Trust-based Service Composition and Binding with Multiple Objective Optimization in Service-Oriented Mobile Ad Hoc Networks

Yating Wang†, Ing-Ray Chen†, Jin-Hee Cho*, Ananthram Swami* and Kevin S. Chan*

Abstract— With the proliferation of fairly powerful mobile devices and ubiquitous wireless technology, we see a transformation from traditional mobile ad hoc networks (MANETs) into a new era of service-oriented MANETs wherein a node can provide and receive services. Requested services must be decomposed into more abstract services and then bound; we formulate this as a multi-objective optimization (MOO) problem to minimize the service cost, while maximizing the quality of service and quality of information in the service a user receives. The MOO problem is an SP-to-service assignment problem. We propose a multidimensional trust based algorithm to solve the problem. We carry out an extensive suite of simulations to test the relative performance of the proposed trust-based algorithm against a non-trust-based counterpart and an existing single-trust-based beta reputation scheme. Our proposed algorithm effectively filters out malicious nodes exhibiting various attack behaviors by penalizing them with loss of reputation, which ultimately leads to high user satisfaction. Further, our proposed algorithm is efficient with linear runtime complexity while achieving a close-to-optimal solution.

Index Terms— service-oriented ad hoc networks, service composition, trust management, multi-objective optimization.

I. INTRODUCTION

A service-oriented mobile ad hoc network (MANET) is populated with service providers (SPs) and service requesters (SRs). A realization of service-oriented MANETs is a peer-to-peer service system with SPs providing web services and SRs requesting services, each requiring dynamic service composition and binding [26]. Unlike a traditional web service system in which nodes are connected to the Internet, nodes in service-oriented MANETs are mobile and an SR will need to request services from available SPs it encounters and with which it interacts dynamically. One can view a service-oriented MANET as an instance of Internet of Things (IoT) systems with a wide range of mobile applications including smart city, smart tourism, smart car, smart environmental monitoring, and healthcare [6]**.** It is particularly suitable to military MANET applications where nodes are mobile with multi-hop communication.

In this paper, we are concerned with satisfying user service requests with multiple objectives including maximizing quality-of-service (QoS) and quality-of-information (QoI) while minimizing the service cost with *user satisfaction* (US) ultimately measuring success. With a service request in hand, an SR has to first formulate a service composition plan based on the available SPs, and then determine the best node-to-service assignment for achieving multi-objective optimization (MOO). Dynamic service composition and binding is especially complicated in MANETs because of the space-time complexity of mobile devices (space is related to mobility and time is related to dynamic status change), and no existence of a trusted third party for centralized control. This issue is further compounded by the fact that nodes may exhibit malicious behavior (explained later in Section III.C) and the information received is often erroneous, uncertain and incomplete in MANET environments [12].

Our approach is to use trust [4], [5], [9], [10], [12], [13], [16], [24], [25] for decision making of service composition and binding. We demonstrate the resiliency and convergence properties of our trust protocol design for service-oriented MANETs in the presence of malicious nodes showing various attack behaviors (discussed in detail in Section III.C).

The unique contributions of the paper are as follows:

1. Our work is the first to propose a dynamic trust-based service composition and binding algorithm for MOO in service-oriented MANETs. Our proposed scheme has only linear runtime complexity for solution search but approaches the ideal performance obtainable by the Integer Linear Programming (ILP) solution which has exponential runtime complexity, and is thus applicable only to small-sized problems.

† Yating Wang and Ing-Ray Chen are with the Department of Computer Science, Virginia Tech, Falls Church, VA 22043. E-mail: {yatingw, irchen}@vt.edu.
**Jin-Hee Cho, Ananthram Swami, and Kevin S. Chan are with Computational and Information Sciences Directorate, U.S. Army Research Laboratory, Powder Mill Rd. Adelphi, MD 20783. E-mail: {jinhee.cho, ananthram.swami, kevin.s.chan}@us.army.mil*

2. We are the first to conduct a comparative performance analysis of non-trust vs. single-trust vs. multi-trust protocols for peer-to-peer trust evaluation in service-oriented MANETs. Trust-based service composition and binding has been studied in the web services domain [3], [15], [47], but only a single trust score on service quality was considered, although the single trust score may derive from multiple service quality metrics such as response time, throughput, availability, etc. This largely ignores the multidimensional concept of trust. Identifying proper trust components and forming the overall trust out of multiple trust components is critical to maximize application performance. We consider two key trust dimensions in service request execution, namely, *competence* and *integrity*, as the building blocks of a composite trust metric.
3. We use trust to effectively prevent malicious nodes from disrupting the operation of service-oriented MANETs. We conduct a detailed performance analysis and demonstrate that our trust-based algorithm can effectively filter out malicious nodes, which ultimately leads to high user satisfaction.

The rest of this paper is organized as follows. Section II surveys related work and contrasts our approach with existing work. We also explain the new contributions of this work compared to our published preliminary work [48]. Section III describes our system model, including the network model and the threat model. Section IV describes service composition and binding. Section V defines the problem and presents our solution methodology. Section VI discusses trust protocol designs considered in this work including the proposed multi-trust based algorithm, and an existing single-trust based algorithm [16], [24]. Section VII describes the three algorithm designs considered in this work. Section VIII reports the comparative performance analysis results of the three algorithms, and performs sensitivity analysis of the results with respect to key design parameters and their impact on resiliency. Section IX concludes the paper and outlines future research areas.

II. RELATED WORK

Service composition has been widely studied in Internet-based web services [8], [26], [27], [35], [36], [40], [42], [44], [52]. Service composition and binding comes in two forms: (a) *goal-oriented composition* where a goal and a set of available services are given and the system completes the goal by planning and service binding; and (b) *workflow-based composition* where the workflow with constraints is given as input [36]. Our work takes the latter approach, with service composition being formulated as a workflow problem (based on the user's location and the availability of SPs an SR encounters), and service binding being formulated as a node-to-service assignment problem.

MOO has also been extensively studied in Internet-based web services [41]. Wagner et al. [46] proposed a planning assistant that achieves MOO for three objectives including price, response time, and reliability by approximating Pareto-optimal solutions. Yu and Lin [50] studied MOO with end-to-end QoS objectives, including response time, service cost, availability and reliability. Alrifai and Risse [2] considered end-to-end QoS objectives during the runtime service binding process. Both [2] and [50] used a multi-choice multidimensional knapsack problem to formulate the MOO problem.

Weighted-sum is a common approach used in service composition with MOO. Yu et al. [51] addressed a service selection problem by aggregating multiple QoS objectives into a weighted utility function to be maximized subject to QoS resource needs with each weight representing the importance of each QoS attribute. Zeng et al. [52] considered five objectives, namely, price, duration, reliability, availability and reputation and formulated the MOO problem as a single objective problem using weighted sum. Similar to [51], [52], we adopt weighted sum to formulate our MOO problem for its simplicity and proven effectiveness.

The above cited work had a common drawback. Namely, their solutions have exponential time complexity because the MOO problem to be solved is NP-hard [17]. Our work remedies this problem by devising trust-based heuristic solutions that incur only linear runtime complexity, and verifying that the performance of our trust-based solution approaches the ideal performance obtainable by ILP solution.

Service composition in MANETs is still in its infancy. Existing work only focused on mechanisms for enabling service composition without considering MOO. Sheu et al. [38] designed a tactical pervasive service collaboration platform to enable service composition considering dynamic task arrival, data synchronization, and task failure. Johnsen et al. [21], [22] suggested a semantic service discovery solution with a rich and expressive service description provided to facilitate service composition. Wright et al. [49] proposed the use of UML 2.0 Activity Diagrams as a workflow specification language for describing service construction and composition. Suri [43] studied the deployment issues of service-oriented architectures and developed a middleware to facilitate dynamic service composition to cope with those issues. Compared with the above cited work, our work is the first to propose a dynamic service composition and binding algorithm using multi-trust protocols for MOO in service-oriented MANETs.

Singh [39] indicated that trust is an important factor in service-oriented computing where user experience is the main factor for trust establishment. Bansal et al. [3] and Dai et al. [15] proposed trust-based web service composition, but only a single dimension of trust was considered. Relative to the above work, we consider multi-trust, recognizing multi-dimensional trust assessment is critical to decision making. We demonstrate that our multi-trust-based

algorithm outperforms its single-trust-based counterparts such as BRS [16], [24].

Wahab et al. [47] provided an excellent review of trust and reputation models for Web services. Mehdi et al. [30], [31], [32] considered multiple QoS metrics (e.g., response time, throughput, availability, etc.) for assessing the service quality of a web service. A trust score is derived from combining multiple QoS metrics to assess the trustworthiness of a web service. This single trust score considered in [30], [31], [32] in effect corresponds to the *competence* trust score considered in our work. In addition, we also consider the *integrity* trust score (the other metric of our multi-trust design) for measuring the degree to which a node complies with the prescribed service protocol. Khosravifar et al. [28] considered multiple reputation factors (user satisfaction and popularity) and analyzed their relationships. We do not consider popularity as a trust metric. Instead, we consider integrity as a trust metric to cope with malicious attacks. Hang et al. [18], [20] modeled a composite service as a statistical mixture, and showed that their approach can dynamically punish or reward the constituents of composite services while making only partial observations. Hang et al. [19] later developed a probabilistic trust model considering not only the trust level, but also the amount of evidence supporting the trust level. They showed that their trust model yields higher prediction accuracy than traditional approaches suffering from situations in which witnesses are unreachable or are reachable only by untrustworthy referrals. In our work, we consider confidence in the context of trust formation. That is, integrity trust is used as confidence to assess the validity of competence trust based on the rationale that competence trust ultimately ensures service success. This is discussed in more detail in Section VI.B.

This work is substantially extended from our preliminary work [48] as follows: (a) we apply a scaling technique to scale service quality metrics so that all service quality metrics are normalized to the same scale and order, and the MOO problem may be formulated as a maximization problem; (b) we develop new single-trust and multi-trust protocols and examine their impact on performance with respect to key design parameters; (c) we add a new simulation study for small-sized to large-sized MOO problems for simulation validation, and analyze the effect of node and operation characteristics on performance; (d) we analyze the effect of node and network dynamics on both trust-based and non-trust-based algorithms; (e) we devise heuristic-based solutions which yield linear runtime complexity for solution efficiency without sacrificing solution optimality compared with the optimal solution generated by ILP; and (f) we conduct extensive sensitivity analysis to identify conditions under which each of the studied schemes perform best: the proposed multi-trust-based scheme, and the non-trust-based and single-trust-based counterparts.

III. SYSTEM MODEL

A. Service Provider and Service Requester Model

We consider a service-oriented MANET in which a node has two roles: a service provider (SP) for abstract services it is capable of providing, and a service requestor (SR) for issuing service requests on behalf of its owner. Conceptually an SR is like a user in service-oriented MANETs. A user can issue a sequence of service requests as it moves from one location to another. An example is a user in a smart city who first issues a service request "take me to a nice Thai restaurant nearby with drunken noodle on its menu" with a service quality specified in terms of QoI, QoS, and cost for the overall service request (e.g., the cost and duration of travel), as well as for individual abstract services (e.g., cost of drunken noodle). Once she finishes her meal, she issues another service request "take me to a nice night club in town" again with a minimum service quality specified in terms of QoI, QoS, and cost. Each of these service requests involves a service composition phase to compose a service plan out of the transportation services (e.g., taxi, bus, subway, etc.) and Thai food/night club services available to the user, followed by a service binding phase to select the best SPs out of all SPs available to the user at the time the service request is issued. The overall goal of the user is to maximize the QoS and QoI, while minimizing the cost for all requested services.

We consider a service-oriented MANET environment with $|\mathcal{N}|$ nodes moving according to the small world in motion (SWIM) mobility model [29]. We select SWIM because it captures key properties of human mobility in social network settings. Mobility introduces dynamic topology changes and affects the reliability of packet routing over multiple hops from a source to a destination. In particular, it affects the success probability of recommendation packet delivery which in turn affects trust protocol performance. To conserve resources, we assume that only a single copy of the recommendation about a target node (node j) is transmitted from the recommender node (node k) to the trustor (node i). Then, the recommendation packet from node k is lost when there is no route to reach node i from any intermediate node because of topology changes, when there is a channel error with probability $p_{e,}$ or when any intermediate node maliciously performs packet dropping attacks.

B. Service Quality Criteria

Without loss of generality, we consider three service quality criteria: QoI, service delay (as a QoS attribute), and cost. We denote them by Q, D, and C which may be measured after service invocations are performed. While D and C are easily measureable physical quantities, Q is specific to the application domain. For example, in environment monitoring service, Q is measured by the extent to which the output contributes to the ground truth data [45]. In sensing service, Q is measured by the extent to

which the sensing data contributes to the ground truth picture.

We first scale our service quality metrics, Q, D and C, to the range [0, 1] so that the higher the value, the better the quality [52], as follows:

$$\bar{Q} = \frac{Q - Q_{min}}{Q_{max} - Q_{min}}; \quad (1)$$

$$\bar{D} = \frac{D_{max} - D}{D_{max} - D_{min}}; \ \bar{C} = \frac{C_{max} - C}{C_{max} - C_{min}}$$

Here Q_{max} and Q_{min}, D_{max} and D_{min}, and C_{max} and C_{min} are the maximum and minimum possible values of Q, D, and C, respectively,. They are known a priori. With this normalization we transform MOO into multi-objective maximization, i.e., from maximizing Q and minimizing D and C, into maximizing $\bar{Q}$, $\bar{D}$ and $\bar{C}$. From a pragmatic perspective, scaling facilitates a fair quantitative comparison of different service quality criteria, as each service quality criterion is in the range of [0, 1] with a higher value representing a higher service quality.

C. Threat Model

Just like Internet-based web services, in a service-oriented MANET there are malicious SPs acting for their own gain. The common goal of malicious nodes is to increase their chance of being selected during a service binding phase. Malicious nodes can collude to achieve this common goal. We assume that a malicious node exhibits the following behaviors:

1. *Self-promotion*: it can promote its importance by reporting false service quality information in QoI, QoS, and cost (i.e., Q, D, and C) so as to increase its chance to be selected as the SP, but then provide opportunistic service.
2. *Opportunistic service*: it can provide "just enough" service to meet the minimum quality service requirement and user satisfaction expectation to improve the chance of the service request being completed successfully for it to gain good reputation.
3. *Bad-mouthing attack* (BMA): it can collude with other malicious nodes to ruin the reputation of a good node by providing bad recommendations so as to decrease the chance of this good node being selected to provide services.
4. *Ballot stuffing attack* (BSA): it can collude with other malicious nodes to boost the reputation of a bad node by providing good recommendations for the bad node so as to increase the chance of it being selected to provide services.
5. *Packet dropping*: it may drop packets passing through it during packet routing if the source node is a good node so as to launch a bad reputation attack against the source node.

A malicious node may also perform data modification attacks to ruin the reputation of a good node. PKI with assured digital signing [14] can be used to ensure data trustworthiness via source authenticity, integrity, and non-repudiation. A malicious node may also perform denial of service (DoS) attacks to overwhelm an SP. Counter-DoS mechanisms [1], [33] can be used to make DoS attacks largely ineffective to mitigate such attacks.

IV. Service Composition and Binding

A. Service Advertisement

A node as an SP advertises its service availability when a peer node (i.e., an SR) shows interest [26], [27]. An SP responds with an advertisement message only if it is capable of providing the requested services. Specifically, it responds with an advertisement message Ad_{SP} comprising four-tuple records, one for each abstract service S_k it can provide, as follows:

$$Ad_{SP}: [k, Q_k, D_k, C_k] \text{ for } S_k \quad (2)$$

Here k indicates the index of the service S_k; Q_k the level of QoI the SP can provide; D_k the level of service delay (for QoS); and C_k the service cost.

B. Dynamic Service Composition

For convenience, we use m to index service requests, k to index services, and i, j, or r to index nodes. We also use the notation O_m to refer to service request m, and the notation SR_m to refer to the SR who issues m. A service request (e.g., take me to a nice Thai restaurant nearby with drunken noodle on its menu) requires a number of abstract services $S_k's$ (e.g., transportation service, food service, etc.). For each service request in hand, the SR broadcasts the set of abstract services needed to which all qualified SPs respond with the 4-tuple records in (2). Based on the responses received, the SR then constructs a service composition specification (SCS) to specify the service plan for satisfying the service request. An example SCS is:

$$SCS_m = \langle [S_0], [S_2, S_4], [S_3], [S_7], [S_4, S_8], [S_2] \rangle \quad (3)$$

where $[S_2, S_4]$ specifies that S_2 and S_4 are to be executed concurrently; $[S_3], [S_7]$ specifies that S_3 and S_7 are to be executed sequentially. The user also specifies a minimum service quality requirement at the service request level and at the abstract service level as follows:

$$SCS_m^{THRES} = (Q_m^{THRES}, D_m^{THRES}, C_m^{THRES}) \quad (4)$$

$$S_k^{THRES} = (Q_k^{THRES}, D_k^{THRES}, C_k^{THRES})$$

The SR then decides the best SPs among all responders to execute SCS_m, while meeting the minimum service quality levels at both the service request level and the abstract service level. If the minimum service quality constraint is not satisfied, then it means that there are not enough qualified SPs available to provide service and O_m is considered a failure.

C. Service Binding

An SP capable of providing multiple abstract services can be selected to execute multiple abstract services in a

service request. However, to avoid schedule conflicts among concurrent service requests (issued by multiple SRs) and to avoid degrading an SP's service quality due to heavy workloads, the SP can only commit to one service request. That is, the SP can only participate in one service request at a time to ensure its availability and commitment to a single service request.

V. PROBLEM DEFINITION AND METRICS

A. Problem Definition

Given that multiple SPs may meet the service threshold criteria in (4), an SR must choose SPs so as to maximize the aggregate $\overline{\mathrm{Q}}$, $\overline{\mathrm{D}}$ and $\overline{\mathrm{C}}$. An SCS for serving a service request is essentially a flow structure consisting of series or parallel substructures. For the SCS in (3), there is one series structure consisting of 6 substructures, $[S_0], [S_2, S_4], [S_3], [S_7], [S_4, S_8]$, and $[S_2]$, at the top level, and there are two parallel substructures, $[S_2, S_4]$ and $[S_4, S_8]$, at the bottom level. Let $\overline{\mathrm{Q}}_\mathrm{m}$, $\overline{\mathrm{D}}_\mathrm{m}$ and $\overline{\mathrm{C}}_\mathrm{m}$ be the scaled Q, D, and C scores of $\mathrm{O_m}$, and $\overline{\mathrm{Q}}_{\mathrm{m,S}}$, $\overline{\mathrm{D}}_{\mathrm{m,S}}$ and $\overline{\mathrm{C}}_{\mathrm{m,S}}$ be the scaled Q, D, and C scores of substructure S. The service quality of $\mathrm{O_m}$ measured by $\overline{\mathrm{Q}}_\mathrm{m}$, $\overline{\mathrm{D}}_\mathrm{m}$ and $\overline{\mathrm{C}}_\mathrm{m}$ (the larger the better) after service binding can be computed recursively as follows:

a) For a parallel structure S consisting of two concurrent substructures $[S_1, S_2]$, the maximum $\overline{\mathrm{Q}}$ and $\overline{\mathrm{D}}$ scores are limited by the minimum service quality score (which we want to avoid through node selection), and the maximum $\overline{\mathrm{C}}$ score is bounded by the sum of $\overline{\mathrm{C}}$ scores (since cost is additive), i.e.,

$$\begin{aligned} \overline{\mathrm{Q}}_{\mathrm{m,S}} &= \min\left(\overline{\mathrm{Q}}_{\mathrm{m},S_1}, \overline{\mathrm{Q}}_{\mathrm{m},S_2}\right); \\ \overline{\mathrm{D}}_{\mathrm{m,S}} &= \min\left(\overline{\mathrm{D}}_{\mathrm{m},S_1}, \overline{\mathrm{D}}_{\mathrm{m},S_2}\right); \\ \overline{\mathrm{C}}_{\mathrm{m,S}} &= \overline{\mathrm{C}}_{\mathrm{m},S_1} + \overline{\mathrm{C}}_{\mathrm{m},S_2}. \end{aligned} \tag{5}$$

When combining scaled Q or D scores of two concurrent substructures using the min operator, the minimum of scaled Q scores turns out to be the scaled minimum of the unscaled Q scores. That is, $\min\left(\overline{\mathrm{Q}}_{\mathrm{m},S_1}, \overline{\mathrm{Q}}_{\mathrm{m},S_2}\right) = \min\left(\frac{\mathrm{Q_{m,S_1}} - \mathrm{Q_{min}}}{\mathrm{Q_{max}} - \mathrm{Q_{min}}}, \frac{\mathrm{Q_{m,S_2}} - \mathrm{Q_{min}}}{\mathrm{Q_{max}} - \mathrm{Q_{min}}}\right) = \frac{\min(\mathrm{Q_{m,S_1}}, \mathrm{Q_{m,S_2}}) - \mathrm{Q_{min}}}{\mathrm{Q_{max}} - \mathrm{Q_{min}}}$. The combined scaled Q score stays in the range of [0, 1], if each substructure is a single abstract service at the bottom level of an SCS. When combining scaled C scores of two concurrent substructures, we use the addition operator because each substructure will unavoidably incur a separate service cost which must be accounted for. The combined C score as a result of using the addition operator is no longer scaled in [0, 1].

b) For a series structure S consisting of two sequential substructures $[S_1], [S_2]$, the maximum score is limited by the sum of service quality scores, i.e.,

$$\begin{aligned} \overline{\mathrm{Q}}_{\mathrm{m,S}} &= \overline{\mathrm{Q}}_{\mathrm{m},S_1} + \overline{\mathrm{Q}}_{\mathrm{m},S_2}; \\ \overline{\mathrm{D}}_{\mathrm{m,S}} &= \overline{\mathrm{D}}_{\mathrm{m},S_1} + \overline{\mathrm{D}}_{\mathrm{m},S_2}; \\ \overline{\mathrm{C}}_{\mathrm{m,S}} &= \overline{\mathrm{C}}_{\mathrm{m},S_1} + \overline{\mathrm{C}}_{\mathrm{m},S_2}. \end{aligned} \tag{6}$$

When combining scaled Q, D, or C scores of two sequential substructures, we use the addition operator because the two substructures will be sequentially executed and each score (which we want to maximize through node selection) must be separately accounted for. The combined score as a result of using the addition operator is no longer scaled in [0, 1].

Here we note that at the bottom level of an SCS, a substructure is only an abstract service. If node j is selected to bind to this abstract service then $\overline{\mathrm{Q}}_{\mathrm{m,S}} = \overline{\mathrm{Q}}_{\mathrm{m,j}}$, $\overline{\mathrm{D}}_{\mathrm{m,S}} = \overline{\mathrm{D}}_{\mathrm{m,j}}$ and $\overline{\mathrm{C}}_{\mathrm{m,S}} = \overline{\mathrm{C}}_{\mathrm{m,j}}$. The top level, on the other hand, is either a series substructure or a parallel substructure. Let θ be the top level substructure of this SCS for $\mathrm{O_m}$. Then, the overall service quality score of $\mathrm{O_m}$ (after service binding) is given by:

$$\overline{\mathrm{Q}}_\mathrm{m} = \overline{\mathrm{Q}}_{\mathrm{m},\theta};\ \overline{\mathrm{D}}_\mathrm{m} = \overline{\mathrm{D}}_{\mathrm{m},\theta};\ \overline{\mathrm{C}}_\mathrm{m} = \overline{\mathrm{C}}_{\mathrm{m},\theta} \tag{7}$$

B. MOO Problem Formulation

We use the *weighted sum* form [37] allowing a user to express its preferences regarding service quality criteria. Let $\omega_{\mathrm{Q,m}}$, $\omega_{\mathrm{D,m}}$ and $\omega_{\mathrm{C,m}}$ be the weights associated with $\overline{\mathrm{Q}}_\mathrm{m}$, $\overline{\mathrm{D}}_\mathrm{m}$ and $\overline{\mathrm{C}}_\mathrm{m}$ for $\mathrm{O_m}$ issued by the user, with $\omega_{\mathrm{Q,m}} + \omega_{\mathrm{D,m}} + \omega_{\mathrm{C,m}} = 1$. Another compelling justification of using weighted sum is that expressing the optimization criterion of a multi-objective problem by means of a weighted sum corresponds to a Lagrangian formulation [7] with multiple Lagrange multipliers, thereby effectively sweeping the lower convex envelope of the objective surface. With this simple additive weighting technique, we formulate our MOO problem at the service-request level as:

$$\text{Maximize } \mathrm{MOO_m} = \omega_{\mathrm{Q,m}}\overline{\mathrm{Q}}_\mathrm{m} + \omega_{\mathrm{D,m}}\overline{\mathrm{D}}_\mathrm{m} + \omega_{\mathrm{C,m}}\overline{\mathrm{C}}_\mathrm{m} \tag{8}$$

subject to the service request level constraint $\mathrm{SCS_m^{THRES}}$ and the abstract service level constraint $\mathrm{S_k^{THRES}}$ specified in (4) by the user. As there may be multiple SRs issuing service requests and performing service composition and binding concurrently, we formulate our MOO problem at the system level as:

$$\text{Maximize } \mathrm{MOO} = \sum_{\mathrm{m}\in\mathcal{T}} (\omega_{\mathrm{Q,m}}\overline{\mathrm{Q}}_\mathrm{m} + \omega_{\mathrm{D,m}}\overline{\mathrm{D}}_\mathrm{m} + \omega_{\mathrm{C,m}}\overline{\mathrm{C}}_\mathrm{m}) \tag{9}$$

where $\mathcal{T}$ is the set of concurrent service requests issued by multiple SRs who are competing for the use of SPs available to them. It is noteworthy that (8) and (9) solve the service binding problem, given a service composition specification (SCS) formulated as in (3).

C. MOO Value and User Satisfaction as Performance Metrics

While the final MOO value defined in (9) above can be used to measure MOO performance, *user satisfaction* ultimately determines if a service request is a success or a

failure. The *user satisfaction* level of the SR toward SPs selected for executing O_m,denoted as US_m, can be measured by the ratio of the actual service quality received to the best service quality available among SPs for executing O_m. We allow a user to specify a minimum *user satisfaction threshold*, denoted as UST_m, which specifies the minimum service quality the user can accept. This is to be compared against US_m to decide if the service experience of the user toward SPs selected for executing O_m is positive or negative. If $O_m$fails because of failing to satisfy the service request level constraint SCS_m^{THRES}, then US_m is zero. If the service experience is negative, culprit SPs are identified and penalized with reputation loss. Conversely, if the service experience is positive, all constituent SPs are rewarded with reputation gain based on US_m obtained. For notational convenience, let $\overline{SQ}_m^R = \omega_{Q,m}\overline{Q}_m^R + \omega_{D,m}\overline{D}_m^R + \omega_{C,m}\overline{C}_m^R$ denoting the actual service quality received after service binding and execution of O_m, $\overline{SQ}_m^{max} = \omega_{Q,m}\overline{Q}_m^{max} + \omega_{D,m}\overline{D}_m^{max} + \omega_{C,m}\overline{C}_m^{max}$ denoting the best service quality that can ever be achieved, and $\overline{SQ}_m^{min} = \omega_{Q,m}\overline{Q}_m^{THRES} + \omega_{D,m}\overline{D}_m^{THRES} + \omega_{C,m}\overline{C}_m^{THRES}$ denoting the minimum service quality that must be obtained in order to satisfy the service request level constraint SCS_m^{THRES}. Then, with score scaling, US_m can be computed as:

$$US_m = \begin{cases} \dfrac{\overline{SQ}_m^R - \overline{SQ}_m^{min}}{\overline{SQ}_m^{max} - \overline{SQ}_m^{min}} & \text{if } \overline{SQ}_m^R \geq \overline{SQ}_m^{min} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Here $\overline{SQ}_m^R$, $\overline{SQ}_m^{max}$and $\overline{SQ}_m^{min}$ are the received, maximum, and minimum service quality scores, respectively, for executing O_m, calculated based on $\overline{Q}_m$, $\overline{D}_m$ and $\overline{C}_m$in (7). The second condition in (10) is for the case in which the received service quality is less than the required minimum service quality. Again we note that because of scaling, the large the $\overline{Q}$, $\overline{D}$ and $\overline{C}$ values, the better the service quality. Also note that $\overline{SQ}_m^R = \omega_{Q,m}\overline{Q}_m^R + \omega_{D,m}\overline{D}_m^R + \omega_{C,m}\overline{C}_m^R$, so maximizing MOO_min (8) is equivalent to maximizing US_min (10). Therefore, the MOO problem to solve is in effect a user satisfaction maximization problem.

VI. TRUST MANAGEMENT PROTOCOL

In this section, we first discuss a well-known trust management scheme based on the single-trust beta reputation system (BRS) [16], [24], as the baseline scheme against which our multi-trust protocol will be compared. We choose BRS because of its sound statistical basis compared to other schemes using intuitive and ad-hoc methods for measuring trust. In addition, it enables a trustor to ensure tractability of trust evidence over time. We note that a trust model based on Dirichlet distribution [23], [31], which is a generalization of BRS, can also be used as the single-trust baseline scheme for performance comparison. However, since a user can specify a minimum *user satisfaction threshold* to decide if a service experience is positive or negative (a binary classification), BRS suffices. We then describe our trust protocol with multi-trust design.

A. Single-trust Baseline Protocol Design

The baseline BRS protocol is based on Bayesian inference with the trust value modeled as a random variable in the range of [0, 1] following the Beta (α, β) distribution; the numbers of positive and negative experiences are modeled as binomial random variables. Since the beta-binomial is a conjugate pair, this leads to a posterior beta distribution with updated parameters. Here α/(α+β) is the estimated mean of "direct" trust evidence of an SP where α is the number of positive interactions and β is the number of negative interactions. A positive evidence is observed when SR_m is satisfied. More specifically, when US_m exceeds UST_m, it is counted as positive evidence and all constituting SPs in O_m are rewarded. In the case of positive evidence, α is incremented by 1 for all SPs in O_m. On the other hand, when US_m is less than UST_m, SR_m identifies the culprits with low performance (i.e., the actual service quality is lower than the advertised service quality) and considers it as negative evidence against these culprits. In this case, β is increased by 1 for all identified culprits. SPs with expected performance (i.e., the actual service quality is about the same as the advertised service quality) are identified as benign and will not be penalized. After a service request is completed, the SR propagates its updated trust of the SPs involved in the service request to other nodes in the system. See Appendix A of the supplemental file for details of trust propagation and aggregation in the single-trust baseline protocol design, allowing a node receiving a trust update to update its (α, β) pair toward an SP.

B. Multi-trust Protocol Design

Our trust management protocol design centers on the concept of multi-trust. Multi-trust refers to the use of multiple dimensions of trust for more accurately describing multiple and often distinct factors contributing to successful service execution. For service composition and binding, we choose two unique trust properties: competence and integrity. We chose these two dimensions because the capability for service provision (i.e., competence) and the compliance to the prescribed service protocol (i.e., integrity) are the key criteria of quality service provision for high user satisfaction. The two trust dimensions are:

- **Competence**: This refers to an SP's capability to satisfactorily serve the received request. This is largely determined by the intrinsic service capability of the SP, as modeled by the SP's "true" Q, D, and C scores.
- **Integrity**: This refers to the degree to which a node complies with the prescribed protocol, including the service contract and the trust protocol. A node may violate its service contract when it performs self-promotion attacks. That is, a node lies about Q, D, and C scores of its own capability so as to increase its chance to be included in executing service requests but then it fails to honor the service contract or just performs opportunistic service once it is selected for service request execution.

We denote node i's trust toward node j in X (i.e., C for competence, and I for integrity) as $T_{i,j}^{X}$. We adopt BRS [16], [24] to assess node i's mean *direct trust* toward node j in trust property X as $\alpha_{i,j}^{X}/(\alpha_{i,j}^{X}+\beta_{i,j}^{X})$ where $\alpha_{i,j}^{X}$ is the number of positive and $\beta_{i,j}^{X}$ is the number of negative experiences in trust property X, which are accumulated upon trust update. To update $(\alpha_{i,j}^{C}, \beta_{i,j}^{C})$ for competence trust, node i (acting as the SR) compares UST_m with US_m as described in the baseline BRS scheme. To update $(\alpha_{i,j}^{I}, \beta_{i,j}^{I})$ for integrity trust, node i considers it positive evidence if it sees node j's observed Q, D and C scores are close to node j's advertised scaled Q, D, and C scores. Node i (as SR_m) assesses node j's compliance degree ($CD_{m,j}$) as:

$$CD_{m,j} = \min\left(\frac{\bar{Q}_{m,j}^{observed}}{\bar{Q}_{m,j}^{advertised}}, \frac{\bar{D}_{m,j}^{observed}}{\bar{D}_{m,j}^{advertised}}, \frac{\bar{C}_{m,j}^{observed}}{\bar{C}_{m,j}^{advertised}}\right) \quad (11)$$

Here $\bar{Q}_{m,j}^{advertised}$, $\bar{D}_{m,j}^{advertised}$ and $\bar{C}_{m,j}^{advertised}$ are node j's advertised "scaled" Q, D, and C scores, while $\bar{Q}_{m,j}^{observed}$, $\bar{D}_{m,j}^{observed}$ and $\bar{C}_{m,j}^{observed}$ are node j's "scaled" Q, D and C scores actually observed by node i (acting as SR_m) during O_m execution. Each user defines its minimum compliance degree threshold for service request m, denoted by CDT_m. If $CD_{m,j} \geq CDT_m$, then it is counted as a positive experience for node j for integrity and $\alpha_{i,j}^{I}$ is incremented by 1; otherwise, it is counted as a negative experience for node j and $\beta_{i,j}^{I}$ is incremented by 1. We note that this can effectively capture self-promotion attack behavior. Trust propagation and aggregation are again based on the concept of belief discounting [24], i.e., the $(\alpha_{i,j}^{X}, \beta_{i,j}^{X})$ pair of node i toward node j is merged with n_{rec} pairs of $(\alpha_{r,j}^{X}, \beta_{r,j}^{X})$ from n_{rec} recommenders whom node i trusts the most.

There are multiple ways to form the overall trust $T_{i,j}$ from $T_{i,j}^{C}$ and $T_{i,j}^{I}$. In this work, we explore the *Trust + Confidence* formation model by which integrity trust is used as confidence to assess the validity of competence trust based on the rationale that competence trust ultimately ensures service success. If integrity trust falls below a threshold, competence trust is invalid or scaled down. We investigate two relationships under this model: *drop-to-zero* and *scaling*. In the *threshold-based relationship model* (TRM), if integrity trust falls below a threshold, $T_{i,j}^{I,THRES}$, competence trust drops to zero. In the *scaling relationship model* (SRM), competence trust scales up (to 1 maximum) or down (to 0 minimum), depending on whether integrity trust is higher or lower than the threshold. The integrity threshold $T_{i,j}^{I,THRES}$ may be individual-based (for node i toward node j) in service-oriented MANETs populated with human operators. More specifically, TRM computes the overall trust as:

$$T_{i,j} = T_{i,j}^{C} \text{ if } T_{i,j}^{I} \geq T_{i,j}^{I,THRES}; \text{ 0 otherwise} \quad (12)$$

where $T_{i,j}^{I,THRES}$ is the minimum integrity trust threshold.

SRM computes the overall trust as:

$$T_{i,j} = \min\left(1, T_{i,j}^{C} \times \frac{T_{i,j}^{I}}{T_{i,j}^{I,THRES}}\right) \quad (13)$$

VII. ALGORITHM DESCRIPTION

In this section, we describe the three trust-based service composition and binding algorithms (based on BRS, TRM, and SRM), and the non-trust-based counterpart. After an SR formulates an SCS (e.g., (3)) based on available SPs for executing O_m, multiple solutions may exist to meet the service requirements and constraints. The SR then chooses a solution among all candidate solutions to maximize MOO_m in (8). The four algorithms are:

- ***Non-trust-based***: While there is no trust in place, each SR keeps a blacklist of SPs with which it has negative interaction experience, i.e., $CD_{m,j} < CDT_m$. When selecting the best node-to-service assignment, it only considers SPs that are not blacklisted.
- ***Trust-based*** (BRS, TRM and SRM): Each SR selects the best node-to-service assignment that maximizes MOO_m in (8) with $\bar{Q}_{m,j}$, $\bar{D}_{m,j}$ and $\bar{C}_{m,j}$ (of node j at the bottom level of the SCS defined in (5) and (6)) multiplying by $T_{SR,j}$ which is the SR's overall trust toward node j obtained from running a trust protocol (BRS, TRM or SRM) as discussed in Section V. The basic idea of trust-based service composition and binding is that an SP's advertised Q, D and C scores are discounted by the SR's trust towards the SP. When the trust estimate is accurate, it can effectively defend against malicious nodes performing attacks discussed in Section III.C.

Each algorithm described above can be solved by ILP (see Appendix B of the supplemental file for detail) with exponential runtime complexity of $O(2^{|\mathcal{N}|})$ where $|\mathcal{N}|$ is the number of SPs, assuming that the total number of abstract services in a set of concurrent service requests is much smaller than the number of SPs available.

To circumvent high runtime complexity which renders it infeasible for runtime operations, we develop heuristic-based solutions with linear runtime complexity of $O(|\mathcal{N}|)$ for solution efficiency. For all algorithms, an SR simply ranks all eligible SPs for executing an abstract service specified in SCS_m by $\omega_{Q,m}\bar{Q}_{m,j} + \omega_{D,m}\bar{D}_{m,j} + \omega_{C,m}\bar{C}_{m,j}$ and selects the highest ranked SP as the winner for executing that particular abstract service. It examines all SPs which responded to its query in a single round and performs ranking and service binding for all abstract services defined in its SCS_m. Then, the SR notifies SPs that are selected without coordination with other concurrent SRs. As a result, an SP may receive multiple offers from multiple SRs executing concurrent service requests.

In the *non-trust-based* algorithm, an SP receiving multiple offers randomly selects one SR among all to serve. In the trust-based algorithm, an SP resolves the tie-breaker by selecting the SR for which it has the highest trust to ensure the highest success probability as it will increase its

chance of gaining good reputation. The other SRs not selected will be informed of the decision by the SP and will then select other SPs that are still available to provide the particular abstract service. The time to complete the node-to-service selection thus is linear to the number of eligible SPs multiplied by the number of concurrent service requests because each SR will only examine and rank the advertised service quality scores by all eligible SPs once to select a subset of SPs that maximizes its own ranking.

Here we note that the heuristics employed by non-trust-based and trust-based algorithms for service binding are not necessarily optimal because the resulting solution may not maximize MOO_m in (8) or MOO defined in (9). As we shall see later, our heuristic design is able to achieve a solution quality approaching that generated by ILP but with significantly lower complexity.

TABLE I: Input Parameters Values/Ranges.

Parameter	Value	Parameter	Value
$\lvert\mathcal{T}\rvert$	15	$\lvert\boldsymbol{\mathcal{N}}\rvert$	60, 120, 240
$\lvert\boldsymbol{\mathcal{S}}\rvert$	9	p_e	5%
n_{rec}	3	$\lvert\mathcal{S}_m\rvert$	4, 8, 16
P_{bad}	10-50%	P_{risk}	0-100%
Q_{bad}	[1-3]	Q_{good}	[3-5]
D_{bad}	[3-5]	D_{good}	[1-4]
C_{bad}	[2-5]	C_{good}	[1-2]
$(Q_k^{THRES}, D_k^{THRES}, C_k^{THRES})$	(1,5,5)	$(Q_m^{THRES}, D_m^{THRES}, C_m^{THRES})$	(4,20,20)
$T_{i,j}^{I,THRES}$	0.5	CDT_m	50-100%
$\omega_{Q,m}$: $\omega_{D,m}$: $\omega_{C,m}$	1/3:1/3:1/3	UST_m	50-100%
(α, β) for BMA	(1, 10)	(α, β) for BSA	(10, 1)

VIII. RESULTS AND ANALYSIS

In this section, we evaluate performance of trust-based algorithms (with BRS, TRM or SRM for trust computation) vs. non-trust-based algorithms (with blacklisting).

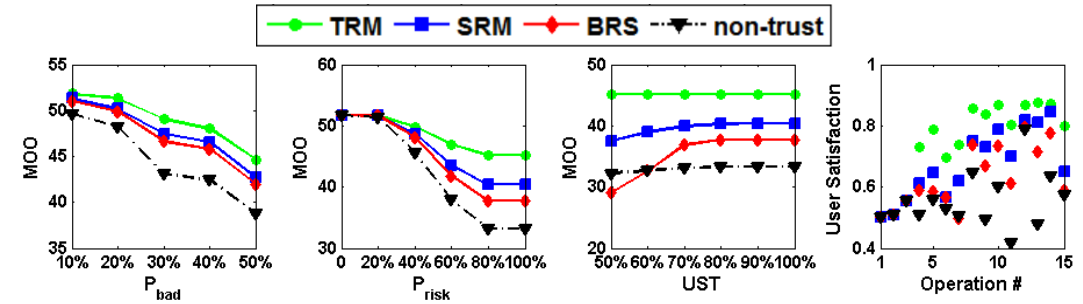
A. Experiment Setup

Table I lists input parameters and their default values for performance analysis. Below we explain each parameter in the table. For the small, medium, and large-sized problems, we consider $|\boldsymbol{\mathcal{N}}|$=60, 120, and 240 SPs respectively. There are $|\boldsymbol{\mathcal{S}}|$=9 abstract services, S_0 to S_8, provided by these SPs. For simplicity, each SP is assumed to be specialized to one abstract service S_k which is randomly assigned initially. All nodes follow the SWIM mobility model as described in Section III.A with p_e = 0.5%. We consider a scenario with 15 service requests ($|\mathcal{T}|$=15) divided into 9 sequential chunks, i.e., {1, 2}, {3}, {4, 5}, {6, 7, 8}, {9}, {10}, {11, 12}, {13}, and {14, 15}, where a chunk is defined as a set of concurrent service requests overlapping in execution time. The performance outcome is about the same when the number of chunks is more than 9 or when the number of service requests is longer (30 and 60), so these scenarios are not presented here. For simplicity, each service request has only one SCS consisting of $|\mathcal{S}_m|$ abstract services connected in a series structure, with $|\mathcal{S}_m|$= 4, 8 and 16 for small, medium and large sized problems, respectively. In our case study, we consider only one SCS in each service request. The abstract services are randomly selected from S_0 to S_8 so that the demand to each node is roughly equal in these different sized problems. In case an SR cannot find enough SPs to satisfy the SCS, the service request fails and US_m=0.

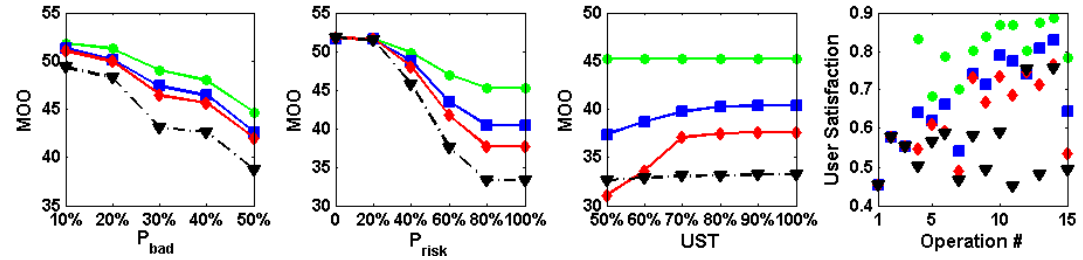
We model the hostility of the environment by the percentage of malicious nodes, P_{bad}, in the range of [0-50%] with the default value set at 30%. A malicious node performs all attacks as described in the threat model. In particular, the self-promotion attack behavior is modeled by a risk parameter, P_{risk}, in the range of [0-100%] with the default set at 50%. A malicious node performs self-promotion attacks by boosting its advertised Q, D and C scores by multiplying its true Q, D and C scores by $(1+P_{risk})$, $(1-P_{risk})$, and $(1-P_{risk})$, respectively. Q_{bad}, D_{bad} and C_{bad} give the *true* Q, D and C scores for bad nodes, which can be boosted during service advertisement. Q_{good}, D_{good} and C_{good} give the *true* Q, D and C scores for good nodes, which will be reported by good nodes faithfully during service advertisement. The Q, D, C values of a node are generated from a uniform distribution at the beginning and are not changed during system operation. The default values are set for the case in which the service quality of bad nodes is inferior to that of good nodes to reveal interesting design tradeoffs. Later we will perform a sensitivity analysis of the effect of Q, D, and C score distributions for good and bad nodes on performance.

We initially set $(Q_k^{THRES}, D_k^{THRES}, C_k^{THRES}) = (1, 5, 5)$ at the abstract service level and $(Q_m^{THRES}, D_m^{THRES}, C_m^{THRES}) = (4, 20, 20)$ at the service request level for the light minimum service quality constraint case. Later we will perform a sensitivity analysis of the effect of service quality constraints.

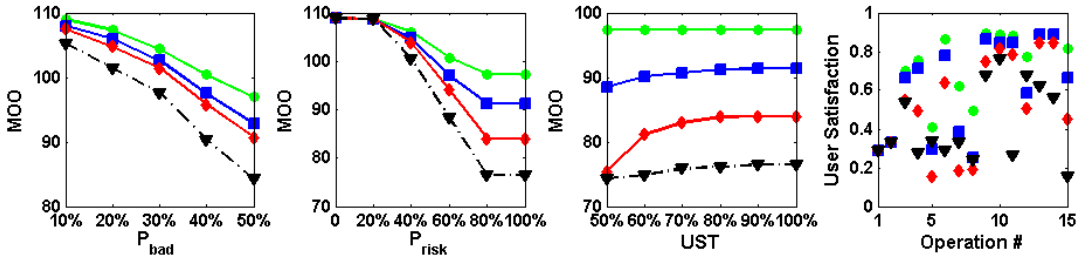
The initial trust values (for integrity and competence in the case of multi-trust) are set to 0.5 for all nodes meaning ignorance (no knowledge). The integrity trust threshold $T_{i,j}^{I,THRES}$ for TRM and SRM is set to 0.5 to punish a node with integrity trust less than ignorance trust as time progresses. The protocol compliance degree threshold CDT_m is set to 0.9 to accommodate a 10% maximum detection error for assessing protocol compliance behavior. For simplicity we set $\omega_{Q,m} = \omega_{D,m} = \omega_{C,m} = 1/3$ in (8). The number of recommenders n_{rec} is set to 3 so node i will only allow 3 nodes with the highest $T_{i,j}$ values as the recommenders during trust update. When a bad node is mistakenly chosen as a recommender, it can perform a bad-mouthing attack (BMA) on a good trustee node. This is done by providing a very low α, and a very high β, e.g., (α, β) = (1, 10), with α representing the number of positive and β the number of negative experiences, to ruin the reputation of the good trustee node. A bad node serving as a recommender can also perform a ballot-stuffing attack



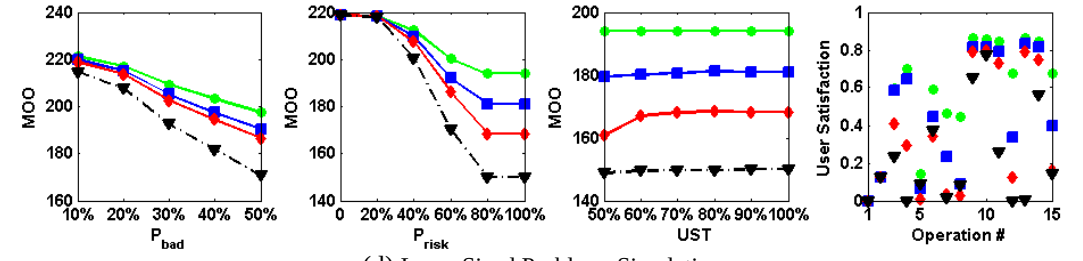
(a) Small Sized Problem - ILP.



(b) Small Sized Problem -Simulation.



(c) Medium Sized Problem - Simulation.



(d) Large Sized Problem -Simulation.

Fig. 1: Performance Comparison for Small, Medium and Large Sized MOO Problems.

(BSA) on a bad trustee node by providing a very high α, and a very low β, e.g., (α, β) = (10, 1), to boost the reputation of the bad trustee node.

B. Comparative Performance Analysis

In this section, we compare MOO performance of trust-based and non-trust-based algorithms via MATLAB simulation. Specifically, we simulate non-trust-based and trust-based algorithms (both have linear runtime complexity) under the same environment setting defined in Table I with $|\mathcal{N}|$=60, 120 and 240, and correspondingly $|\mathcal{S}_m|$ = 4, 8 and 16 for small, medium and large sized problems, respectively.

Fig. 1(a) shows the ILP results for the small model. Figs. 1(b)-(d) show the simulation results for the small, medium and large sized problems, respectively. The simulation results for the small sized problem in Fig. 1(b) are to be compared against the ILP analytical results in Fig. 2(a) to reveal the tradeoff between solution efficiency gained (because of linear runtime complexity) vs. solution optimality sacrificed (because of the use of heuristics). The ILP solution for generating the analytical results is listed in Appendix B of the supplemental file. Here we note that medium to large sized problems ($|\mathcal{N}|$=120 and 240) can only be evaluated by simulation since ILP is not able to generate a solution due to the high computational complexity. We observe that for the small-sized problem, the simulation results based on heuristic designs are remarkably similar to the ideal performance results both in shape and value. This demonstrates that the heuristic design can achieve solution efficiency without sacrificing solution optimality too much.

The leftmost two graphs in each row of Fig. 1 examine the negative impact of increasing P_{bad} and P_{risk} on MOO performance in (9). Compared with the non-trust-based algorithm, trust-based algorithms show high resilience against increased attack intensity with more malicious entities (P_{bad}) or higher self-promotion attack behavior (P_{risk}). TRM has the best MOO performance among all because the drop-to-zero trust based on trust threshold can effectively filter out bad nodes.

The 3rd graph of each row in Fig. 1 examines the impact of UST_m on MOO performance. Recall that UST_m is to be compared with US_m to determine if a service experience is positive or negative for trust assessment for BRS and for competence trust assessment for SRM and TRM. We observe that as UST_m increases, MOO performance increases (and levels off). The reason is that a high UST_m has the effect of penalizing bad nodes with low trust and can effectively remove bad nodes from participating in future service requests. Another noteworthy observation is that unlike BRS and SRM, TRM is relatively insensitive to UST_m. We attribute the insensitivity to TRM severely punishing a bad node (i.e., dropping its trust to zero) once a bad node's integrity trust falls below the minimum ignorance trust of 0.5.

The rightmost graph of each row in Fig. 1 compares US_m calculated from (10) as more service requests (labeled as "Operation #" on the x coordinate) are executed over time for the three trust-based algorithms against the non-trust-based algorithm. We consider a combination of P_{risk}=70% and P_{bad}=30% to reveal interesting trends. Because of high P_{risk}, even trust-based algorithms are fooled into selecting bad nodes in the first few service requests. So the first few service requests do not pass UST_m. As a result, bad nodes selected to provide services in the first few service requests are penalized with trust decrease and likely to be filtered out from later service requests. This is evidenced by the result that the last 8 service requests have high US_m values. In particular, for multi-trust TRM, US_m is above 90% for the last 8 service requests because mostly only good nodes are being selected by the trust-based algorithm due to dynamic trust update. On the contrary, the non-trust-based algorithm consistently yields a low US_m service request by service request because it has no effective way of filtering out bad nodes. This trend supports our claim that trust-based algorithms can effectively achieve high user satisfaction despite the presence of bad nodes performing self-promotion attacks, especially after trust convergence occurs. In particular, we observe that US_m under TRM or SRM is consistently higher than that under BRS. We attribute this to the use of integrity trust as confidence for competence trust, thus providing a more accurate estimate of the trustworthy service quality of an SP. Furthermore, TRM consistently outperforms SRM because of its ability to discern bad nodes from good nodes. It is worth noting that one main reason for having a low US_m in a service request is that the SR does not have enough information about the reputation of SPs bidding for services. Consequently, the SR must select SPs for service request execution based on their advertised scores. This is a serious problem for the non-trust-based algorithm because SRs do not share experiences through recommendations. As a result, we see that the non-trust-based algorithm has the most severe zigzag pattern of US_m among all. In particular, the zigzag pattern is most pronounced for service requests 5, 9, 11, 13 and 15 at which the responsible SRs have never issued a service request before.

As we go from small to large sized problems, we observe a remarkable similarity with TRM outperforming SRM and BRS. This demonstrates the scalability of our trust algorithm design. That is, our trust-based algorithm especially with TRM as the underlying trust protocol can find near optimal solutions with linear runtime complexity as the problem size increases from small ($|\mathcal{N}|$=60 and $|\mathcal{S}_m|$=4), to medium ($|\mathcal{N}|$=120 and $|\mathcal{S}_m|$=8) and large ($|\mathcal{N}|$=240 and $|\mathcal{S}_m|$=16).

TABLE II: Minimum Service Quality Constraints.

Case	$(Q_k^{THRES}, D_k^{THRES}, C_k^{THRES})$	$(Q_m^{THRES}, D_m^{THRES}, C_m^{THRES})$
Light	(1, 5, 5)	(4, 20, 20)
Medium	(2, 4, 3)	(8, 16, 12)
Tight	(3, 3, 2)	(12, 12, 8)

C. Effect of Service Quality Constraints and Opportunistic Service Attacks

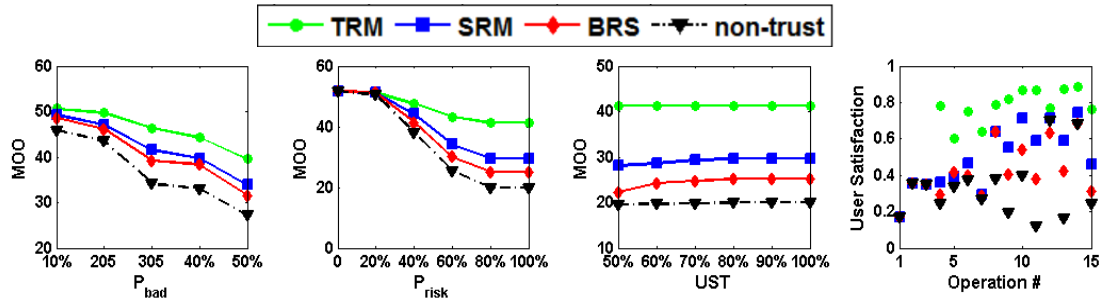
In this section, we perform a sensitivity analysis of the results with respect to the minimum service quality constraints including the minimum service quality requirement at the abstract service level $S_k^{THRES} = (Q_k^{THRES}, D_k^{THRES}, C_k^{THRES})$ and the minimum service quality requirement at the service request level $SCS_m^{THRES} = (Q_m^{THRES}, D_m^{THRES}, C_m^{THRES})$ specified by the user as in (4).

Table II lists three conditions: light, moderate, and tight. Service quality constraints induce "opportunistic service" attack behaviors in two ways.

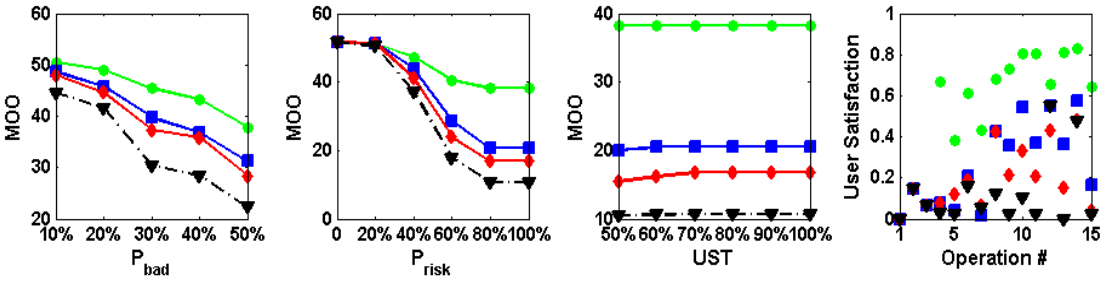
First, a bad node may want to lie about its service quality to at least pass the abstract service level Q, D, C constraints in order for it to have a chance to be selected for service request execution. Second, once a bad node is selected for a service request, it is strongly motivated to contribute at least a minimum effort to satisfy the service request level constraint; otherwise, the service request is considered a failure from the user's perspective and the malicious node will be penalized with a trust loss.

Figs. 2(a)-(c) show the simulation results for the light, medium, and tight minimum service quality constraint cases, respectively, for the small sized problem.

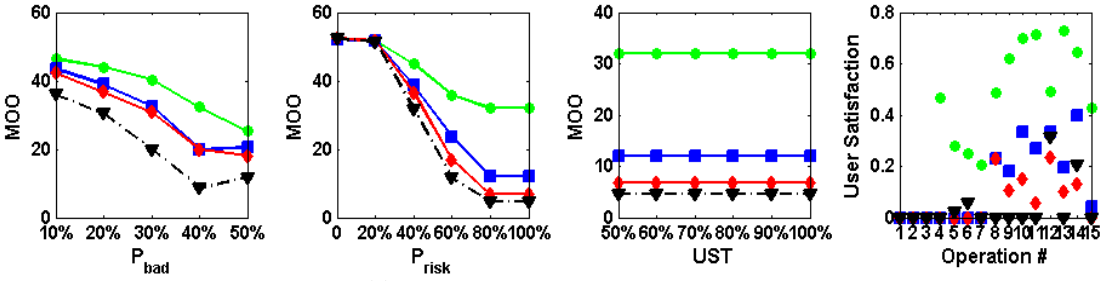
For comparison, the light minimum service quality constraint case shown in Fig. 2(a) corresponds to Fig. 1(b) except that a bad node will perform opportunistic service attacks. Even with this opportunistic service attack behavior, we observe from Fig. 2 that the relative performance rank of TRM, SRM, BRS and non-trust-based



(a) Light Service Quality Constraints.



(b) Medium Service Quality Constraints.



(c) Tight Service Quality Constraints.

Fig. 2: Effect of Service Quality Constraints and Opportunistic Service Attacks.

design (in this order) remains the same. A major reason is that a bad node often must perform self-promotion attacks first to increase its chance of being selected after which it can perform opportunistic service attacks. However, opportunistic service attacks cannot offset low user satisfaction toward the bad node resulting from self-promotion attacks. We observe from Fig. 2 that TRM outperforms SRM and BRS because of its ability to discern bad nodes from good nodes even in the presence of opportunistic service attacks.

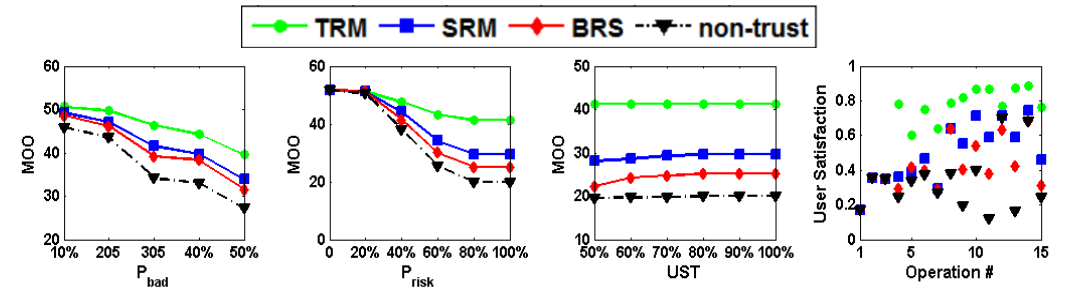
TABLE III: Q, D, C Score Distribution.

Case	$(Q_{bad}, D_{bad}, C_{bad})$	$(Q_{good}, D_{good}, C_{good})$
Good nodes have better service quality	$([1-3],[3-5],[2-5])$	$([3-5],[1-4],[1-2])$
Equal	$([3-5],[1-4],[1-2])$	$([3-5],[1-4],[1-2])$

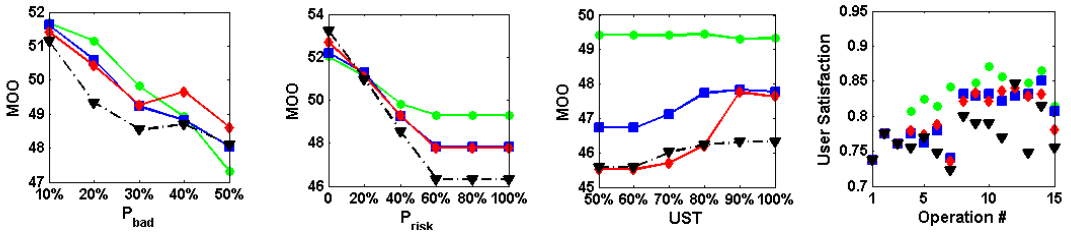
D. Effect of Q, D, C Score Distribution

In this section, we perform a sensitivity analysis of the results with respect to Q, D, C score distributions for bad nodes and good nodes in terms of $(Q_{bad}, D_{bad}, C_{bad})$ and $(Q_{good}, D_{good}, C_{good})$. Table III lists two test cases: "good nodes have better service quality" vs. "equal" (good nodes and bad nodes have equal service quality). We ignore the case in which bad nodes have better service quality than good nodes because bad nodes will prevail anyway even with trust-based service composition and binding. All other conditions remain the same as specified in Table I. Note that the results presented so far are based on the "good nodes have better service quality" case.

Figs. 3(a)-(b) show the simulation results for the "good nodes have better service quality" and "equal" test cases, for the small sized problem. Fig. 3(a) is the same as Fig. 1(b) and is replicated here for ease of comparison. We see that in the case of equal service quality, i.e., Fig. 3(b), trust-based design still outperforms non-trust-based design with the same performance ranking preserved. However, we see that BRS is just as good as SRM, especially when UST is high and/or more service requests are executed. The reason is that both BRS and SRM in this equal service quality case tend to select moderate to high service quality nodes, regardless of whether they are good or bad nodes. Although SRM also applies scaling trust reward/penalty to good/bad nodes based on integrity trust information, this small differentiation does not prevent bad nodes with medium to high service quality from being selected. As a result, they both select moderate to high service quality nodes for service request execution.



(a) Good Nodes have Better Service Quality.



(b) Good Nodes and Bad Node have Equal Service Quality.

Fig. 3: Effect of Q, D, and C Score Distributions for Good and Bad Nodes on Performance.

On the other hand, TRM applies a rather strict trust penalty to bad nodes, so it essentially excludes bad nodes from selection and only good nodes with high service quality are selected for service request execution. The results exhibited in Fig. 3 reveal conditions ("good nodes have better service quality" vs. "equal") under which our trust-based algorithm design is effective compared with the non-trust-based algorithm design. More specifically, with the service quality about being equal for both good and bad nodes, TRM performs the best when there are more good nodes than bad nodes. This is so because of TRM's unique ability to discern bad nodes from good nodes and to select only good nodes with high service quality. When there are more bad nodes than good nodes, on the other hand, BRS performs the best especially when bad nodes do not need to lie about their service quality for them to be selected for service request execution (i.e., when P_{risk} is low) because in this case BRS, due to its inability to discern bad nodes from good nodes, tends to select high service quality nodes for service request execution even if they are bad nodes.

Another interesting result is that when bad nodes have about the same service quality as good nodes, there is an optimal UST level under which the performance of our trust-based algorithm is maximized. This is evident from the MOO vs. UST graph (the 2nd rightmost graph) in Fig. 3(b). The reason is that if UST is too high (e.g., 100%), US<UST is true and bad nodes with high service quality originally selected for service request execution will be identified as culprits and will be penalized with trust degradation. This in effect will block bad nodes from participating in future service request executions. Consequently, the system will be forced to select only good nodes for servicing future service request. Since not all good nodes are of high service quality, inevitably the resulting MOO value is not as high as it would be if only high service quality nodes (good or bad) are selected for service request execution. In effect, UST can be used as a design parameter to maximize the MOO value.

IX. CONCLUSION

We proposed a trust-based service composition and service binding algorithm with *linear runtime complexity* for multi-objective optimization in service-oriented MANETs characterized by space-time complexity of mobile devices wherein nodes may be malicious and/or information received is often erroneous, partly trusted, uncertain and incomplete.

We investigated single-trust and multi-trust as the building block for our trust-based algorithm design and demonstrated that our proposed algorithm outperforms the non-trust-based and single-trust-based counterparts, and approaches the ideal solution quality obtainable by the ILP solution. We also performed sensitivity analysis to identify conditions under which trust-based service composition is most effective compared with non-trust-based service composition. We discovered that our threshold based model (TRM) performs the best among all because TRM can best discern bad nodes from good nodes. Our analysis result backed by simulation reveals that in case good nodes have higher service quality than bad nodes, multi-trust protocols and in particular TRM, which severely penalizes malicious attack behavior, will perform the best. However, in case bad nodes have equal service quality as good nodes, TRM will perform the best only when there are more good nodes than bad nodes. Otherwise, single-trust protocols such as the beta reputation system will perform the best, especially when bad nodes do not need to lie about their service quality for them to be selected for service request execution. In the latter case, there exists an optimal user satisfaction threshold in service quality under which the protocol performance is maximized.

In the future, we plan to leverage game theory to capture the dynamics between attacker/defense behaviors [11], [34] and reason how a service requester can perform counterattacks by choosing the best user satisfaction threshold (UST) and integrity threshold ($T_{i,j}^{I,THRES}$) for achieving multi-objective optimization of service quality. Another future direction is to seek real databases to validate simulation results.

ACKNOWLEDGEMENTS

This work was supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number W911NF-12-1-0445. This research was also partially supported by the Department of Defense (DoD) through the office of the Assistant Secretary of Defense for Research and Engineering (ASD (R&E)). The views and opinions of the authors do not reflect those of the DoD or ASD (R&E).

REFERENCES

[1] I. Aad, J.-P. Hubaux, and E.W. Knightly, "Impact of Denail of Service Attacks on Ad Hoc Networks," *IEEE/ACM Trans. Networking*, vol. 16, no. 4, 2008.

[2] M. Alrifai and T. Risse, "Combining Global Optimization with Local Selection for Efficient QoS-Aware Service Composition," in *Proc. 18th Int'l Conf. World Wide Web*, 2009, pp. 881-990.

[3] S.K. Bansal, A. Bansal, and M.B. Blake, "Trust-based dynamic web service composition using social network analysis," in *Proc. 2010 IEEE Int'l Workshop Business Applications of Social Network Analysis*, 2010.

[4] F. Bao and I.R. Chen, "Dynamic trust management for internet of things applications," in *Proc. 2012 ACM International Workshop on Self-Aware Internet of Things*, 2012, pp. 1-6.

[5] F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Trust-based Intrusion Detection in Wireless Sensor Networks," in *IEEE International Conference on Computer Communications*, 2011, pp. 1-6.

[6] E. Borgia, "The Internet of Things Vision: Key Features, Applications and Open Issues," *Computer Communications*, vol. 54, pp. 1-31, December 2014.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.

[8] D. Chakraborty, Y. Yesha, and A. Joshi, "A Distributed Service Composition Protocol for Pervasive Environments," in *IEEE Wireless Communications and Networking Conf.*, 2004, pp. 2575-2580.

[9] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 5, 2014, pp. 1200-1210.

[10] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Trust Management for Encounter-Based Routing in Delay Tolerant Networks," in *Proc. 2010 IEEE Global Comm. Conf.*, 2010.

[11] J.H. Cho, I.R. Chen, and P.G. Feng, "Effect of Intrusion Detection on Reliability of Mission-Oriented Mobile Group Systems in Mobile Ad Hoc Networks," *IEEE Trans. Reliability*, vol. 59, no. 1, pp. 231-241, 2010.

[12] J.H. Cho, A. Swami, and I.R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE Comm. Surveys and Tutorials*, vol. 13, no. 4, pp. 562-583, 2011.

[13] J.H. Cho, A. Swami, and I.R. Chen, "Modeling and Analysis of Trust Management for Cognitive Mission-Driven Group Communication Systems in Mobile Ad Hoc Networks," in *Int'l. Conf. Computational Science and Engineering*, 2009, pp. 641-650.

[14] W. Dai, T.P. Parker, H. Jin, and S. Xu, "Enhancing Data Trustworthiness via Assured Digital Signing," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 6, 2012.

[15] G. Dai and Y. Wang, "Trust-Aware Component Service Selection Algorithm in Service Compositio," in *Proc. 4th Int'l. Conf. Frontier of Computer Science and Technology*, 2009.

[16] S. Ganeriwal, L.K. Balzano, and M.B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *ACM Trans. Sensor Networks*, vol. 4, no. 3, article no. 15, pp. 1-37, 2008.

[17] C. Glaßer, C. Reitwießner, H. Schmitz, and M. Witek, "Approximability and Hardness in Multi-Objective Optimization," in *Programs, Proofs, Processes.*: Springer, 2010, pp. 180-189.

[18] C.W. Hang and M.P. Singh, "Trustworthy Service Selection and Composition," *ACM Trans. Autonomous and Adaptive Systems*, vol. 6, no. 1, article no. 5, pp. 1-17, 2011.

[19] C.W. Hang, Z. Zhang, and M.P.Singh, "Shin: Generalized Trust Propagation with Limited Evidence," *IEEE Computer*, vol. 46, no. 3, pp. 78-85, 2013.

[20] C.W. Huang, A.K. Kalia, and M.P. Singh, "Behind the Curtain: Service Selection via Trust in Composite Services," in *IEEE 19th Int'l Conf. Web Services*, 2012, pp. 9-16.

[21] F.T. Johnsen, J. Flathagen, and T. Hafsoe, "Pervasive Service Discovery Across Heterogeneous Tactical Networks," in *IEEE 2009 Military Comm. Conf.*, 2009, pp. 1-8.

[22] F.T. Johnsen, M. Rustad, T. Hafsoe, A. Eggen, and T. Gagnes, "Semantic Service Discovery for Interoperability in Tactical Military Networks," *The Int'l C2 J.*, vol. 4, no. 1, 2010, pp. 1-24.

[23] A. Jøsang and J. Haller, "Dirichlet Reputation Systems," in *Proc. 2nd Int'l. Conf. Availability, Reliability and Security*, 2007, pp. 112-119.

[24] A. Jøsang and R. Ismail, "The Beta Reputation System," in *15th Bled Electronic Commerce Conf.*, 2002, pp. 1-14.

[25] A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618-644, 2007.

[26] S. Kalasapur, M. Kumar, and B.A. Shirazi, "Dynamic Service Composition in Pervasive Computing," *IEEE Trans. Parallel and Distributed Systems*, pp. 907-918, 2012.

[27] E. Karmouch and A. Nayak, "A Distributed Constraint Satisfaction Problem Approach to Virtual Device Composition," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 11, pp. 1997-2009, 2012.

[28] B. Khosravifar, J. Bentahar, and A. Moazin, "Analyzing the Relations between Some Parameters of Web Services Reputation," in *IEEE 2010 Int'l Conf. Web Services*, 2010, pp. 329-336.

[29] S. Kosta, A. Mei, and J. Stefa, "Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking," in *7th IEEE Comm. Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks*, 2010.

[30] M. Mehdi, N. Bougnuila, and J. Bentahar, "Correlated Multi-Dimensional QoS Metrics for Trust Evaluation Within Web Services," in *Proc. 13th International Conference on Autonomous Agents and Multiagent Systems*, 2014, pp. 1605-1606.

[31] M. Mehdi, N. Bouguila, and J. Bentabar, "Probabilistic Approach for QoS-Aware Recommender System for Trustworthy Web Service Selection," *Applied Intelligence*, vol. 41, no. 2, pp. 503-524, 2014.

[32] M. Mehdi, N. Bouguila, and J. Bentahar, "QoS-Based Reputation Feedback Fusion under Unknown Correlation," in *Adaptive and Intelligent Systems.*: Springer, 2014, pp. 172-181.

[33] R. Mitchell and I.R. Chen, "A Survey of Intrusion Detection in Wireless Network Applications," *Computer Communications*, vol. 42, pp. 1-23, 2014.

[34] R. Mitchell and I.R. Chen, "Behavior-Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1254-1263, 2013.

[35] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," *LNCS*, vol. 3387, pp. 43-54, 2005.

[36] C. Sandionigi, D. Ardagna, G. Cugola, and C. Ghezzi, "Optimizing Service Selection and Allocation in Situational Computing Applications," *IEEE Trans. Services Computing*, vol. 6, no. 3, pp. 414-428, 2013.

[37] S. Sariel-Talay, T.R. Balch, and N. Erdogan, "A Generic Framework for Distributed Multirobot Cooperation," *J. of Intelligent and Robotic Systems*, vol. 63, no. 2, pp. 323-358, 2011.

[38] R-Y. Sheu et al., "Real-time Multi-agent Adaptive Tactical Pervasive Service-Oriented Collaborations (TAPS)," in *ACM Int'l Conf. Pervasive Services*, 2010, pp. 1-8.

[39] M.P. Singh, "Trustworthy Service Composition: Challenges and Research Questions," *LNCS*, vol. 2631, pp. 39-52, 2003.

[40] B. Srivastava and J. Koehler, "Web Service Composition - Current Solutions and Open Problems," in *Proc. 13rd Int'l Conf. Automated Planning and Scheduling Workshop Planning for Web Service*, 2003, pp. 28-35.

[41] A. Strunk, "QoS-Aware Service Composition: A Survey," in *Proc. 8th European Conference on Web Services*, 2010, pp. 67-74.

[42] P. Sun, "Service Composition and Optimal Selection with Trust Constraints," in *Proc. 2010 IEEE Asia-Pacific Services Computing Conf.*, 2010, pp. 645-653.

[43] N. Suri, "Dynamic Service-Oriented Architectures for Tactical Edge Networks," in *The 4th Workshop Emerging Web Service Technology*, 2009, pp. 3-10.

[44] F. Tao, "Resource Service Composition and Its Optimal Selection Based on Particle Swarm Optimization in Manufacturing Grid System," *IEEE Trans. Industrial Informatics*, vol. 4, no. 4, pp. 311-327, 2008.

[45] D.J. Thornley, R. Young, and J. Richardson, "From Mission Specification to Quality of Information Measures Closing the Loop in Military Sensor Networks," in *Proc. Annual Conference of International Technology Allianc*, 2008.

[46] F. Wagner, B. Kl, F. Ishikawa, and S. Honiden, "Towards Robust Service Compositions in The Context of Functionally Diverse Services," in *Proc. 21st Int'l Conf. World Wide Web*, 2012, pp. 969-978.

[47] O.A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "A Survey on Trust and Reputation Models for Web Services: Single, Composite, and Communities," *Decision Support Systems*, vol. 74, pp. 121-134, 2015.

[48] Y. Wang, I.R. Chen, J.H. Cho, K.S. Chan, and A. Swami, "Trust-Based Service Composition and Binding for Tactical Networks with Multiple Objectives," in *Proc. 32th IEEE Military Comm. Conf.*, 2013.

[49] J. Wright et al., "A Model-Driven Approach to The Construction, Composition and Analysis of Services on Sensor Networks," in *Proc. Annual Conference of International Technology Alliance*, 2010.

[50] T. Yu and K.-J. Lin, "A Broker-Based Framework for QoS-aware Web Service Composition," in *Proc. IEEE Int'l Conf. e-Technology, e-Commerce and e-Service*, 2005, pp. 22-29.

[51] T. Yu, Y. Zhang, and K.J. Lin, "Efficient Algorithms for Web Services Seslection with End-to-End QoS Constraints," *ACM Transactions on the Web*, vol. 1, no. 1, article no. 6, pp. 1-26, May 2007.

[52] L. Zeng et al., "QoS-Aware Middleware for Web Services Composition," *IEEE Trans. Software Engineering*, vol. 30, no. 5, pp. 311-327, 2004.

AUTHOR BIOGRAPHIES

Yating Wang received her Bachelor degree from Hubei University of Technology, Wuhan, China in 2007. Her research interests include security, computer networks, wireless networks, mobile computing, trust management, and reliability and performance analysis. Currently she is pursuing her Ph.D. degree in the Computer Science Department at Virginia Tech.

Ing-Ray Chen received the BS degree from the National Taiwan University, and the MS and PhD degrees in computer science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, wireless systems, security, trust management, and reliability and performance analysis. Dr. Chen currently serves as an editor for *IEEE Communications Letters*, *IEEE Transactions on Network and Service Management*, *The Computer Journal*, and *Security and Network Communications*.

Jin-Hee Cho received the BA from the Ewha Womans University, Seoul, Korea and the MS and PhD degrees in computer science from the Virginia Tech. She is currently a computer scientist at the U.S. Army Research Laboratory, Adelphi, Maryland. Her research interests include wireless mobile networks, mobile ad hoc networks, sensor networks, secure group communications, group key management, network security, intrusion detection, performance analysis, trust management, cognitive networks, social networks, dynamic networks, and resource allocation. She is a senior member of the IEEE and a member of the ACM.

Ananthram Swami is with the U.S. Army Research Laboratory (ARL) as the Army's ST (Senior Research Scientist) for Network Science. He is an ARL Fellow and Fellow of the IEEE. He has held positions with Unocal Corporation, USC, CS-3 and Malgudi Systems. He was a Statistical Consultant to the California Lottery, developed a MATLAB-based toolbox for non-Gaussian signal processing, and has held visiting faculty positions at INP, Toulouse and Imperial College, London. He received the B.Tech. degree from IIT-Bombay, the M.S. degree from Rice University, and the Ph.D. degree from the University of Southern California (USC), all in Electrical Engineering. His research interests are in the broad area of network science.

Kevin Chan is a research scientist with the Computational and Information Sciences Directorate at the U.S. Army Research Laboratory (Adelphi, MD). Previously, he was an ORAU postdoctoral research fellow at ARL. His research interests are in network science, with past work in dynamic networks, trust and distributed decision making and quality of information through ARL's Network Science Collaborative Technology Alliance and Network and Information Sciences International Technology Alliance. Prior to ARL, he received a PhD in Electrical and Computer Engineering (ECE) and MSECE from Georgia Institute of Technology (Atlanta, GA). He also received a BS in ECE/EPP from Carnegie Mellon University (Pittsburgh, PA).