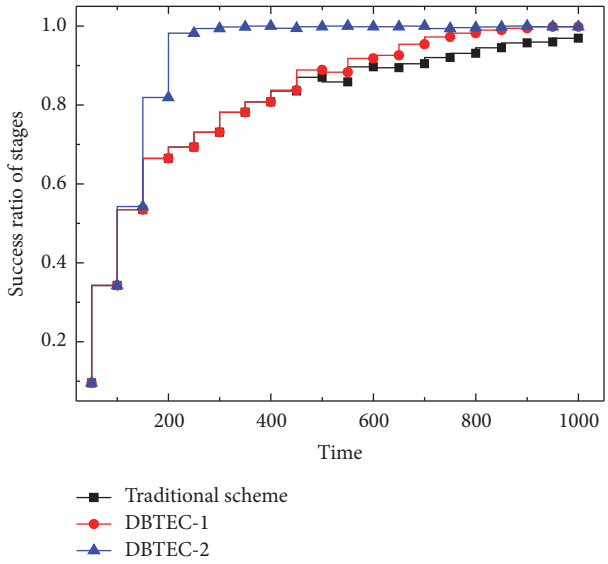

Figure 14: Time-success ratio of stages.

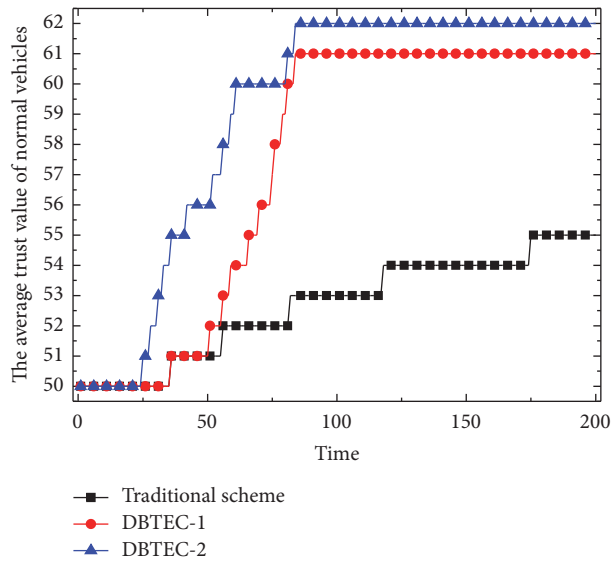

Figure 15: Time-average trust value of normal vehicles.

success ratio of each stage compared to DBTEC-1, especially in the stage of cold start.

5.3.4. Praising Partners and Slandering Normal Vehicles. In this model, which can be regarded as a stronger collusion attack than the previous model, malicious vehicles will report mendacious self-estimated service quality to Public board when registering, report low estimated service quality of normal vehicles to slander them even if these normal vehicles never provide services to them, and praise other malicious partners by reporting high estimated service quality of them. By slandering normal vehicles, malicious vehicles' opportunities of providing services increase indirectly. By collusively praising malicious partners, the overall number of opportunities of malicious vehicles increases significantly. Experimental results proof that DBTEC will also distinguish malicious vehicles and normal ones in this threat model; DBTEC-2 has better effects than DBTEC-1.

We first analyze the influence of DBTEC-1 and DBTEC-2 on average estimated trust value in this threat model.

Figure 15 illustrates the increment of average estimated trust value of normal vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 3. As illustrated by Figure 15, similar results will be obtained as in Figure 3. In traditional scheme, the increment of average estimated trust value of normal vehicles is slow, but when using DBTEC schemes the speed of increment increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of increment more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

Figure 16 illustrates the decrement of average estimated trust value of malicious vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively.

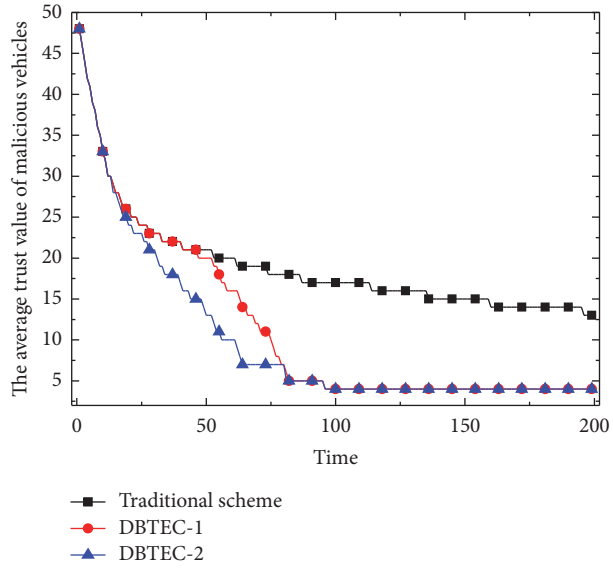

Figure 16: Time-average trust value of malicious vehicles.

The horizontal axis and the vertical axis of this figure are the same as Figure 4. As illustrated by Figure 16, similar results will be obtained as in Figure 4; in traditional scheme, the decrement of average estimated trust value of malicious vehicles is slow, but when using DBTEC schemes the speed of decrement increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of decrement more significantly compared to DBTEC-1, especially when in the stage of cold start, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

In general, Figures 15 and 16 illustrate that DBTEC has significantly positive influence on estimating trust value of vehicles. DBTEC-2 performs much better than DBTEC-1,

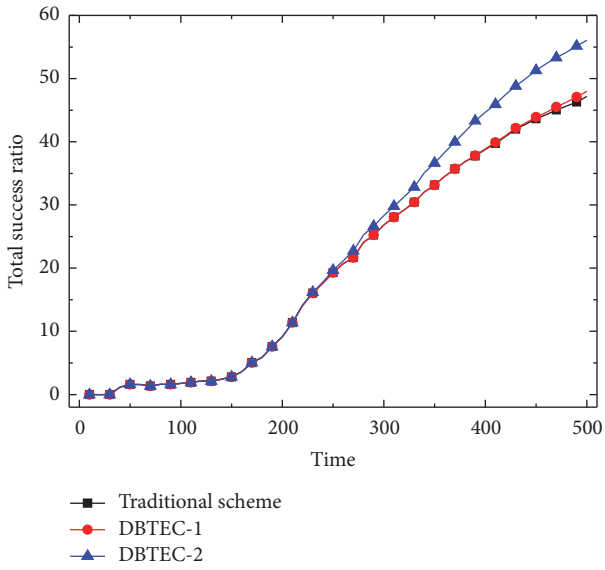

Figure 17: Time-total success ratio.

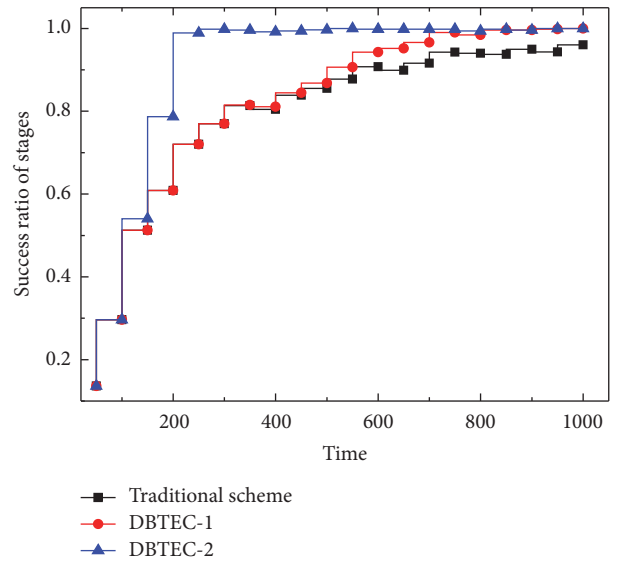

Figure 18: Time-success ratio of stages.

especially in the stage of cold start. DBTEC schemes can take effects even in complicated situations.

Then, we analyze the influence of DBTEC-1 and DBTEC-2 on total success ratio of cooperation requests in this threat model. Figure 17 illustrates the increment of total success ratio as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis are the same as Figure 5. As illustrated by Figure 17, similar results will be obtained as in Figure 5; the positive influence of DBTEC-1 on total success ratio is comparatively small when compared with the positive influence of DBTEC-2. Particularly in the stage of cold start, the total success ratio of DBTEC-1 is nearly equal to the total success ratio of traditional scheme. DBTEC-2 outperforms DBTEC-1 and traditional scheme significantly in the total success ratio. In general, Figure 17 illustrates that DBTEC is better than traditional scheme in the total success ratio, DBTEC-2 increases much more total success ratio than DBTEC-1 and traditional scheme, and DBTEC-1 increases the total success ratio in a small amount.

Finally, we analyze the influence of DBTEC-1 and DBTEC-2 on the success ratio of each stage in this threat model. The experimental method is the same as Figure 6. Figure 18 illustrates the increment of the success ratio of each stage, namely, each 50 timestamps. The horizontal axis and the vertical axis are the same as Figure 6. As illustrated by Figure 18, similar results will be obtained as in Figure 6. The success ratio of stages of both traditional scheme and DBTEC schemes increases as time goes on. Both DBTEC-1 and DBTEC-2's increasing speed outperforms traditional scheme. DBTEC-2's increasing speed outperforms DBTEC-1 in every stage significantly; this phenomenon is significant especially in the stage of cold start. In general, Figure 18 illustrates that DBTEC' success ratio of each stage is larger than traditional scheme in each stage. DBTEC-2 increases larger success ratio of each stage compared to DBTEC-1, especially in the stage of cold start.

5.3.5. Providing Unstable Services. This threat model is the most complicated because of the disguise of malicious vehicles. When registering, malicious vehicles will report a mendacious self-estimated service quality to Public board; as time goes on, malicious vehicles have an unstable performance. Sometimes, they will act just the same as normal vehicles, but, sometimes, they will provide abnormal services, such as extremely poor service quality. Experimental results proof that DBTEC will also distinguish malicious vehicles and normal ones in this threat model. DBTEC-2 has better effects than DBTEC-1.

We first analyze the influence of DBTEC-1 and DBTEC-2 on average estimated trust value in this threat model.

Figure 19 illustrates the increment of average estimated trust value of normal vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 3. As illustrated by Figure 19, similar results will be obtained as in Figure 3. In traditional scheme, the increment of average estimated trust value of normal vehicles is slow, but when using DBTEC schemes the speed of increment increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of increment more significantly compared to DBTEC-1, especially when in the stage of cold start, and the final average estimated trust value is larger than DBTEC-1, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

Figure 20 illustrates the decrement of average estimated trust value of malicious vehicles as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis of this figure are the same as Figure 4. As illustrated by Figure 20, similar results will be obtained as in Figure 4; in traditional scheme,

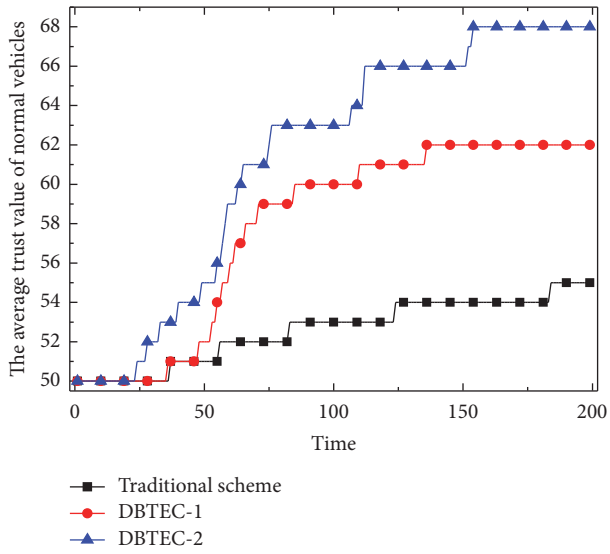

Figure 19: Time-average trust value of normal vehicles.

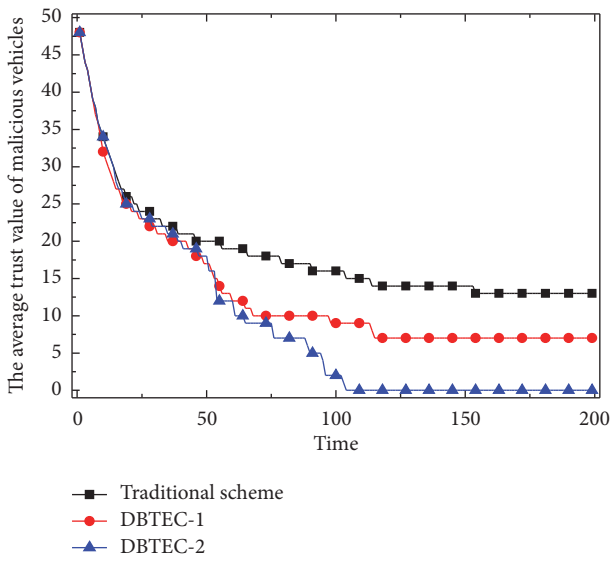

Figure 20: Time-average trust value of malicious vehicles.

the decrement of average estimated trust value of malicious vehicles is slow, but when using DBTEC schemes the speed of decrement increases significantly. When comparing DBTEC-1 and DBTEC-2, it is obvious that DBTEC-2 increases the speed of decrement more significantly compared to DBTEC-1, especially when in the stage of cold start, and the final average estimated trust value is less than DBTEC-1, which proves the improvement of DBTEC-2 as illustrated in Section 4.5.2.

In general, Figures 19 and 20 illustrate that DBTEC has significantly positive influence on estimating trust value of vehicles. DBTEC-2 performs much better than DBTEC-1, especially in the stage of cold start. DBTEC schemes can take effects even in these complicated situations.

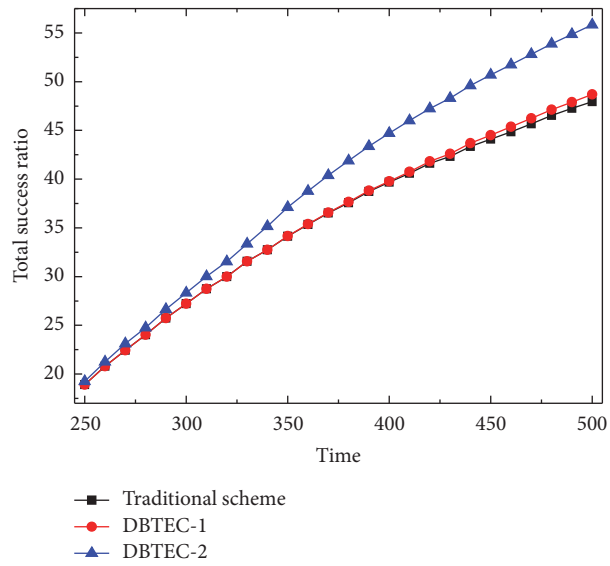

Figure 21: Time-total success ratio.

Then, we analyze the influence of DBTEC-1 and DBTEC-2 on total success ratio of cooperation requests in this threat model. Figure 21 illustrates the increment of total success ratio as time goes on using traditional scheme, DBTEC-1, and DBTEC-2, respectively. The horizontal axis and the vertical axis are the same as Figure 5. As illustrated by Figure 21, similar results will be obtained as in Figure 5; the positive influence of DBTEC-1 on total success ratio is comparatively small when compared with the positive influence of DBTEC-2. Particularly in the stage of cold start, the total success ratio of DBTEC-1 is nearly equal to the total success ratio of traditional scheme. DBTEC-2 outperforms DBTEC-1 and traditional scheme significantly in the total success ratio. In general, Figure 21 illustrates that DBTEC is better than traditional scheme in the total success ratio, DBTEC-2 increases much more total success ratio than DBTEC-1 and traditional scheme, and DBTEC-1 increases the total success ratio in a small amount.

Finally, we analyze the influence of DBTEC-1 and DBTEC-2 on the success ratio of each stage in this threat model. The experimental method is the same as Figure 6. Figure 22 illustrates the increment of the success ratio of each stage, namely, each 50 timestamps. The horizontal axis and the vertical axis are the same as Figure 6. As illustrated by Figure 22, similar results will be obtained as in Figure 6. The success ratio of stages of both traditional scheme and DBTEC schemes increases as time goes on. Both DBTEC-1 and DBTEC-2's increasing speed outperforms traditional scheme. DBTEC-2's increasing speed outperforms DBTEC-1 in every stage significantly; this phenomenon is significant especially in the stage of cold start. In general, Figure 22 illustrates that DBTEC' success ratio of each stage is larger than traditional scheme in each stage. DBTEC-2 increases larger success ratio of each stage compared to DBTEC-1, especially in the stage of cold start.

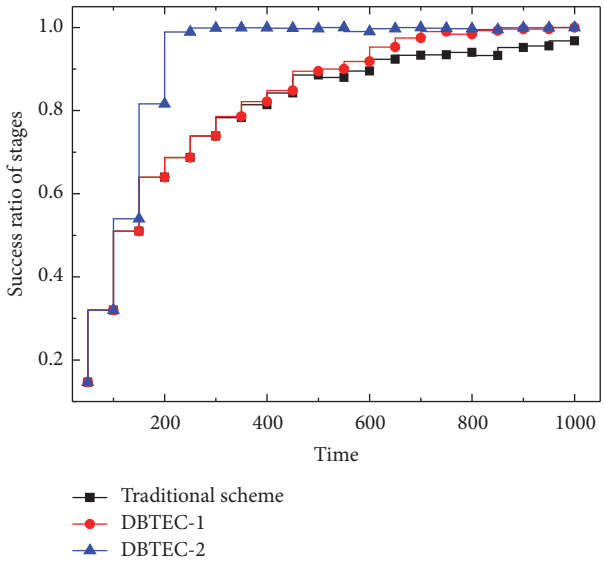

Figure 22: Time-success ratio of stages.

6. Conclusion

In this paper, we propose a trust-based security cooperation model, called DBTEC, which combines direct trust information stored in Private board with indirect trust information stored in Public board to guide the selection of cooperative partners in VCC. The experiments prove the effectiveness of DBTEC schemes.

With the advance of Internet of Things, the form of many practical applications, such as delivering physical objects, has changed. Vehicle networks have made extensive cooperation between vehicles possible. Security is a key requirement for cooperation. The DBTEC schemes give a better solution to security cooperation.

Notations

Φ: The set of all vehicles in VCC
n: The number of vehicles in VCC
V_i: The ith vehicle in VCC
Q: The set of service qualities of all vehicles
Q_i: The ith vehicle's self-estimated service quality
M: The set of malicious vehicles in VCC
N: The set of normal vehicles in VCC
h: The number of malicious vehicles in VCC
$E_j'^i$: The public estimated service quality for vehicle V_j reported by vehicle V_i
$T_j'^i$: The timestamp at which vehicle V_i's estimated service quality for vehicle V_j is reported
E_j^i: Estimated service quality for vehicle V_j in vehicle V_i's Private board
R_j^i: Estimated trust value for vehicle V_j in vehicle V_i's Private board
T_j^i: The timestamp at which vehicle V_i updates E_j^i
T_i^{col}: The timestamp at which vehicle V_i updates R_k^i from column perspective because of trusting in vehicle V_t
$T_{i,t\to k}^{\text{row}}$: The timestamp at which vehicle V_i updates R_k^i from row perspective because of trusting in vehicle V_t.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61379110, 61073104, 61572528, 61272494, and 61572526) and the National Basic Research Program of China (973 Program) (2014CB046305).

References

[1] R. Lu, X. Lin, X. Liang, and X. Shen, "A dynamic privacy-preserving key management scheme for location-based services in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 127–139, 2012.

[2] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-Time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 1664–1678, 2015.

[3] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors*, vol. 16, no. 1, article 88, 2016.

[4] Y. Hu and A. Liu, "Improvement the quality of mobile target detection through portion of node with fully duty cycle in WSNs," *Computer Systems Science and Engineering*, vol. 31, no. 9, pp. 5–17, 2016.

[5] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1931–1942, 2013.

[6] Y. Liu, A. Liu, Y. Hu et al., "FFSC: an energy efficiency communications approach for delay minimizing in internet of things," *IEEE Access*, vol. 4, pp. 3775–3793, 2016.

[7] X. Liu, K. Ota, A. Liu, and Z. Chen, "An incentive game based evolutionary model for crowd sensing networks," *Peer-to-Peer Networking and Applications*, vol. 9, no. 4, pp. 692–711, 2016.

[8] H. Dai, G. Chen, C. Wang, S. Wang, X. Wu, and F. Wu, "Quality of energy provisioning for wireless power transfer," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 527–537, 2015.

[9] R. Hussain, Z. Rezaeifar, Y. Lee, and H. Oh, "Secure and privacy-aware traffic information as a service in VANET-based clouds," *Pervasive and Mobile Computing*, vol. 24, pp. 194–209, 2015.

[10] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: when QoE meets QoP," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 74–80, 2015.

[11] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1711–1720, 2016.

[12] A. Liu, Y. Hu, and Z. Chen, "An energy-efficient mobile target detection scheme with adjustable duty cycles in wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 22, no. 4, pp. 203–225, 2016.

[13] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2016.

[14] R. Xie, A. Liu, and J. Gao, "A residual energy aware schedule scheme for WSNs employing adjustable awake/sleep duty cycle," *Wireless Personal Communications*, vol. 90, no. 4, pp. 1859–1887, 2016.

[15] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2014.

[16] A. Liu, X. Liu, and Y. Liu, "A comprehensive analysis for fair probability marking based traceback approach in WSNs," *Security and Communication Networks*, vol. 9, no. 14, pp. 2448–2475, 2016.

[17] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.

[18] X. Liu, M. Dong, K. Ota, P. Hung, and A. Liu, "Service pricing decision in cyber-physical systems: insights from game theory," *IEEE Transactions on Services Computing*, vol. 9, no. 2, pp. 186–198, 2016.

[19] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: secure and trustable routing in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2013–2027, 2016.

[20] H. Dai, X. Wu, L. Xu, F. Wu, S. He, and G. Chen, "Practical scheduling for stochastic event capture in energy harvesting sensor networks," *International Journal of Sensor Networks*, vol. 18, no. 1-2, pp. 85–100, 2015.

[21] M. Dong, K. Ota, and A. Liu, "RMER: reliable and energy-efficient data collection for large-scale wireless sensor networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 511–519, 2016.

[22] J. Gui and Z. Zeng, "Joint network lifetime and delay optimization for topology control in heterogeneous wireless multi-hop networks," *Computer Communications*, vol. 59, pp. 24–36, 2015.

[23] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1632–1646, 2016.

[24] S. He, J. Chen, X. Li, X. S. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1268–1282, 2014.

[25] L. Yang, J. Cao, W. Zhu, and S. Tang, "Accurate and efficient object tracking based on passive RFID," *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2188–2200, 2015.

[26] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2364–2376, 2015.

[27] N. Haddadou, A. Rachedi, and Y. Ghamri-Doudane, "A job market signaling scheme for incentive and trust management in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3657–3674, 2015.

[28] Q. Yang and H. Wang, "Toward trustworthy vehicular social networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 42–47, 2015.

[29] K. Rostamzadeh, H. Nicanfar, N. Torabi, S. Gopalakrishnan, and V. C. 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.

[30] A. Liu, X. Liu, and J. Long, "A trust-based adaptive probability marking and storage traceback scheme for WSNs," *Sensors*, vol. 16, no. 4, article 451, 2016.

[31] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 640–651, ACM, Budapest, Hungary, May 2003.

[32] G. Yan, D. Wen, S. Olariu, and M. C. Weigle, "Security challenges in vehicular cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 284–294, 2013.