Decentralized Trust Management: Risk Analysis and Trust Aggregation

XINXIN FAN, Institute of Computing Technology, Chinese Academy of Sciences, China
LING LIU, School of Computer Science, Georgia Institute of Technology, USA
RUI ZHANG, Institute of Information Engineering, Chinese Academy of Sciences, China
QUANLIANG JING and JINGPING BI, Institute of Computing Technology, Chinese Academy of Sciences, China

Decentralized trust management is used as a referral benchmark for assisting decision making by human or intelligence machines in open collaborative systems. During any given period of time, each participant may only interact with a few other participants. Simply relying on direct trust may frequently resort to random team formation. Thus, trust aggregation becomes critical. It can leverage decentralized trust management to learn about indirect trust of every participant based on past transaction experiences. This article presents alternative designs of decentralized trust management and their efficiency and robustness from three perspectives. First, we study the risk factors and adverse effects of six common threat models. Second, we review the representative trust aggregation models and trust metrics. Third, we present an in-depth analysis and comparison of these reference trust aggregation methods with respect to effectiveness and robustness. We show our comparative study results through formal analysis and experimental evaluation. This comprehensive study advances the understanding of adverse effects of present and future threats and the robustness of different trust metrics. It may also serve as a guideline for research and development of next-generation trust aggregation algorithms and services in the anticipation of risk factors and mischievous threats.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Security and privacy** → **Trust frameworks**; • **Information systems** → *Collaborative and social computing systems and tools*;

Additional Key Words and Phrases: Trust management, adverse effect, threat risk, trust aggregation

ACM Reference format:
Xinxin Fan, Ling Liu, Rui Zhang, Quanliang Jing, and Jingping Bi. 2020. Decentralized Trust Management: Risk Analysis and Trust Aggregation. *ACM Comput. Surv.* 53, 1, Article 2 (February 2020), 33 pages.
https://doi.org/10.1145/3362168

The authors from Chinese Academy of Sciences are supported by the National Natural Science Foundation of China under Grants No. 61702470 and No. 61472403. The author from Georgia Institute of Technology, USA is partially funded by the USA National Natural Science Foundation under Grants No. 1547102 and No. SaTC 1564097 and an IBM faculty award.
Authors' addresses: X. Fan, Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexueyuan South Road, Beijing, 100190, China; email: fanxinxin@ict.ac.cn; L. Liu, School of Computer Science, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, Georgia, 30332-0765, USA; email: lingliu@cc.gatech.edu; R. Zhang, Institute of Information Engineering, Chinese Academy of Sciences, No. 89 Minzhuang Road, Beijing, China; email: zhangrui@iie.ac.cn; Q. Jing and J. Bi, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China; emails: {Jingquanliang, bjp}@ict.ac.cn.

0360-0300/2020/02-ART2 $15.00
https://doi.org/10.1145/3362168

1 INTRODUCTION

Trust is an abstract, multi-faceted, and subjective concept [Liu and Loper 2018]. Trust has been investigated in multiple disciplines in addition to computer science, ranging from business, philosophy, social science. Researchers from different domains agree with the fundamental definition of trust, i.e., trust describes an anticipation/trustworthiness level of an individual (as a human being or an intelligence machine). Trust is often derived from certain feedback ratings through trust aggregation. For example, Gambetta [1998] presented trust as the subjective probability in social science that a trustor anticipated a trustee to execute an action beneficial to her/him. Lahno [1999] introduced the philosophy of distrust as the betrayal of moral behavior. In economics, trust is reflected by the decision to maximize the trustor's interest by trading off between the potential risks and the possible utility gains [Cho et al. 2015]. In computer science, many research branches have adopted trust to mitigate various threats and risks through tracking and leveraging historical interaction experiences in open computing systems and networks. Trust is regarded as an essential pillar for our digital economy and our cyber infrastructure [Liu and Loper 2018].

Trust management refers to managing trust in a computing system, including defining trust, identifying the elements that establish trust, and mechanisms for trust computation, trust propagation, trust aggregation, trust data storage, as well as the usage models of trust and trust enhanced service provisioning. One can provide the above functionalities using a centralized computing architecture or a decentralized computing architecture or a hybrid of centralized and decentralized computing architectures, which allows certain trust functionality to be implemented and supported using distributed computing platforms and distributed computation algorithms. **Decentralized trust management** refers to managing trust in either fully decentralized computing systems or a hybrid of centralized and decentralized computing systems.

Over the past decade, trust management has penetrated diverse collaborative networked computing systems, ranging from peer to peer and eCommerce, social networks and online community, cloud and edge computing, mobile ad hoc networks and wireless sensor networks, to crowdsourcing, multi-agent, and Internet of things (IoTs) [Liu et al. 2016].

Peer-to-peer Trust. Trust management in Peer-to-peer systems has been studied for more than two decades [Fan et al. 2017; Jøsang and Ismail 2002; Kamvar et al. 2003; Su et al. 2015; Xiong and Liu 2004] and surveyed [Jøsang et al. 2007; Suryanarayana and Taylor 2004] in the context of decentralized overlay networks and applications. Suryanarayana and Taylor [2004] compared trust metrics from trust attributes and discovery mechanism. Jøsang e al. [2007] summarized a number of main concerns in establishing and utilizing trust, such as low incentive for providing ratings, biases toward positive feedback, colluding participants, unfair feedback ratings from mischievous participants, changing identities, and so on.

Multi-Agent Trust. Trust management in a multi-agent system (MAS) utilizes trust to improve the collaboration among multiple autonomous agents in accomplishing a task. Balaji and Srinivasan [2010] defined the trust of an agent in terms of autonomy, inferential capability, responsiveness, and social behavior. Granatyr et al. [2015] reviewed trust models for MAS by analyzing a set of trust dimensions, such as trust semantics, trust preference, delegation, risk measure, incentive feedback, initial trust, open environment, hard security threats, and requirements, and they identified the linking between trust dimensions and types of interactions, such as coalition, argumentation, negotiation, and recommendation. Pinyol and Sabater-Mir [2013] reviewed trust metrics in terms of trust cognition, procedure, and generality. Yu et al. [2013] reviewed existing trust metrics from a game theoretic perspective. Braga et al. [2018] unveiled some characteristics of trust, such as the usage of multiple input sources, cheating assumptions, provision of procedural and cognition concepts.

Social Network Trust. Trust management in social networks and online communities has been an active research in the past decade [Caverlee et al. 2010; Jiang et al. 2015; Wang and Wu 2011]. Caverlee et al. [2010] proposed social incentives with personalized similarity to improve the aggregation of reputation trust in a large-scale social network in which participants often did not know each other a priori. Sherchan et al. [2013] surveyed several critical attributes of social trust, such as dynamic, propagative, non-transitive properties, interaction behaviors, and historical experiences. Jiang et al. [2016a] reviewed the graph-based trust evaluation for online social networks (OSNs) in two broad categories: graph simplification-based approaches and graph analogy-based approaches. Jiang et al. [2015] also proposed another work to focus on one-hop recommender selection problems in OSNs, e.g., selecting all/a fixed number of/a fixed proportion of/top m qualified neighbors.

Mobil and Wireless Ad Hoc Network Trust. Trust is a popular mechanism for secure routing in mobile and wireless ad hoc networks (MANETs). Zhang et al. [2010] incorporated trust metrics into the routing protocol in wireless ad hoc networks, and provided a theoretical analysis in the perspectives of correctness, optimality and inter-operativity. Movahedi et al. [2016] reviewed several trust frameworks to tackle the bad-mouthing attack and double-face attack. Several survey articles [Agrawal and Verma 2016; Cho et al. 2011; Govindan and Mohapatra 2012; Tangade and Manvi 2013] reviewed trust metrics for MANETs and presented a more comprehensive categorization of potential attacks, such as routing loop attack, wormhole attack, blackhole attack, grayhole attack, DoS attack, on-off attack, package modification/insertion, incomplete information, selective misbehaving attack, conflicting behavior attack, and so on. These attacks enlarge our horizon on threats and vulnerability risks, and provide the basis for the verification of trust management.

Cloud Computing Trust. Noor et al. [2013] classified trust management in cloud computing into four categories with respect to the roles of service requester and service provider: (i) policy; (ii) recommendation; (iii) reputation; and (iv) prediction. Ahmed et al. [2019] presented a trust evaluation survey for the cross-cloud federation, namely, a federation comprised of unknown cloud service providers with heterogeneous infrastructures sharing resource for a limited period. It argued the overall requirements for trust evaluation should include the special requirement from cross-cloud federation, encompassing the architecture and operational principles of federation.

Trust and Cryptography. Kerrache et al. [2016] proposed an adversary-oriented survey on trust and cryptography for vehicular network in terms of security communication, safety application, and infotainment application. The security communication mainly contained certificate replication attack, eavesdropping attack and vehicle/driver privacy attack. The safety application primarily included denial of service (DoS), jamming attack, coalition and platooning attack and betrayal attack. The infotainment application mainly involved replayed/altered/injected message attack and illusion attack, in addition to the common attacks, such as masquerading attack and impersonation attack, Sybil attack, and GPS position faking attack, timing attack, and blackhole/gray-hole attacks.

Trust from Multi-disciplinary Research. Trust has been a common theme from a multi-disciplinary perspective. Cho et al. [2015] surveyed composite trust through deriving trust factors from communication, information, society and cognition, and discussed trust in a comprehensive extent, covering artificial intelligence, human computer interaction, data fusion, human-machine fusion, computer networking and network security, data mining, and automation.

Contributions and Scope. Comparing with existing surveys on trust management in different subject areas above, our article presents three unique contributions: (1) We provide an in-depth characterization of the inherent vulnerabilities and robustness of existing trust metrics through multi-dimensional analysis and extensive experiments, in addition to the root-cause investigation. (2) We formalize the attack cost and the adverse effect using six representative threat models for

risk analysis and trust robustness evaluation with comprehensive experimental verification. (3) By taking into account direct trust aggregation and various trust propagation kernels, we summarize the existing trust metrics into six classifications and evaluate their robustness and adaptability against the six threat models.

In short, this survey focuses on integrating threat models with trust management and provides an in-depth study of fundamental factors for trust establishment, trust propagation, and trust utility in the presence of six categories of common risks and threats. It can serve as a guideline for research and development of next generation trust aggregation algorithms and assist human or intelligence machines to leverage trust management for making decisions in the anticipation of various risk factors and mischievous threats.

First, we introduce how to establish direct trust among individual participants in an interactive network. We review different methods to derive indirect trust from direct trust information, and introduce common reference model for trust establishment, such as honest/dishonest rating, non-creditable rating, feedback credibility-weighted (FCW) direct trust, uniformly distributed trust propagation and threshold-controlled trust propagation kernels.

Second, we categorize common threats and risks emerged in trust management for diverse interactive networks into six types of threat models to characterize six differential types of mischievous adversaries. We quantitatively infer the adverse effect and attack cost for each threat model with experimental analysis and demonstration.

Third, we provide an in-depth analytical comparison of the state of the art trust research from two core-components of decentralized trust management systems: direct trust aggregation and trust propagation kernel design. We study the inherent vulnerabilities of existing trust metrics and evaluate their attack resilience in the context of the six threat models. In addition, we summarize existing trust metrics into six categories based on different trust propagation kernels and direct trust aggregation fashions.

Organization. The rest of this survey article is organized as follows. We first describe and compare the state of the art research in trust management from direct trust aggregation and indirect trust aggregation in Sections 2 and 3, respectively. Then, we describe and categorize attacks and risks into six threat models and present quantitative analysis of the adverse effect and attack cost for the six threat models in Section 4. We compare different trust aggregation kernels and study the root-causes of their vulnerabilities in the presence of the six threat models in Section 5 and provide design principle of decentralized trust management in Section 6. We describe trust applications in edge computing, blockchain, and trust data storage in Section 7, and we conclude this survey in Section 8.

2 DIRECT TRUST ESTABLISHMENT

2.1 Transactions and Interactions in Distributed Open Systems

We can broadly classify distributed networked systems into four categories: (1) distributed clients and centralized servers; (2) distributed clients and distributed servers, and no direct communications between distributed clients; (3) the hybrid, supporting client-server communication in both peer to peer among distributed clients in addition to those from clients to centralized severs or distributed servers; and (4) the peer-to-peer decentralized networked system, where no centralized servers are supported and all client-server communications are done among distributed nodes that serve dual roles of client and server. Web service provisioning from enterprises, such as Amazon, Uber, Airbnb, would belong to the distributed networked systems of type (1) or type (2), and the systems of type (1) and type (2) offer centralized services to a large population of distributed clients. Skype and WeChat are examples of type (3) and the systems of type (3) provide both

centralized servers and decentralized servers as their service provisioning platforms. Bitcoin and Tor are representative peer-to-peer systems of type (4) and the systems of type (4) have no centralized management and participants of the type (4) systems form a peer-to-peer overlay network with decentralized routing protocols (e.g., such as neighbor-based broadcast) to reach the rest of the network and share resources with, or provide services to, the other participants in the network.

In this article, we primarily focus on decentralized trust management in distributed networked systems of types (3) and (4). By decentralized trust management, we mean that the majority of the trust management functionalities, such as trust establishment, trust computation, trust aggregation, trust propagation, will be provided using a decentralized computing architecture. For instance, a participant in an open network can issue a query service to the network as a client to search for a resource, some other participants in the network may respond and provide the resource as the server. Considering the dynamics of open networks, participants may join or depart the network randomly, and the network structure is continuously changing over time. Thus, different requests may require different sets of participants to work together for effective service provisioning. Assuming each request is being served by one sever even when multiple participants may be able to provide the same service. We refer to each service interaction with the completion of a service request as a transaction query between a pair of participants, one as a client and the other as a server. Once one transaction is accomplished, the client participant may give its feedback rating on the server participant with respect to the quality of the transaction query Q. Within open networks, each participant can be a client or a server to a Q and also can be a feedback rater if it is a client participant for Q or a feedback rating receiver if it is a server participant.

Trust management in an open system can also be categorized into three categories based on the three types of interaction patterns: Human to Human (H2H), Human to Machine (H2M), or Machine to Machine (M2M).

Human-to-Human (H2H). H2H trust represents the trustworthy relationship among humans in a physical or virtual community, in which individuals employ a computer-assisted networked system to establish interaction among or with friends and families, as well as unknown individuals in social, physical or virtual community. The direct trust for a pair of participants in such a community reflects the actual interactions driven by common social or business interest, direct or indirect friendship through shared background or experiences, and so on. It heavily reflects the human attributes, e.g., emotion, intimacy and mutual reciprocity [Granovetter 1973]. H2H trust management illuminates the complex trustworthiness relationships in a variety of disciplines, including anthropology [Sherchan et al. 2013], sociology [Golbeck 2005, 2006a, 2006b], economics [Xiong and Liu 2003; Zhang et al. 2015], social psychology [Cook et al. 2005; Rotter 1967] and organizational studies [Jackson 1999].

Human-to-machine (H2M). H2M trust describes the trustworthiness between humans and machine hosted services in a computer-aided networked system. H2M trust management can be regarded as an interface between human consumers and machine-supported services, which, on one hand, assists consumers to select trustworthy services and, on the other hand, prevents consumers from getting untrusted services or blocks attacks to the service hosting system. Noor et al. [2016] proposed to aggregate the consumers' feedback ratings on cloud services. Habib et al. [2011] proposed a trust mechanism to guarantee clients to receive only trustworthy cloud services. Hwang and Li [2010] built a trust-based cloud service architecture to protect both cloud providers and consumers. Li et al. [2010] proposed a trust-aware service brokering system to assist the selection of trustworthy cloud services.

Machine-to-machine (M2M). M2M trust management refers to the mechanisms that measure and manage the trustworthiness of the functionalities performed by the participants of the open networked system, which are typically software agents hosted on the network nodes (virtual or

physical machines). When the pair of communicating nodes can accomplish the intended transaction with an expected result [Liu et al. 2016], the client machine can provide a good feedback rating for the server machine based on the expected behavior, typically defined by the trust model in the form of trust policies. Walter et al. [2009] proposed a trust-based recommendation approach to assist the selection of well-behaved vehicles on demand. Tan et al. [2016] proposed a trust management scheme to secure the data plane of ad-hoc networks. Nitti et al. [2014] proposed to use trust metrics to assist the establishment of trustworthy IoTs.

In this article, although we focus primarily on decentralized trust management in machine to machine communication scenarios, many of the design principles and trust management algorithms such as trust aggregation and trust propagation kernels can be easily adapted to the H2M and H2H trust management systems.

2.2 Direct Trust with Local Trust Aggregation

Generally, the feedback rating is positive if the transaction query is satisfied, or negative if unsatisfied. Nevertheless, the emergence of strategically mischievous participants (SMPs) breaks this routine feedback pattern, i.e., the SMPs, on the one hand, provide high-quality transaction queries to get honest (positive) ratings from other service receivers as server participants but, on the other hand, give dishonest (negative) feedback ratings to other service providers as client participants ignoring whatever the transaction queries are satisfied or unsatisfied [Fan et al. 2012, 2017; Kamvar et al. 2003; Su et al. 2013]. In addition, the query transaction may fail with non-response, or delivering faulty/low-quality results due to unintended reasons, such as network bandwidth jitter, cooling-induced cloud server downtime, and so on. Upon the above analysis, we know that the mischievous behavior can be studied from (i) two-facet intended manners, i.e., service-based misbehaved manipulation, rating-based misbehaved manipulation; and (ii) one unintended manner, i.e., system/network reliability factors-induced natural failure.

Service-based Feedback Rating. Normally, we refer the feedback that a service provider receives a positive/negative rating while providing an authentic/inauthentic service as honest rating. Inversely, we refer the feedback that a service provider receives a negative/positive rating while providing an authentic/inauthentic service as dishonest rating. Each individual can alternately play the two roles during the interactions, i.e., service provider (a.k.a. server participant) and service consumer (a.k.a. client participant). The vicious participants can be categorized as independently mischievous, collusively mischievous, randomly mischievous, occasionally mischievous and persistently mischievous [Fan et al. 2017]. In fact, the rating is reflected by feedback rater being honest or dishonest, being independently or collusively dishonest, being randomly or occasionally dishonest or persistently dishonest, but all about the rater's perception on the query transaction quality.

The different categories of mischievous participants may have some overlap in terms of malicious manipulation, such as they all provide inauthentic services and dishonest feedback ratings, but for different threat models the malicious behaviors may be naively malicious or strategically malicious in serving or rating or both. Prior to establishing direct trust, we first give some basic definitions.

Definition 2.1 (Honest Rating). The rating is strictly subject to the query transaction quality, i.e., positive rating for authentic service and negative rating for inauthentic service. It can be trusted by the network as a local trust metric for the feedback receiver.

Definition 2.2 (Dishonest Rating). The rating is alternatively subject to the transaction target rather than the truthful query transaction quality, i.e., positive rating for colluding participant and

negative rating for routine participant. It ought to be weighted by feedback credibility prior to being trusted by the network.

Usually, 5% dishonest ratings are allowed to reflect the randomly or occasionally dishonest behavior of an honest rater [Fan et al. 2017; Kamvar et al. 2003] due to some unintended reasons. The direct trust from a participant p_i to another participant p_j based on one-time transaction can be demonstrated using binary or multiscale rating. For example, the pioneering trust metric EigenTrust [Kamvar et al. 2003] defined the direct trust for a pair of transacted participants using binary rating $\{-1, +1\}$, $tr(p_i, p_j) = -1$ denoted a negative rating from p_i to p_j, and $tr(p_i, p_j) = +1$ represented a positive rating. The heritage EigenTrust^{++} [Fan et al. 2012] and GroupTrust [Fan et al. 2017], both employed this kind of binary rating. Differently, ServiceTrust [Su et al. 2013] and ServiceTrust^{++} [Su et al. 2015] utilized the multiscale rating $\{-1, 0, 1, 2, 3, 4, 5\}$ indicating bad, no-rating, neutral, fair, good, very good and excellent query transaction, respectively.

For SMPs, one smart way to subvert the system is to firstly collect positive ratings to yield high trust through providing authentic services, then utilize the advantage of gained trust to participate the query transactions to provide inauthentic services. Hence, a reliable trust metric ought to take the historical feedback information into account, i.e., recent feedback ratings should be more valuable and historical feedback ratings should be less valuable. The studies [Li and Wang 2008; Li et al. 2009; Wang and Li 2011; Zacharia and Maes 2000] also confirmed the ratings in a recent time period weighted more than the former emerged ratings. In this way, once SMPs are found to change their transactional behaviors from honestly providing authentic services to dishonestly offering inauthentic services, their trust would be degraded shortly using weight parameter even they possessed high trust already. For instance, Li et al. [2011] used the weight decimal $[0, 1]$ manner to distinguish the different-time feedback ratings for personal rating and indirect recommendation. For a given period of time interval $[t_1, t_n]$, the direct trust level from p_i to p_j can be calculated as

$$tr_h(p_i, p_j)^{(t_i)} = w_{t_i} \cdot tr(p_i, p_j)^{(t_i)} \quad t_i \in [t_1, t_n], \tag{1}$$

where w_{t_i} set as $a^{t_n - t_i}$ $(0 < a \leq 1)$ is the weight of the rating $tr(p_i, p_j)^{(t_i)}$ at time t_i.

Rating-based Feedback Rating. The SMPs can indeed earn high trust by aggregating honest positive ratings from good participants via serving authentic resources and dishonest positive ratings from their colluders. This violates the initial aim of trust metrics at degrading the trust levels of mischievous participants. Thus only utilizing honest rating is inadequate for the SMPs, we need further explore creditable rating to amend the deficiency.

Definition 2.3 (Creditable Rating). The rating for a single feedback rater or pairwise feedback rating score is integrated by feedback credibility factor.

Oppositely, non-creditable rating can be defined as follows:

Definition 2.4 (Non-creditable Rating). The rating for a single feedback rater or pairwise feedback rating score is always negative without referring to any creditable factor.

Upon Definition 2.3, we can further study the creditable ratings from two levels, namely, feedback rater and feedback rating score.

Definition 2.5 (Feedback Rater Level Credibility). The interactive system endows each feedback rater a credibility weight for local trust aggregation.

Definition 2.6 (Feedback Rating Score Level Credibility). The interactive system endows each feedback rating score over a pair of transacted participants a credibility weight for local trust aggregation.

Xiong and Liu [2004] had proposed two kinds of credibility measure fashions from the standpoints of feedback rater level and feedback rating score level. The former (PeerTrustTVM) set self-trust as credibility weight, which interpreted the feedback rating of a trustworthy participant possessed more credibility than that of an untrustworthy participant:

$$Cr_{p_i} = \frac{T(p_i)}{\sum_{p_m=1}^{|tr(p_j)|} T(p_m)} \quad p_i \in tr(p_j), \tag{2}$$

where $tr(p_j)$ represented the set of participants that had transactions with p_j, $T(p_i)$ denoted the trust value of p_i. The latter (PeerTrustPSM) employed feedback similarity as credibility factor, the mischievous participants had low feedback similarity with good participants due to greatly different ratings to commonly transacted participants:

$$Cr_{p_i p_k} = \frac{sim(p_i, p_k)}{\sum_{p_m=1}^{|tr(p_j)|} sim(p_i, p_m)} \quad p_i, p_k \in tr(p_j), \tag{3}$$

$$sim(p_v, p_w) = 1 - \left(\frac{\sum_{p_x \in comn(p_v, p_w)} \left(\sum_{|tr(p_v, p_x)|} \frac{tr(p_v, p_x)}{|tr(p_v, p_x)|} - \sum_{|tr(p_w, p_x)|} \frac{tr(p_w, p_x)}{|tr(p_w, p_x)|} \right)^2}{|comn(p_v, p_w)|} \right)^{\frac{1}{2}}, \tag{4}$$

where $sim(p_v, p_w)$ was the feedback similarity through inferring the standard deviation of feedback ratings to the commonly rated participants $comn(p_v, p_w)$. Similarly, GroupTrust [Fan et al. 2017] used exponential function to define feedback rating score level credibility:

$$Cr'_{p_i p_j} = exp\left\{1 - \frac{1}{sim(p_i, p_j)}\right\}. \tag{5}$$

Raw Local Trust Aggregation. The direct trust over each pair of transacted participants can be calculated via local trust aggregation. At present, the commonly used aggregating fashions can be roughly classified into two manners: (i) transaction success ratio; and (ii) beta function probability expectation. Intuitively, the transaction success ratio-employed direct trust from p_i to p_j can be defined as

$$s_{p_i p_j} = \begin{cases} \frac{\delta_{p_i p_j}}{\delta_{p_i p_j} + \sigma_{p_i p_j} + 1} \frac{\sigma_{p_i p_j}}{\delta_{p_i p_j} + \sigma_{p_i p_j} + 1} \leq \theta \\ \frac{1}{2} \quad otherwise \end{cases}, \tag{6}$$

where θ implies good participants misbehave in a tiny probability due to system reliability factors-induced natural failure, usually set as 5% [Fan et al. 2012; Kamvar et al. 2003; Su et al. 2013]. $\delta_{p_i p_j}$ denotes the number of successful transactions between p_i and p_j, $\sigma_{p_i p_j}$ is the number of unsuccessful transactions.

The beta probability density functions [Jøsang and Ismail 2002; Klos and Alkemade 2005; Walter et al. 2009] can be expressed via gamma function:

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \rho^{\alpha-1}(1-\rho)^{\beta-1}, \tag{7}$$

where $\alpha, \beta > 0$, and $0 \leq \rho \leq 1$, $\rho \neq 0$ if $\alpha < 1$ and $\rho \neq 1$ if $\beta < 1$. At present, scholars straightly define the direct trust as the probability expectation of beta distribution:

$$s^{\beta}_{p_i p_j} = \alpha/(\alpha + \beta) = (\delta_{p_i p_j} + 1)/(\delta_{p_i p_j} + \sigma_{p_i p_j} + 2), \tag{8}$$

where $\alpha = \delta_{p_i p_j} + 1$, $\beta = \sigma_{p_i p_j} + 1$. Essentially, this beta function-based direct trust reflects the transaction success ratio as well. For easy understanding, we call both $s_{p_i p_j}$ and $s^{\beta}_{p_i p_j}$ as raw direct trust

with different local trust aggregation fashions given they only adopt positive and negative ratings to produce direct trust for a pair of transacted participants.

Without a doubt, the local trust aggregation can pull-in some more crucial impact factors, e.g., the aforementioned history feedback factor and feedback credibility. Naturally, referring to historical feedback factor, the local trust aggregation $s^h_{p_ip_j}$ can be defined as

$$s^h_{p_ip_j} = \frac{\sum_{i=1}^{n} tr_h(p_i,p_j)^{(t_i)}}{\sum_{i=1}^{n} w_{t_i}} \quad t_i \in [t_1, t_n]. \tag{9}$$

Furthermore, referring to feedback credibility factor, we can define the FCW direct trust.

Definition 2.7 (Feedback Credibility-weighted Direct Trust). The direct trust over each pair of transacted participants is yielded through integrating creditable trust factors from the viewpoint of a single feedback rater or pairwise feedback rating score.

Accordingly, the feedback rater level credibility-based direct trust $s'^{Cr}_{p_ip_j}$ is defined as

$$s'^{Cr}_{p_ip_j} = Cr_{p_i} \cdot s_{p_ip_j} \quad p_i \in tr(p_j). \tag{10}$$

The pairwise feedback rating score level credibility-based direct trust $s^{Cr}_{p_ip_j}$ ($s^{Cr'}_{p_ip_j}$) is defined using feedback similarity among the two transacted participants [Fan et al. 2017]:

$$s^{Cr'}_{p_ip_j} = Cr'_{p_ip_j} \cdot s_{p_ip_j} \quad p_i, p_j \in tr(p_k), \tag{11}$$

or using feedback similarity of reference third-party participants [Xiong and Liu 2004]:

$$s^{Cr}_{p_ip_j} = Cr_{p_ip_k} \cdot s_{p_kp_j} \quad p_k, p_i \in tr(p_j). \tag{12}$$

Trust itself is a complex and subjective concept impacted by multiple factors with respect to the diversities of misbehaved participants, thereby, for mischievous participants especially SMPs, it is hard to constrain them from gaining high direct trust from honestly transacted participants. Therefore, an effective trust metric need take into account these impact factors to infer a multi-perspective and rational direct trust for each two transacted participants.

3 INDIRECT TRUST WITH NETWORK-BASED TRUST AGGREGATION

3.1 Network-Scoped Trust Aggregation

The baseline inference on indirect trust from participants p_u to p_v is to recursively aggregate the third-party participant p_k's direct trust placed on p_v within the holistic network, i.e., $\sum_{p_k} s_{p_up_k} \cdot s_{p_kp_v}$ or $\sum_{p_k} s^{Cr}_{p_up_k} \cdot s^{Cr}_{p_kp_v}$. For an interactive network with N_{net} participants, the network-scoped trust can be derived using the power iteration of adjacent matrix in which each element stands for the direct trust value over each pair of participants:

$$T_G^{(k+1)} = M^T \cdot T_G^{(k)}, \tag{13}$$

where $T_G^{(k+1)}$ denotes the (k+1)th iteration trust vector of N_{net} participants, M is the normalized direct trust matrix: $m_{p_up_v} = s_{p_up_v} / \sum_{p_m} s_{p_up_m}$, if $\sum_{p_m} s_{p_up_m} \neq 0$, otherwise $m_{p_up_v} = 0$. This iteration operation, in fact, is also a trust propagation/diffusion process hop by hop, k controls the propagating scope of trust. Obviously, each participant's trust can be propagated to the whole network with certain iteration rounds. Inversely, each participant can also receive trust from the whole network. Thus, we define the global trust as follows.

Definition 3.1 (Global Trust). We define the trust score computed by indirect trust aggregation over the entire network through the network topology as the global trust from one participant p_u to another participant p_v, provided that p_v is reachable from p_u by network traversal. The global

trust value can be viewed as the comprehensive confidence that the entire network as a community places on the participant p_v via the view of p_u.

EigenTrust [Kamvar et al. 2003] is the first to introduce the use of the pre-trusted nodes as the authority participants to address the "cold start" problem. Taking into account the pre-trusted participants, the authors defined the eigenvector-based global trust as

$$T_G^{(k+1)} = (1-\varepsilon) \cdot M^T \cdot T_G^{(k)} + \varepsilon \cdot \overrightarrow{P}, \tag{14}$$

where ε denoted the probability a stranger or newcomer would like to trust the system-generated pre-trusted participants P, $p_{p_j} = 1/|P|$ if participant $p_j \in P$, otherwise $p_{p_j} = 0$.

3.2 Trust Propagation Kernel

From Equation (14), we know that an individual's global trust is aggregated through asking the other participants' feedback ratings placed on this individual. We define this kind of trust propagation kernel as uniformly distributed trust propagation.

Definition 3.2 (Uniformly Distributed Trust Propagation, UDTP). For each participant, it propagates self-global trust to all the neighboring participants in the light of direct trust values placed on the neighboring participants.

Although this UDTP kernel is the core to aggregate global trust at present, it confronts rigorous inherent vulnerabilities. It is this UDTP kernel that enhances the global trust scores of SMPs, i.e., they can repeatedly gain high trust deriving from authentic-service provision activities. If no mischievous participant exists, then this UDTP kernel can yield correct and rational trust level for each participant. To address the vulnerability, a threshold-controlled trust propagation kernel was proposed in Fan et al. [2017] and Su et al. [2015].

Definition 3.3 (Threshold-Controlled Trust Propagation, TCTP). For a participant, the decision whether it can propagate trust to its neighboring participants strictly depends on the system-inferred critical threshold.

Generally, TCTP kernel adopts the direct trust to compare with the system-inferred critical threshold [Fan et al. 2017; Su et al. 2015], if larger, trust propagation is permitted; otherwise, trust propagation is blocked. Appropriately setting on TCTP can validly control trust propagation; otherwise, it puts TCTP at a disadvantage; i.e., if the threshold is too low, then it cannot block SMPs receiving trust propagation from good participants, and if the threshold is too high, then it might block trust propagation among good participants. An attack resilient trust metric should have the capability to differentially propagate trust among good and different categories of mischievous participants rather than simply utilizing UDTP kernel.

4 THREAT MODELS AND ADVERSE EFFECTS

4.1 Reference Threat Models

Massive simple/strategic threats and risks have been penetrating open networked systems, such as bad-mouthing [Sun et al. 2006], self-promoting [Hoffman et al. 2009], ballot stuffing behavior [He et al. 2012; Hu et al. 2017], collusively malicious [Fan et al. 2017; Jiang et al. 2016b; Sun et al. 2006], on-off attack [Chae et al. 2015; Sun et al. 2006], Sybil [Liu et al. 2015; Wang et al. 2015], spy [Fan et al. 2017; Kamvar et al. 2003], black-hole and grey-hole attacks [Kerrache et al. 2016], and so on. For the sake of easy understanding, according to Fan et al. [2017], Kamvar et al. [2003] and Su et al. [2015], we summarize these threats and risks as several threat models referring to attack policies and characteristics. Besides the four representative threat models, we also in advance propose another two more-sophisticated threat models to support our deep arguments.

Definition 4.1 (Threat Model A-Independently Mischievous). All mischievous participants perform bad services and dishonest feedbacks independently. Concretely, they provide inauthentic services when selected as transaction service providers (server participants) and they always offer non-creditable ratings to other transacted participants ignoring whether the received services are authentic or inauthentic as feedback raters. The mischievous participants in this category always receive bad ratings from good participants.

Definition 4.2 (Threat Model B-Collectively Mischievous). All mischievous participants are organized in a chain to collude with each other. They always give inauthentic services as server participants, and they always provide dishonest ratings as feedback raters, i.e., giving dishonest (negative) ratings to good participants but dishonest (positive) ratings to other colluding participants over the chain. This Threat Model B adds colluding effect on feedback ratings compared to the Threat Model A.

Definition 4.3 (Threat Model C-Camouflage Collective). All mischievous participants are organized in a chain to collude mutually. They give authentic services in a probability f when selected as server participants, and provide dishonest ratings as feedback raters, i.e., giving dishonest (negative) ratings to good participants but dishonest (positive) ratings to colluding participants in the chain. The Threat Model C adds *a camouflage strategy*: Instead of providing bad services all the time, mischievous participants play camouflage games at a probability f, aiming to cheat the trust system, i.e., the mischievous camouflage participants will receive honest (positive or negative) ratings from some good participants and thus gain relatively higher trust through aggregating positive ratings.

Definition 4.4 (Threat Model D-Group-based Spies). All mischievous participants are divided into two types: Type B acting like the vicious participants in Threat Model A (provide bad services and give dishonest feedbacks) and Type D vicious participants do good services but give dishonest feedbacks. It changes the malicious method of selectively providing good services in Threat Model C to use a subset of mischievous participants to provide good service all the time, but give dishonest feedbacks. It is another malicious strategy intended to cheat the trust system. This threat model adds *another malicious strategy*: Type-D participants give negative ratings to good participants but positive ratings to all Type-B participants.

Definition 4.5 (Threat Model E-Camouflage Collective with Honest Rating). Like Threat Model C, all mischievous participants are organized in a chain to collude each other, but they play camouflage game at both service provisioning and feedback rating. They give authentic services in a probability f when selected as server participants and will offer honest ratings with probability η as feedback raters. This is a change to Threat Model C with the goal of providing a *third malicious strategy* to cheat the trust system: By playing camouflage game also as feedback raters, it allows the good participants to receive positive ratings from camouflage participants at probability η, making it hard for the trust system to detect and identify mischievous participants.

Definition 4.6 (Threat Model F-Group-based Spies with Honest Rating). All mischievous participants are composed of Type-D participants and Type-B participants. When selected as server participants, Type-D participants always give authentic services and Type-B participants always give inauthentic services. However, as feedback raters, the mischievous participants offer honest ratings with probability γ, that is to say the good participants can receive positive ratings from mischievous spies with probability γ. This is a change to Threat Model D with the goal of providing the *fourth malicious strategy* to cheat the trust system: The good participants can receive positive ratings from spy participants at probability γ, making it hard for the trust system to detect and identify mischievous participants.

Threat Models A-D have been used by some existing trust metrics, e.g., EigenTrust [Kamvar et al. 2003], ServiceTrust [Su et al. 2013], and so on. Threat Models E and F are introduced in this survey to support the most strategically malicious attacks. Threat Models A and B are simple, Threat Models C and D are somewhat more sophisticated, and Threat Models E and F are most strategically malicious, as shown in Table 3.

4.2 Adverse Effect and Cost of Attacks

4.2.1 Attack Cost and Attack Success Ratio. To formally infer adverse effect and attack cost for each threat model, we first give the definition of attack cost with respect to interactive properties.

Definition 4.7 (Attack Cost). Attack cost is comprehensively reflected by the price the mischievous participants need to pay for launching an attack successfully in terms of the amounts of mischievous participants and dishonest feedback ratings, the numbers of authentic services and honest ratings offered by SMPs.

For each participant, its global trust, in fact, contains both trustworthiness and untrustworthiness factors, this is because the direst trust over each pair of transacted participants as the base of global trust calculation, is inferred using both positive and negative ratings. To interpret adverse effect, we need separately derive the inherent trust ingredient contribution and distrust ingredient contribution.

Definition 4.8 (Trust Ingredient). For a participant p_i, its trust ingredient $T_{ti}(p_i)$ is aggregated by the honest feedback ratings it received:

$$T_{ti}(p_i) = \sum_{tr(p_w,p_i)\in R_H} m_{p_w p_i} \cdot T(p_w), \tag{15}$$

where R_H is the set of positive feedback ratings offered by good participants or SMPs.

Definition 4.9 (Distrust Ingredient). For a participant p_i, its distrust ingredient $T_{di}(p_i)$ is aggregated by the dishonest and non-creditable feedback ratings it received:

$$T_{di}(p_i) = \sum_{tr(p_u,p_i)\in R_{dH}\cup R_{nH}} m_{p_u p_i} \cdot T(p_u), \tag{16}$$

where R_{dH} denotes the set of negative ratings offered by collectively mischievous, camouflage and spy participants in Threat Models B-F, R_{nH} represents the set of non-creditable ratings offered by independently mischievous participants in Threat Model A.

Next, we utilize trust ingredient and distrust ingredient to define attack success ratio.

Definition 4.10 (Attack Success Ratio). Given an attack target participant p_i, once the distrust ingredient it received is larger than the trust ingredient, that is to say this participant is successfully attacked. Accordingly, the attack success ratio $As(p_i)$ is defined as

$$As(p_i) = T_{di}(p_i)/T_{ti}(p_i). \tag{17}$$

Obviously, if the adversary participants want to launch an attack successfully on a target p_i, they must yield larger distrust ingredient compared to trust ingredient, i.e., $As(p_i) > 1$. For clear description, we show the related notations and presentations in Table 1.

Table 1. Notations and Presentations

Notation	Presentation
N_{net}	number of system participants (network size)
R_{dH}	set of dishonest ratings offered by collective, camouflage and spy attackers
R_{nH}	set of non-creditable ratings offered by independent attackers
R_H	set of honest rating offered by good participants and SMPs
$T_G(p_i)$	global trust score of participant p_i
$T_{ti}(p_i)$	trust ingredient of participant p_i
$T_{di}(p_i)$	distrust ingredient of participant p_i
T_{net}^G	network level average trust of good participants
T_{net}^M	network level average trust of mischievous participants
N_{dH}	number of participants that offer dishonest ratings
N_{nH}	number of participants that offer non-creditable ratings
N_H	number of participants that offer honest ratings
N_C	number of camouflage participants launched by Threat Models C and E
N_D	number of Type-D participants launched by Threat Models D and F
N_B	number of Type-B participants launched by Threat Models D and F

4.2.2 Adverse Effect and Attack Cost of Threat Model A. Based on Equation (17), for independently mischievous participants, if they attack a target participant p_i successfully, the cost must meet the following condition:

$$\sum_{tr(p_u,p_i)\in R_{nH}} m_{p_u p_i} T_G(p_u) / \sum_{tr(p_w,p_i)\in R_H} m_{p_w p_i} T_G(p_w) > 1. \tag{18}$$

We utilize raw direct trust to replace normalized direct trust $m_{p_u p_i}$, and beta function-based expectation to interpret direct trust. Thus, we can transform Equation (18) into

$$\sum_{tr(p_u,p_i)\in R_{nH}} \frac{1}{|tr(p_u,p_i)|+2} \cdot T_G(p_u) > \sum_{tr(p_w,p_i)\in R_H} \frac{|tr(p_w,p_i)|+1}{|tr(p_w,p_i)|+2} \cdot T_G(p_w). \tag{19}$$

As transaction increases, the beta function-based direct trust over a pair of good participants p_w and p_i will enlarge, we can see that via Equation (19). Inversely, the direct trust over a pair of good and mischievous participants p_u and p_i will decline. Thereby, the appropriate time to launch an attack is at the beginning period, otherwise the mischievous participants would pay more cost. Thus, we calculate the adverse effect and attack cost when the transaction is performed only one-time, and suppose the feedback employs binary rating. In addition, we assume the number of independently mischievous participants is N_{nH} with the network level average trust T_{net}^M, the number of good participants is N_H with the network level average trust T_{net}^G. Thus, we can rewrite the condition Equation (19) as

$$N_{nH} \cdot \frac{1}{1+2} \cdot T_{net}^M > N_H \cdot \frac{1+1}{1+2} \cdot T_{net}^G. \tag{20}$$

Accordingly, the number of mischievous participants is $N_{nH} = (\lfloor 2N_H \times T_{net}^G / T_{net}^M \rfloor + 1)$. Given the independency of mischievous participants, each needs to launch at least one-time non-creditable rating to the target participant, thus the total non-creditable ratings are at least $N_{nR} = (\lfloor 2N_H \times T_{net}^G / T_{net}^M \rfloor + 1)$.

4.2.3 Adverse Effect and Attack Cost of Threat Model B. The mischievous participants organize a chain, and each mischievous participant in the chain would offer a high dishonest rating (1.0) to its partner. Nevertheless, since these mischievous participants cannot provide authentic services, they hardly gain positive ratings from good participants, this implies the chain-reinforced function, in fact, loses the trust transitivity effect. Thus, we have the following attack success condition:

$$\sum_{tr(p_u,p_i)\in R_{dH}} m_{p_u p_i} T_G(p_u) \Big/ \sum_{tr(p_w,p_i)\in R_H} m_{p_w p_i} T_G(p_w) > 1. \tag{21}$$

Accordingly, we have that

$$N_{dH} \cdot \frac{1}{1+2} \cdot T^M_{net} > N_H \cdot \frac{1+1}{1+2} \cdot T^G_{net}. \tag{22}$$

Thereby, the number of mischievous participants is $N_{dH} = (\lfloor 2N_H \times T^G_{net}/T^M_{net} \rfloor + 1)$. Besides dishonest (negative) ratings to target participant, the mischievous participants need give their chain-based partners dishonest (positive) ratings, the total dishonest ratings are $N_{dR} = 2 \cdot (\lfloor 2N_H \times T^G_{net}/T^M_{net} \rfloor + 1)$.

4.2.4 Adverse Effect and Attack Cost of Threat Model C. The camouflage participants not only form the reinforced trust-transitivity chain, but they can also provide authentic services with probability f to gain positive ratings from good participants. We assume the amount of authentic services provided by one camouflage participant is I_H and simultaneously it receives I_H positive ratings from good participants. Thus, we aggregate trust ingredient through received positive ratings:

$$\begin{aligned} T_{ti}(p_c) &= \sum_{tr(p_w,p_c)\in R_H} \frac{|tr(p_w,p_c)|+1}{|tr(p_w,p_c)|+2} \cdot T_G(p_w) \\ &= I_H \cdot \frac{1+1}{1+2} \cdot T^G_{net} \end{aligned}, \tag{23}$$

where $T_{ti}(p_c)$ denotes the trust ingredient of camouflage participant p_c. We assume the number of camouflage participants is N_C, they in return use gained trust ingredient $T_{ti}(p_c)$ as distrust ingredient to attack target participant. Thus, we can replace condition Equation (22) as

$$\begin{aligned} & lN_C \cdot \frac{1}{1+2} \cdot I_H \cdot \frac{1+1}{1+2} \cdot T^G_{net} > N_H \cdot \frac{1+1}{1+2} \cdot T^G_{net}. \\ & N_C > 3 \cdot N_H / I_H \end{aligned} \tag{24}$$

Given the direct trust over the chain is high (1.0), which implies the camouflage participants as a community only need one member to provide authentic services to gain positive ratings, this member in return can propagate its gained trust to its partner along the chain, by analogy, all camouflage participants can get the same trust value through the chain-based direct trust. Therefore, the attack cost includes: (i) the number of camouflage participants is $(\lfloor 3N_H/I_H \rfloor + 1)$; (ii) the amount of authentic services provided by camouflage participants is I_H; (iii) the dishonest ratings to target participant are $(\lfloor 3N_H/I_H \rfloor + 1)$; and (iv) as well as the dishonest ratings to partners over the chain are $(\lfloor 3N_H/I_H \rfloor + 1)$.

4.2.5 Adverse Effect and Attack Cost of Threat Model D. We assume each Type-D participant provides I_H authentic services and receives I_H positive ratings from good participants. According

to Equation (23), we have each Type-D participant's trust ingredient:

$$\begin{aligned} T_{ti}(p_D) &= \sum_{tr(p_w,p_c)\in R_H} \frac{|tr(p_w,p_d)|+1}{|tr(p_w,p_d)|+2} \cdot T_G(p_w) \\ &= I_H \cdot \frac{1+1}{1+2} \cdot T_{net}^G \end{aligned}, \tag{25}$$

where $T_{ti}(p_D)$ denotes the trust ingredient of Type-D participant aggregated through gained positive ratings. Given Type-D participants cooperate with Type-B participants, i.e., Type-D participants proportionally give each Type-B participant direct trust $1/|N_B|$. Thus, each Type-B participant's trust ingredient can be calculated as

$$\begin{aligned} T_{ti}(p_B) &= N_D \cdot \frac{1}{N_B} \cdot T_{ti}(P_D) \\ &= N_D \cdot \frac{1}{N_B} \cdot I_H \cdot \frac{1+1}{1+2} \cdot T_{net}^G \end{aligned}, \tag{26}$$

where N_D and N_B ($\neq$0) stand for the numbers of Type-D and Type-B participants. To achieve attacking target, it must meet the following condition:

$$\begin{gathered} \frac{N_D}{1+2} \cdot T_{ti}(p_D) + \frac{N_B}{1+2} \cdot T_{ti}(p_B) > \frac{2N_H}{1+2} \cdot T_{net}^G \\ N_D \cdot \frac{1}{1+2} \cdot I_H \cdot \frac{1+1}{1+2} \cdot T_{net}^G + N_B \cdot \frac{1}{1+2} \cdot N_D \cdot \frac{1}{N_B} \cdot I_H \cdot \frac{1+1}{1+2} \cdot T_{net}^G > N_H \cdot \frac{1+1}{1+2} \cdot T_{net}^G. \\ N_D > \frac{3N_H}{2I_H} \end{gathered} \tag{27}$$

Therefore, the attack cost includes: (i) Type-D participants' amount is $(\lfloor 3N_H/2I_H \rfloor + 1)$, Type-B participants' amount is N_B; (ii) Type-D participants need to altogether provide good participants $((\lfloor 3N_H/2I_H \rfloor + 1) \cdot I_H)$ authentic services; (iii) offer $(\lfloor 3N_H/2I_H \rfloor + 1) \cdot N_B$ dishonest ratings simultaneously to Type-B participants; and (iv) Type-D and Type-B individuals need offer $((\lfloor 3N_H/2I_H \rfloor + 1) + N_B)$ dishonest ratings to target participant. From Equation (27), we can see that the number of Type-B participants does not affect the adverse effect of Type-D individuals, they just can be viewed as a trust transitivity bridge to receive Type-D participants' trust ingredient to further perform dishonest transactions.

4.2.6 Adverse Effect and Attack Cost of Threat Model E. The camouflage participants not only provide authentic services with probability f to gain positive ratings as feedback receivers but also offer honest ratings with probability η as feedback raters. According to Equation (23), we know the camouflage participants can gain the trust ingredient $T_{ti}(p_c)$ by contributing I_H honest transactions. Since they offer honest ratings with probability η, which means they will propagate trust ingredient to good participants with proportion η, and trust ingredient to the chain-based mischievous partners with probability (1-η). For the sake of explicitly understanding, we identify the camouflage participants in the chain as $p_{c_1}, p_{c_2}, \ldots, p_{c_{Nc}}$. Might as well assume the first camouflage participant has provided I_H authentic services and gained trust ingredient $T_{ti}(p_{c_1})$ already. Thus, with the trust transitivity the ith camouflage participant's trust ingredient can be calculated as

$$T_{ti}(p_{c_i}) = T_{ti}(p_{c_1}) \cdot (1-\eta)^{(i-1)}. \tag{28}$$

To meet the attack success condition, we can rewrite the Equation (24) as

$$\begin{aligned} &\frac{1}{1+2}\cdot T_{ti}(p_{c_1})+\frac{1}{1+2}\cdot T_{ti}(p_{c_2})+\cdots+\frac{1}{1+2}\cdot T_{ti}(p_{c_{N_C}})>N_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}\\ &\frac{1}{1+2}\cdot I_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}\cdot(1+(1-\eta)+\cdots+(1-\eta)^{(N_C-1)})>N_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}\\ &N_C>\log_{(1-\eta)}\left(1-\frac{3N_H\cdot\eta}{I_H}\right)\end{aligned}. \tag{29}$$

Like Threat Model C, only one camouflage participant needs to contribute authentic services, the others can gain trust ingredient through trust transitivity. The attack cost includes: (i) the number of camouflage participants is $(\lfloor\log_{(1-\eta)}(1-3N_H\cdot\eta/I_H)\rfloor+1)$; (ii) the number of authentic services is I_H; (iii) the amount of dishonest ratings given to target participant is $(\lfloor\log_{(1-\eta)}(1-3N_H\cdot\eta/I_H)\rfloor+1)$; (iv) the dishonest ratings given to mischievous partners are $(\lfloor\log_{(1-\eta)}(1-3N_H\cdot\eta/I_H)\rfloor+1)$; and (v) given the amount of dishonest ratings to the mischievous partners and attack target, in addition to the probability η, we can derive the total ratings offered by camouflage participants are $(\lfloor\frac{2(\lfloor\log_{(1-\eta)}(1-3N_H\cdot\eta/I_H)\rfloor+1)}{(1-\eta)}\rfloor+1)$, the number of honest ratings is $(\lfloor\frac{2\eta\cdot(\lfloor\log_{(1-\eta)}(1-3N_H\cdot\eta/I_H)\rfloor+1)}{(1-\eta)}\rfloor+1)$.

4.2.7 Adverse Effect and Attack Cost of Threat Model F. The honest ratings offered by spy participants mainly influence the trust transitivity from Type-D to Type-B participants, i.e., the Type-D participants aim to split the gained trust ingredient from good participants, in return to rate back to the good ones honestly. In this way, the trust ingredient propagated to Type-B participants from Type-D participants declines by probability γ. Taking into account this, we redefine the trust gradient for each Type-B participant as

$$\begin{aligned} T_{ti}(p_B)&=N_D\cdot\frac{1}{N_B}\cdot(1-\gamma)\cdot T_{ti}(P_D)\\ &=N_D\cdot\frac{1}{N_B}\cdot(1-\gamma)\cdot I_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}\end{aligned}. \tag{30}$$

In accordance, we rewrite the attack success condition Equation (27) as

$$\begin{aligned} &N_D\cdot\frac{1}{1+2}\cdot T_{ti}(p_D)+N_B\cdot\frac{1}{1+2}\cdot T_{ti}(p_B)>N_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}\\ &N_D\cdot\frac{1}{1+2}\cdot I_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}+N_B\cdot\frac{1}{1+2}\cdot N_D\cdot\frac{1}{N_B}\cdot(1-\gamma)\cdot I_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}>N_H\cdot\frac{1+1}{1+2}\cdot T^G_{net}.\\ &N_D>\frac{3N_H}{(2-\gamma)\cdot I_H}\end{aligned} \tag{31}$$

Therefore, the attack cost includes: (i) Type-D participants' amount is $(\lfloor\frac{3N_H}{(2-\gamma)\cdot I_H}\rfloor+1)$, Type-B participants' amount is N_B; (ii) Type-D participants need offer $(\lfloor\frac{3N_H}{(2-\gamma)\cdot I_H}\rfloor+1)\cdot I_H$ authentic services; and (iii) give $(\lfloor\frac{3N_H}{(2-\gamma)\cdot I_H}\rfloor+1)\cdot N_B$ dishonest ratings to Type-B participants; (iv) Type-D and Type-B participants need offer $((\lfloor\frac{3N_H}{(2-\gamma)\cdot I_H}\rfloor+1)+N_B)$ dishonest ratings to target participant; (v) based on the dishonest ratings given to attack target and Type-B participants, referring to probability γ, we can infer the total ratings offered by Type-D participants are $(\lfloor(\lfloor\frac{3N_H}{(2-\gamma)\cdot I_H}\rfloor+1)\cdot(1+N_B)/(1-\gamma)\rfloor+1)$, the honest ratings are $(\lfloor(\lfloor\frac{3N_H}{(2-\gamma)\cdot I_H}\rfloor+1)\cdot(1+N_B)\cdot\gamma/(1-\gamma)\rfloor+1)$. Like Threat Model D, the number of Type-B participants does not affect the adverse effect of Type-D participants, they just receive Type-D participants' trust ingredient for launching more mischievous transactions.

Table 2. Variables and Parameters

TM	N_{nH}	N_{dH}	N_H	R_{nH}	R_{dH}	T^M_{net}	T^G_{net}	N_C	I_H	N_B	N_D	η	γ
A	v	/	h(1-19)	v	/	0.35	0.75, 0.85, 0.95	/	/	/	/	/	/
B	/	v	h(1-19)	/	v	0.35	0.75, 0.85, 0.95	/	/	/	/	/	/
C	/	/	5, 10, 15	/	v	/	/	v	h(1-19)	/	/	/	/
D	/	/	5, 10, 15	/	v	/	/	/	h(1-19)	v	v	/	/
E	/	/	5, 10, 15	/	v	/	/	v	h(1-19)	/	/	0.2	/
F	/	/	5, 10, 15	/	v	/	/	v	h(1-19)	v	v	/	0.2

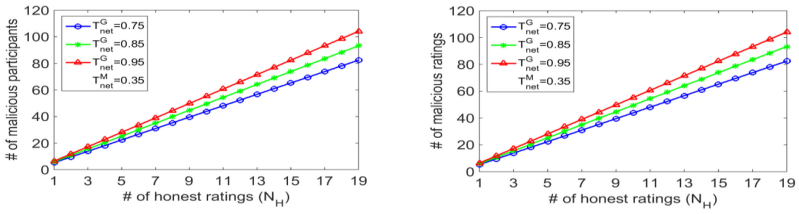
v–observed value (vertical axis), h–observed value (horizon axis), /–variable inapplicable.

4.3 Attack Behavior Analysis and Evaluation

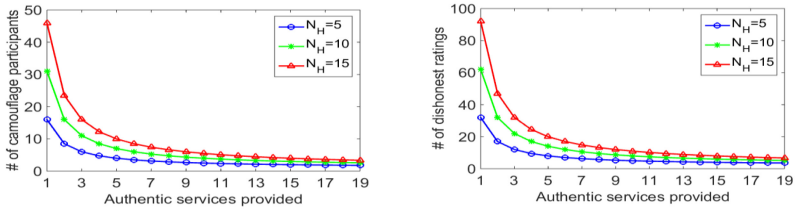
We run a group of experiments to further analyze the six attack behaviors referring to Table 2, wherein "TM" denotes threat models. The attack behavior is deeply analyzed through observing the number (#) of mischievous participants and the number (#) of honest/dishonest ratings with varying SMPs' authentic services.

Figure 1 shows the attack costs under Threat Models A-F. We can observe several interesting and reasonable phenomena: (i) for Threat Models A and B, since the mischievous participants cannot offer authentic services to gain trust ingredient, the # of mischievous participants needed goes up linearly as the honest ratings given by good participants increase, as well as the # of dishonest ratings enlarges linearly; (ii) as the # of authentic services enlarges in Threat Model C, both camouflage participants and dishonest ratings decline gradually. Due to the reinforced trust transitivity of chain-based camouflage participants, it only needs one camouflage participant to provide authentic services. The camouflage participants not only need provide dishonest ratings to target participant, but they also need provide dishonest ratings to their partners along the chain, hence the dishonest ratings are twice that of camouflage participants; (iii) for Threat Model D, we set Type-D participants and Type-B participants equally, although the amount of Type-B participants does not affect the "trust ingredient" of Type-D group. We can observe from Figure 1(c) that the spy participants and dishonest ratings have a declining tendency as authentic services uploaded by Type-D participants increase. However, apart from both Type-D and Type-B individuals give target participant dishonest ratings, each Type-D participant would give all Type-B participants dishonest (positive) ratings to promote trust ingredient. Therefore, the dishonest ratings are much more than that in Threat Model C in which each camouflage participant only needs one dishonest rating to build the chain. In addition, all Type-D participants need provide I_H authentic services rather than only needing one individual. Thereby, the attack cost in Threat Model D is much more than that in Threat Model C; (iv) since the exponential function-based trust propagation deriving from the honest ratings from camouflage participants to good ones in Threat Model E, in fact, diminishes trust ingredient of the mischievous partners in the chain, this naturally requires more camouflage participants and dishonest/honest ratings compared with Threat Model C. Owing to $\log_{(1-\eta)}(1 - \frac{3N_H \cdot \eta}{I_H})$ must be subject to $0 < 1 - \frac{3N_H \cdot \eta}{I_H} < 1$ on the condition $0 < (1\text{-}\eta) < 1$, we have $I_H > 3N_H \cdot \eta$. Normally, since mischievous participants would not like to give honest ratings with high probability η, we set $\eta = 0.2$; (v) since Threat Model F also exists "trust leakage", it needs more attack cost compared with Threat Model D. We set $\gamma = 0.2$ as well.

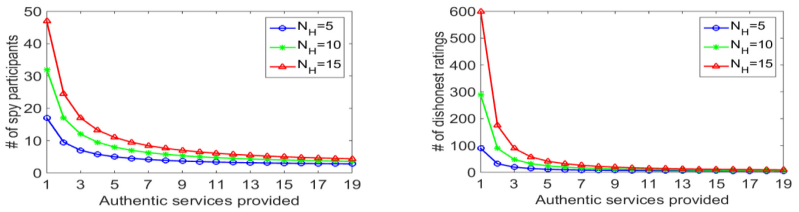
Upon the observation and analysis above, we can conclude the SMPs can indeed decrease attack cost through providing authentic services. This also reveals the ground-truth that the mischievous participants would pay more cost if they occasionally behave honestly rather than purely badly to try to keep them undetected. This reflects the realistic behind reason why mischievous participants are not willing to provide authentic services.



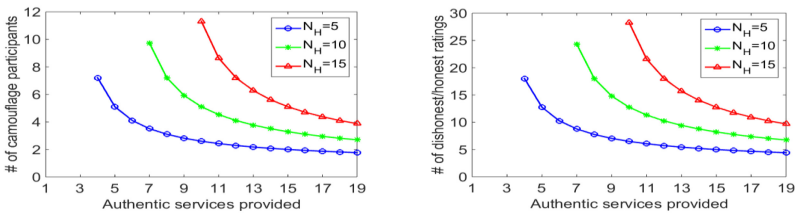
(a) Threat Models A and B



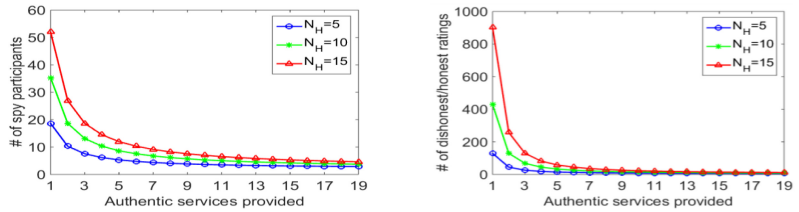
(b) Threat Model C



(c) Threat Model D



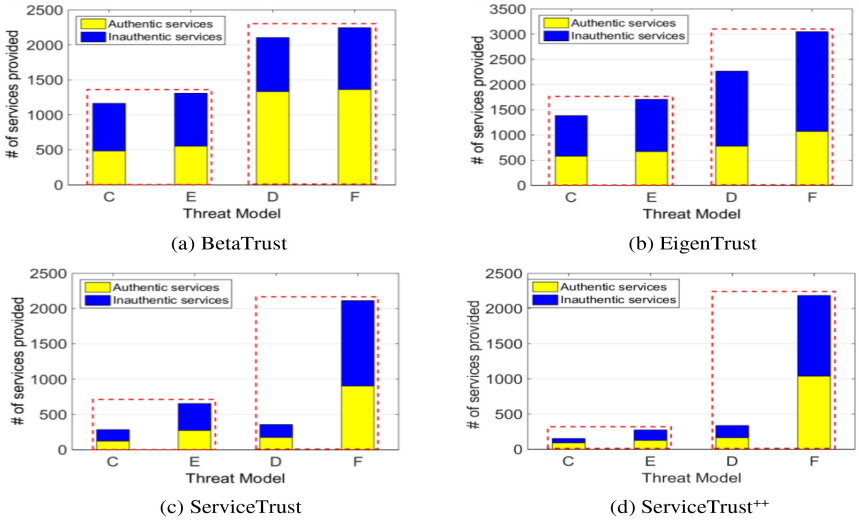
(d) Threat Model E



(e) Threat Model F

Fig. 1. Attack cost evaluation.

The above deduces the attack cost in theory, we next verify our argument through performing a set of experiments using popular trust metrics: BetaTrust [Jøsang and Ismail 2002], EigenTrust [Kamvar et al. 2003], ServiceTrust [Su et al. 2013], and ServiceTrust^{++} [Su et al. 2015]. To fairly evaluate the attack cost, we set same experiment environment as reported in EigenTrust [Kamvar et al. 2003], i.e., the experiment platform includes 60 good participants and 40 mischievous participants. The total transaction number is set to be 10 times the system size, i.e., 1,000 transactions. For



(a) BetaTrust (b) EigenTrust

(c) ServiceTrust (d) ServiceTrust^{++}

Fig. 2. Transactional behaviors of SMPs.

Threat Model C, the probability f is set as 0.4; besides $f = 0.4$, the probability η in Threat Model E is set as 0.2. In addition, the 40 mischievous participants are equally divided into Type-D and Type-B participants in Threat Model D, the probability γ is set as 0.2 likewise in Threat Model F.

From Figure 2, we observe that: (i) the camouflage and spy participants gain high trust ingredient through providing authentic services; however, they perform badly to offer inauthentic services through aggregated global trust; (ii) compared with the SMPs in Threat Models C and D, the more-sophisticated mischievous participants in Threat Models E and F need to provide more authentic services. For example, the numbers of authentic services provided by Threat Models C and E are (483, 551) in BetaTrust, (578, 669) in EigenTrust, (122, 272) in ServiceTrust, and (91, 125) in ServiceTrust^{++}; the numbers of authentic services provided by Threat Models D and F are (1332, 1361), (777, 1070), (173, 902), and (164, 1039), respectively. This interprets the adverse effect of Threat Models E and F is naturally larger than that of Threat Models C and D, i.e., the more-sophisticated misbehavior participants own more opportunities to offer more inauthentic services. For instance, the numbers of inauthentic services provided by Threat Models C and E are (681, 758), (805, 1038), (162, 383), and (62, 149) in the four trust metrics; the numbers of inauthentic services provided by Threat Models D and F are (774, 886), (1488, 1978), (183, 1211), and (173, 1147). Therefore, as analyzed previously, Threat Models E and F indeed need more attack cost and simultaneously bring in more severe attack effect; (iii) compared with the simple trust metrics BetaTrust [Jøsang and Ismail 2002] and EigenTrust [Kamvar et al. 2003], the feedback credibility-based ServiceTrust [Su et al. 2013] and threshold-controlled trust metric ServiceTrust^{++} [Su et al. 2015] could have a much better performance, especially under Threat Models C and D, although they still suffer from the more-sophisticated misbehaviors in Threat Models E and F. Table 3 summarizes and compares the six threat models in terms of attack cost, adverse effect, trust transitivity, and defense strategy.

5 VULNERABILITY ANALYSIS OF TRUST AGGREGATION MODELS

5.1 Trust Aggregation Principle

Trust aggregation in a decentralized network can be deemed as the fusion of feedback information over a graph organized by various nodes (participants). It is naturally subject to two factors:

Table 3. Adverse Behavior Analysis and Evaluation

Threat Model	Mischievous Participants	Mischievous Ratings	Authentic Services	Adverse Effect	Trust Transitivity	Defense strategy
A	linear	linear	none	weak	none	easy
B	linear	linear	none	weak	none	easy
C	medium	small	small	mediocre	existence	solvable
D	medium	large	large	mediocre	existence	solvable
E	medium	small	small	strong	existence	difficult
F	medium	large	large	strong	existence	difficult

(i) pairwise direct trust with local trust aggregation; and (ii) trust propagation kernel-conducted global trust aggregation. Apart from UDTP and TCTP, we add non-propagation (NP) kernel, i.e., the global trust of a participant p_i is aggregated through the direct trust placed on p_i from the neighboring participants plus the recommended trust by other participants. Thereby, trust metrics can be routinely categorized as six combinations with respect to two direct trust aggregation fashions and three trust propagation kernels.

Definition 5.1 (Raw Direct Trust with Non-Propagation Kernel, RNP). Trust is aggregated using the raw direct trust inferred from pairwise positive and negative ratings, the trust, in fact, only captures 1-hop feedback information without trust propagation.

Definition 5.2 (Feedback Credibility-Weighted Direct Trust with Non-Propagation Kernel, CNP). Trust is aggregated using FCW direct trust referring to feedback rater level credibility or feedback rating score level credibility, the trust only captures 1-hop feedback formation without trust propagation.

Definition 5.3 (Raw Direct Trust with Uniformly Distributed Trust Propagation Kernel, RUDP). Trust is aggregated using the raw direct trust inferred from pairwise positive and negative ratings, the trust captures k-hop (network-horizon) feedback information through UDTP kernel.

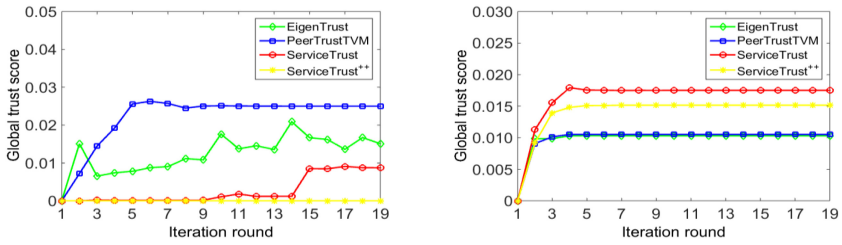
Definition 5.4 (Feedback Credibility-Weighted Direct Trust with Uniformly Distributed Trust Propagation Kernel, CUDP). Trust is aggregated using FCW direct trust referring to feedback rater level credibility or feedback rating score level credibility, the trust captures k-hop (network-horizon) feedback information through UDTP kernel.

Definition 5.5 (Raw Direct Trust with Threshold-Controlled Trust Propagation Kernel, RTCP). Trust is aggregated using the raw direct trust inferred from pairwise positive and negative ratings, the trust, in fact, captures partial intended feedback information through TCTP kernel.

Definition 5.6 (Feedback Credibility-Weighted Direct trust with Threshold-Controlled Trust Propagation Kernel, CTCP). Trust is aggregated using FCW direct trust referring to feedback rater level credibility or feedback rating score level credibility, the trust, in fact, captures partial intended feedback information through TCTP kernel.

5.2 Attack Analysis of Reference Aggregation Models

To deeply study the pros and cons of reference trust aggregation models, we primarily select four representative trust metrics to launch a set of experiments with two strategic Threat Models C and E to exhibit how trust changes as iteration round increases: (i) RUDP trust metric-EigenTrust [Kamvar et al. 2003]; (ii) feedback rater level credibility-based CUDP trust metric-PeerTrustTVM [Xiong and Liu 2004]; (iii) feedback rating score level credibility-based CUDP trust



(a) Threat Model C (b) Threat Model E

Fig. 3. Global trust of camouflage participants.

metric-ServiceTrust [Su et al. 2013]; (iv) feedback rating score level credibility-based CTCP trust metric-ServiceTrust^{++} [Su et al. 2015]. In our experiments, we set the numbers of good, camouflage and pre-trusted participants as 60, 40, and 3; f as 0.4; and η as 0.5. Figure 3 exhibits the global trust of camouflage participants.

We randomly choose one camouflage participant to observe the variation tendency of global trust with increasing the number of iteration rounds. In the presence of the Threat Model C, we observe that: (i) in EigenTrust [Kamvar et al. 2003], the camouflage participant's global trust enlarges gradually as the number of iterations increases; (ii) in PeerTrustTVM [Xiong and Liu 2004], the global trust becomes large shortly within five iteration rounds. In contrast, in ServiceTrust [Su et al. 2013], the global trust goes up slowly. This is because PeerTrustTVM [Xiong and Liu 2004] can promote the trust propagation from a camouflage participant to its partner along the chain, deriving from the high self-trust-based feedback credibility weighted raw direct trust. In comparison, ServiceTrust [Su et al. 2013] dramatically reduces the trust propagation from good participants to camouflage ones by leveraging the dissimilarity between good and camouflage participants. This indicates that the rating similarity-based credibility combined with the dissimilarity-based trust decaying in ServiceTrust [Su et al. 2013] is an effective mechanism to constrain the dishonest ratings to propagate in the presence of camouflage participants; (iii) in ServiceTrust^{++} [Su et al. 2015], the global trust is always zero regardless what the specific iteration round is, this implies that the TCTP kernel succeeds in cutting off trust transitivity paths/edges from good participants to camouflage ones. The primary improvement of ServiceTrust^{++} [Su et al. 2015] over ServiceTrust [Su et al. 2013] is the trust propagation kernel. Put differently, the UDTP kernel-based ServiceTrust [Su et al. 2013] cannot prevent the camouflage participants obtaining positive global trust, but the TCTP kernel-based ServiceTrust^{++} [Su et al. 2015] can throughout block trust propagation through the system-inferred threshold.

From the results of Threat Model E, we observe the four representative trust metrics suffer from this more-sophisticated attack behavior, even the feedback credibility-based trust metrics and TCTP kernel-conducted trust metrics all become ineffective. The root-cause behind lies in Threat Model E invalidates both FCW direct trust and TCTP kernel through making the good in appearance but more-sophisticated mischievous participants imitate good participants as alike as possible. In essence, the radical reasons as analyzed in work [Fan et al. 2017] lies in the pairwise similarity has become inadequate to differentiate camouflage participants from good participants.

Next, we exhibit the inherent vulnerabilities through analyzing the attack behaviors of reference aggregation models in conjunction with transactional performance. We launch six groups of experiments to study the fraction of failed transactions using several referral trust metrics: (i) random trust metric-NoneTrust; (ii) RNP trust metric-BetaTrust [Jøsang and Ismail 2002]; (iii) RTCP trust metric-AdaptiveTrust [Chen et al. 2016], here we can recognize it as a RTCP trust metric due to the setting of minimal honesty trust threshold (0.5) with time slot-based trust update;

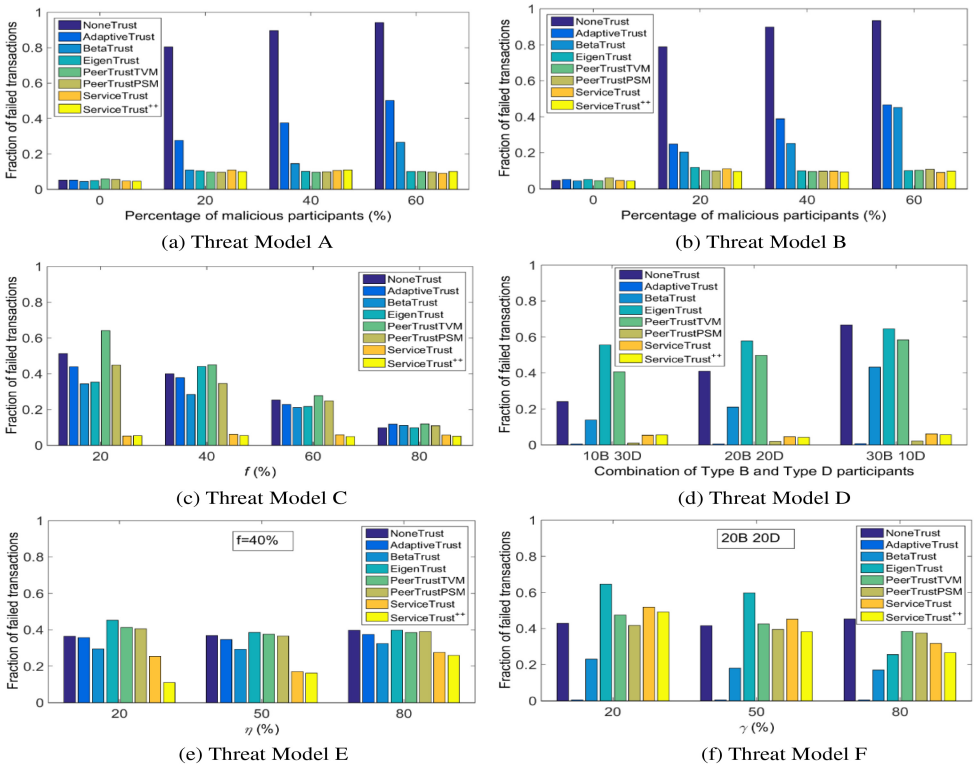


Fig. 4. Performance with Threat Models A–F.

(iv) RUDP trust metric-EigenTrust [Kamvar et al. 2003]; (v) CUDP trust metric-PeerTrustTVM [Xiong and Liu 2004]; (vi) CNP trust metric-PeerTrustPSM [Xiong and Liu 2004]; (vii) CUDP trust metric-ServiceTrust [Su et al. 2013]; and (viii) CTCP trust metric-ServiceTrust^{++} [Su et al. 2015].

To keep experiment configuration identical, we set same environment as EigenTrust [Kamvar et al. 2003], i.e., the experiment platform has 100 participants and 3 pre-trusted participants. In Threat Models A and B, the percentage of mischievous participants varies from 0 to 60%. In Threat Model C, the f increases from 20% to 80%. In Threat Model D, 40 spy participants organize three combinations: (10 Type-B, 30 Type-D), (20 Type-B, 20 Type-D), (30 Type-B, 10 Type-D). In Threat Model E, we set the number of camouflage participants as 40 and keep $f = 0.4$, then vary η from 0.2 to 0.8. In Threat Model F, we divide the spy participants into 20 Type-B and 20 Type-D, then change γ from 0.2 to 0.8. Figure 4 exhibits the performance.

For Threat Models A and B, most trust metrics are effective deriving from the zero-value direct trust from good individuals to mischievous ones, which leads to zero global trust for mischievous participants, such as EigenTrust [Kamvar et al. 2003], PeerTrustTVM [Xiong and Liu 2004], PeerTrustPSM [Xiong and Liu 2004], ServiceTrust [Su et al. 2013], and ServiceTrust^{++} [Su et al. 2015]. However, apart from the random trust metric NoneTrust, which randomly selects transacted participants, the RTCP trust metric AdaptiveTrust [Chen et al. 2016] and RNP trust metric BetaTrust [Jøsang and Ismail 2002] cannot conquer the two simple attacks. In BetaTrust [Jøsang and Ismail 2002], since it derives trust using the difference between successful and unsuccessful transactions without normalizing the direct trust into the interval [0, 1], thus, the mischievous participants can obtain non-zero trust scores. Consequently, some mischievous participants might be

selected as transacted targets according to the probabilistic selection criterion. For AdaptiveTrust [Chen et al. 2016], it assumes each participant's initial trust score as 0.5, thus the mischievous participants can also be selected as transacted targets with a big probability, especially in the beginning stage the difference of most participants' trust is subtle.

For Threat Model C, the random, RNP, CNP, CUDP, and RUDP trust metrics all become invalid due to the existence of strategically mischievous behaviors of the camouflage participants, which are acting as good participants at a certain probability f. For the CUDP trust metric PeerTrustTVM [Xiong and Liu 2004] and CNP trust metric PeerTrustPSM [Xiong and Liu 2004], they behave poorly, since both the feedback rater level credibility and feedback rating score level credibility cannot effectively prevent good participants to give direct trust on camouflage participants. However, ServiceTrust [Su et al. 2013] uses the CUDP trust metric by employing the feedback rating score level credibility to weight direct trust, which effectively decreases the direct trust from good participants to camouflage participants. In addition, with the CTCP trust metric, ServiceTrust^{++} [Su et al. 2015] can effectively cut off dishonest trust propagation to camouflage individuals even though the direct trust weighted by feedback rating score level credibility is not dropped to the ground-truth level (zero). Hence, the CTCP trust metric can effectively conquer strategic camouflage attack.

For Threat Model D, AdaptiveTrust [Chen et al. 2016] conquers, this is because there exists an amending mechanism, i.e., once a participant finds another transacted participant gives a bad service, then it sets this participant's trust as 0.0. For CNP-based PeerTrustPSM [Xiong and Liu 2004] and CUDP-based ServiceTrust [Su et al. 2013], they can conquer this kind of spy attack, since the similarity-inferred feedback credibility approaches are effective to decline the trust of purely mischievous Type-B participants deriving from the total dissimilarity between good participants and spy participants. In ServiveTrust^{++} [Su et al. 2015], both the feedback rating score level credibility and controlled trust propagation kernel doubly constrain these spy participants' trust aggregation. However, for the RNP-based BetaTrust [Jøsang and Ismail 2002], RUDP-based EigenTrust [Kamvar et al. 2003], and CUDP-based PeerTrustTVM [Xiong and Liu 2004], they all suffer from this spy attack. BetaTrust [Jøsang and Ismail 2002] and EigenTrust [Kamvar et al. 2003] would assign high trust scores to Type-D participants. Type-D participants can manipulate PeerTrustTVM [Xiong and Liu 2004] through the high feedback rater level credibility. In summary, the adequate feedback credibility, supported in PeerTrustPSM [Xiong and Liu 2004], ServiceTrust [Su et al. 2013], and ServiceTrust^{++} [Su et al. 2015], can effectively control and block the direct trust from good participants to spy participants.

For Threat Model E, all the trust metrics suffer from this more-sophisticated adversary behavior, even the CUDP-based ServiceTrust [Su et al. 2013] and CTCP-based ServiceTrust^{++} [Su et al. 2015] cannot resist in spite of a better performance compared with other trust metrics. This is because the tremendously similar transactional behavior between camouflage participants with probabilistically honest ratings and good participants makes the trust metrics become extremely difficult to distinguish these good in appearance but mischievous participants. Singly from the viewpoint of direct trust aggregation, it is hard to degrade the raw direct trust or FCW direct trust, not to mention the trust propagation kernels. Compared to the raw direct trust and inadequate FCW direct trust, such as AdaptiveTrust [Chen et al. 2016], BetaTrust [Jøsang and Ismail 2002], EigenTrust [Kamvar et al. 2003], and PeerTrustTVM [Xiong and Liu 2004], the adequate FCW direct trust can make the ratio of failed transactions decline to some extent, such as PeerTrustPSM [Xiong and Liu 2004], ServiceTrust [Su et al. 2013], and ServiceTrust^{++} [Su et al. 2015].

For Threat Model F, apart from AdaptiveTrust [Chen et al. 2016] with amending mechanism, all other trust metrics suffer from this more-sophisticated spy misbehavior as well. The fundamental

reason also lies in these spy participants behave almost alike as good participants to a large extend through adjusting the probability of offering honest ratings.

5.3 Summary of Attack Analysis

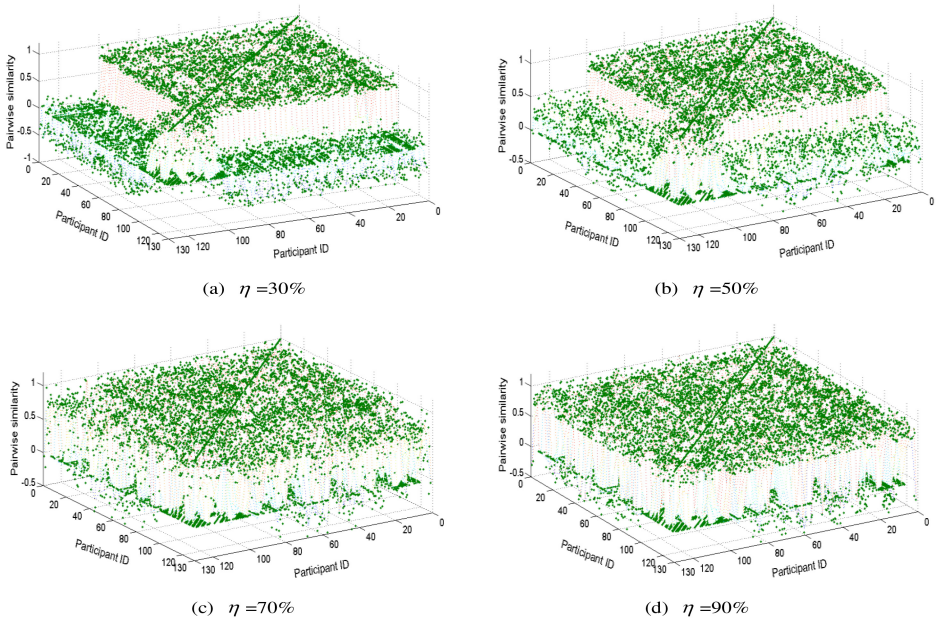
Upon the attack analysis above, we can conclude that: (i) for independently and collectively malicious behaviors in Threat Models A and B, the raw direct trust or FCW direct trust can be an effective fashion as transaction enlarges, accordingly the aggregated trust levels can differentially represent good and mischievous participants using whatever NP, UDTP or TCTP kernels; (ii) for strategic camouflage and spy behaviors in Threat Models C and D, the raw direct trust becomes ineffective. The adequate FCW direct trust becomes valid to a large extent but cannot throughout constrain the trust propagation from good to the camouflage and spy participants into ground-true level. If TCTP kernel can be employed to control how to propagate/block trust among different categories of participants, then the trust metrics would be effective. That is to say a reliable trust metric needs to employ an adequate FCW direct trust plus an appropriate TCTP kernel to resist camouflage and spy attacks; (iii) for more-sophisticated mischievous participants in Threat Models E and F, considering the extremely similar transactional behavior with good participants, it is hard to defend due to high similarity-inferred feedback credibility, in addition to the invalid similarity-inferred threshold. Thereby, even the adequate FCW direct trust and TCTP kernel cannot conquer these more-sophisticated attacks, but can decline the trust levels of camouflage and spy participants to an extent.

To further dig out the root-causes why diverse categories of trust metrics suffer from the more-sophisticated camouflage and spy attacks, we utilize the real-word interactive network-Epinions to launch a group of experiments to unveil the behind reasons. We add 30 nodes into the 100 regular/good nodes organized Epinions network, recognizing the 30 added nodes as strategically mischievous nodes with Threat Models E and F. For regular nodes, we straightly adopt Zipf distribution to assign pairwise edge weight (direct trust) for each pair of connected nodes. The regular nodes select a decimal from interval [0.85, 1.0] to rate the added camouflage nodes, and the added nodes select a decimal from interval $[\eta - 0.05, \eta + 0.05]$ to rate regular nodes under Threat Model E.

Figure 5 exhibits the pairwise similarity among the 130 nodes wherein 100 nodes (ID = 1–100) belong to regular nodes, 30 nodes (ID = 101–130) pertain to mischievous nodes. The experimental results interpret the pairwise similarity between camouflage and good nodes gradually increases as the η enlarges from 30% to 90%. The similarity between camouflage and good nodes is almost in the same level between good nodes themselves when η is up to 90%. This indicates when η is large enough the similarity-based feedback credibility becomes invalid to degrade raw direct trust from good to camouflage nodes, as well as the similarity-based threshold becomes invalid to block dishonest trust propagation owning to high similarity-based direct trust. Similar results can be also obtained under Threat Model F while extending the γ step by step. This reveals our proposed Threat Models E and F can indeed make the more-sophisticated mischievous participants behave extremely similarly as good ones, making them become extremely difficult to defend.

To explicitly illuminate the features of the state of the art trust metrics, we categorize them into six communities to sketch the weakness and attack resilience with respect to the six threat models in Tables 4–9. THR_A-F denotes Threat Models A-F, "+" denotes trust metric can conquer threat model, "-" oppositely indicates trust metric cannot.

RNP-based Trust Metrics. Marti et al. [2004] proposed a voting reputation system wherein a node q would contact a set of nodes *Res* for their own local opinion on the responder r, wherein the final reputation was calculated by summing each voter's rating in addition to the node q's own feedback rating. Jøsang et al. [2006] defined the target Z's reputation score at time t as $R^t(Z) =$



(a) η =30% (b) η =50%

(c) η =70% (d) η =90%

Fig. 5. Pairwise similarity under Threat Model E.

Table 4. Attack Analysis on RNP-based Trust Metrics

Trust Metrics	Defense Countermeasure					
	THR_A	THR_B	THR_C	THR_D	THR_E	THR_F
Marti et al. [2004]	+	+	-	-	-	-
Jøsang et al. [2006]	+	+	-	-	-	-
TrustWalker [2009]	+	+	+	+	-	-
Wang et al. [2009]	+	+	-	-	-	-
Liu et al. [2011]	+	+	+	+	-	-
Li et al. [2012]	+	+	-	-	-	-
CommTrust [2014]	+	+	-	-	-	-
Shabut et al. [2015]	+	+	+	+	-	-

$(\delta + 2a)/(\delta + \sigma + 2)$, where δ and σ denoted the numbers of positive and negative observations, and a expressed a prior or base rate to leverage the weight of positive rating. TrustWalker [Jamali and Ester 2009] estimated the rating for user u on target item i using different random walks, it summed the feedback information returned by different k-scoped random walks as the rating for the source user u on the target item i. Wang et al. [2009] straightly defined the final trust of node i toward node j in d field as the integration of local trust value L_{ijd} and global trust value T_{jd} with a proportional factor. Liu et al. [2011] utilized the first-hop of trust transitivity $T_{a1,a2}$ and the hop number to infer trust transitivity result $T_{a1,a(j+1)}$ for a social trust path $p_{(a_1,\ldots,a_{(j+1)})}$ as $T_{a1,a(j+1)} = T_{a1,a2}+k_2$, where k_2 denoted the slope of Deviation Line referring to the Base Line that started from coordination (1, $T_{a1,a2}$). Li et al. [2012] utilized local trust degree (LTD) $D_L(N_i, N_j)$ placed on node N_j from node N_i and feedback trust degree (FTD) $R_U(N_i, N_j)$ from the third-party nodes, which had interactions with N_j to aggregate global trust. CommTrust [Zhang et al. 2014] defined the overall trust score T for a seller as the weighted aggregation of dimension trust scores, i.e.

Table 5. Attack Analysis on CNP-based Trust Metrics

Trust Metrics	Defense Countermeasure					
	THR_A	THR_B	THR_C	THR_D	THR_E	THR_F
PeerTrustPSM [2004]	+	+	-	+	-	-
TrustGauard [2005]	+	+	-	-	-	-
Yu et al. [2008]	+	+	+	+	-	-
Li and Zhu [2014]	+	+	+	+	-	-
DCMR [2009]	+	+	+	+	-	-

Table 6. Attack Analysis on RUDP-based Trust Metrics

Trust Metrics	Defense Countermeasure					
	THR_A	THR_B	THR_C	THR_D	THR_E	THR_F
SocialTrust [2010]	+	+	-	-	-	-
SORT [2013]	+	+	+	+	-	-
GFTrust [2016b]	+	+	-	-	-	-
PageTrust [2008]	+	+	-	-	-	-
EigenTrust [2003]	+	+	-	-	-	-
PowerTrust [2007]	+	+	-	-	-	-
Guha et al. [2004]	+	+	-	-	-	-
Dual-EigenRep [2010]	+	+	-	-	-	-
Walter et al. [2009]	+	+	-	-	-	-

$T = \sum_{d=1}^{m} t_d \cdot w_d$, where t_d and w_d represented the trust score and weight for the dimension d ($d = 1, \ldots, m$). Then it utilized the beta function-based expectation to calculate trust score $t_d = (|\{v_d = +1\}| + m/2)/(n + m)$, where $n = \{v_d | v_d = +1 \vee v_d = -1\}|$ was the binary positive and negative ratings. Shabut et al. [2015] adopted the direct and indirect trust to aggregate trust score T_{ij} for nodes i and j, i.e., $T_{ij} = w_d \cdot T_{ij}^d + w_i \cdot T_{ij}^i$, where $w_d + w_i = 1$. The direct and indirect trust were inferred by the beta function-based expectation.

CNP-based Trust Metrics. The personalized similarity was utilized to calculate feedback credibility of third-party recommenders in PeerTrustPSM [Xiong and Liu 2004], TrustGauard [Srivatsa et al. 2005], and Yu et al. [2008]. Li and Zhu [2014] also utilized the cosine-based similarity as credibility for recommendation in body area networks. DCMR [Bao et al. 2009] employed the similar users' similarity as the credibility to calculate the rating on an item in collaborative filtering.

RUDP-based Trust Metrics. SocialTrust [Caverlee et al. 2010] inferred a user i's trust from the trust and relationship quality $R(j)$ of other users, as well as the number of user j's relationship, i.e., $Tr_q(i) = \lambda \sum_{j \in rel} R(j) \cdot Tr_q(j)/|rel(j)| + (1 - \lambda)F(i)$, where $rel(j)$ denoted the set of contacts of user j, $F(i)$ represented the feedback rating aggregated by the trust group governing assessment. The relationship quality R(j) was a scoped random walk. SORT [Can and Bhargava 2013] defined participant p_i's estimation about the reputation of p_j through collecting all recommendation trust from its acquaintance p_k, namely, $er_{ij} = \sum_{p_k \in T_i} (rt_{ik} \cdot \eta_{kj} \cdot r_{kj})$, where η_{kj} was the number of p_k's acquaintances that provided recommendations during the calculation of r_{kj}, rt_{ik} denoted the recommendation trust from p_k, and T_i was the set of trustworthy acquaintances selected by p_i. GFTrust [Jiang et al. 2016b] mirrored trust propagation from nodes s to d as network flow with intermediate node v_i in the path $(s, v_1, \ldots, v_m, d)$, i.e., when a flow flw passed this path, the resulting flow would become $flw \cdot \prod_{i \in [1,m]} (1 - leak(v_i))$, where $leak(v_i)$ denoted the flow leakage function. PageRank [Page et al. 1999] is the pioneering page ranking algorithm

Table 7. Attack Analysis on CUDP-based Trust Metrics

Trust Metrics	Defense Countermeasure					
	THR_A	THR_B	THR_C	THR_D	THR_E	THR_F
PeerTrustTVM [2004]	+	+	-	-	-	-
Hu et al. [2008]	+	+	+	+	-	-
EigenTrust^{++} [2012]	+	+	+	+	-	-
ServiceTrust [2013]	+	+	+	+	-	-
Deng et al. [2017]	+	+	+	+	-	-

Table 8. Attack Analysis on RTCP-based Trust Metrics

Trust Metrics	Defense Countermeasure					
	THR_A	THR_B	THR_C	THR_D	THR_E	THR_F
Chen et al. [2008]	+	+	+	-	+	-
Wang and Li [2011]	+	+	-	-	-	-
ReTrust [2012]	+	+	-	-	-	-
Chen et al. [2014]	+	+	+/-	+/-	+/-	+/-
AdaptiveTrust [2016]	+	+	+/-	+	+/-	+

through propagating rank value from one page to its neighboring page(s) along the hyperlink or randomly to another non-hyperlinked page with a probability, it's a typically UDTP. On the basis of PageRank, PageTrust [Kerchove and Dooren 2008] was proposed to infer the trust value for each page. EigenTrust [Kamvar et al. 2003] adopted PageRank into trust management and accomplished UDTP through replacing the degree-based pairwise weight with the ratio of satisfied interactions. The similar manipulations also emerged in PowerTrust [Zhou and Hwang 2007], Dual-EigenRep [Fan et al. 2010], and Guha et al. [2004]. Walter et al. [2009] calculated indirect trust via iterative computation of local trust matrix, the kth power of matrix represented the hops of UDTP.

CUDP-based Trust Metrics. PeerTrustTVM [Xiong and Liu 2004] employed each participant's trust ratio as feedback credibility to indicate how much trust it could propagate to its neighbor(s) using UDTP kernel. For a pair of participants, Hu et al. [2008] defined feedback credibility by multiplying their transaction density factor and difference of ratings, then aggregated global trust through power-iteration-based matrix computation. EigenTrust^{++} [Fan et al. 2012] utilized the pairwise similarity as feedback credibility to weight raw direct trust for further global trust aggregation via UDTP kernel. ServiceTrust [Su et al. 2013] defined positive similarity and negative similarity to merge the feedback credibility, subsequently aggregated global trust via UDTP kernel. Deng et al. [2017] associated the preference similarity to model trust degree for a pair of users. If no direct link existed, then the shortest path-based multiplication was adopted.

RTCP-based Trust Metrics. Chen et al. [2008] proposed an inter-cluster recommendation trust concept and defined the total trust index from node N_i to N_j as $\Gamma(N_i, N_j) = \alpha T_D^{ij} + \beta T_R^j$, $\alpha, \beta \geq 0$, $\alpha + \beta = 1$, where α and β were the impact weights of direct trust T_D^{ij} and recommendation trust T_R^j, respectively. The inter-cluster recommendation trust for N_j was defined as $T_R^j = \sum_{i=1}^{n} T_D^{hi} \cdot T_D^{ij} / \sum_{i=1}^{n} T_D^{hi}$, where $T_D^{hi} > H$, this condition indicated the cluster head (CH) would discard their recommendation to save bandwidth if the node's direct trust by CH was lower than a threshold value H. Wang and Li [2011] calculated the aggregated rating by adopting Gaussian distribution-based upper control limit and lower control limit to filter out the marginal ratings out of the range of boundary. ReTrust [He et al. 2012] utilized the similar range to block bad-mouthing

Table 9. Attack Analysis on CTCP-based Trust Metrics

Trust Metrics	Defense Countermeasure					
	THR_A	THR_B	THR_C	THR_D	THR_E	THR_F
Song et al. [2005]	+	+	-	-	-	-
O'Donovan and Smyth [2005]	+	+	+	+	-	-
ServiveTrust^{++} [2015]	+	+	+	+	-	-
GroupTrust [2017]	+	+	+	+	-	-

recommendation, in addition to the indirect trust inference through trust propagation with the requirement that all direct trust between intermediate nodes must be greater than a threshold. Chen et al. [2014] used a threshold to select trustworthy recommenders to infer indirect trust, in addition to the consideration the trust between the originator node and recommender as a weight to multiply the recommendation trust. In AdaptiveTrust [Chen et al. 2016], the authors set a minimal honesty trust threshold (0.5) during time slot-based trust update.

CTCP-based Trust Metrics. Song et al. [2005] utilized threshold-controlled selected local trust score t_{ji} placed on i from another participant j and corresponding aggregation weight to calculate the global trust, the weight was aggregated by the participant j's trust, transaction date and amount. Consider participant j's trust was under the computation simultaneously, therefore, the procedure of global trust, in fact, was a multiple iterations, which implied the trust was propagated through threshold-controlled edges. O'Donovan and Smyth [2005] qualified which producer profiles were allowed to participate the rating recommendation process through predefining a threshold with which the item/profile-level trust values of candidate producer profiles compared, besides this, the qualified profiles also needed to take the harmonic mean of trust and similarity as recommendation credibility to infer the rating for an item in a consumer profile. ServiveTrust^{++} [Su et al. 2015] employed the similarity as feedback credibility to weight the raw direct trust, in addition that a threshold over the holistic network was set to control the trust propagation. GroupTrust [Fan et al. 2017] utilized the exponent-based feedback credibility, and proposed a fine-grained threshold-controlled trust propagation through studying the susceptible-infected-recovered model.

6 DESIGN PRINCIPLES OF DECENTRALIZED TRUST MANAGEMENT

Based on the formal analysis and experimental evaluation presented, we make three important observations and articulate the three design principles for effective trust management.

(1) Different threat modes have different adverse effects and different attack costs. Threat Models E and F have much more serious adverse effect than Threat Models A, B, C, D, because such attacks make it more difficult to differentiate mischievous participants from good participants in terms of both service provisioning or feedback rating behaviors. Thus, the robustness of decentralized trust management should be based on establishing trust by identifying both the list of priorities in terms of a range of services and the measurement for the service quality. In this article, we only cover the request serving and feedback rating as two types of services. We use the feedback rating to measure the request serving quality, and we leverage feedback similarity as a way to measure feedback rating quality, which are vulnerable under the most strategically malicious Threat Models E and F.

(2) The feedback credibility-weighted local trust aggregation can effectively regulate the dishonest ratings when the fraction of malicious participants is much smaller than the fraction of good participants. However, the effectiveness of using the similarity-based feedback credibility for regulating the local trust aggregation may no longer be effective when the fraction of good

participants is out-numbered by the fraction of mischievous participants. This observation further indicates the importance of developing trust management algorithms that can tolerate unexpected errors and survive unexpected malicious attacks.

(3) The threshold controlled trust-propagation (TCTP) kernel provides the double-filtered function to regulate the trust propagation from good participants to the malicious participants. This enables the global trust scores of mischievous participants to be dramatically decreased, providing flexibility to control how trust is partially propagated over the network of participants through topological traversal. Thus, the TCTP kernel presents a more appropriate defense against those strategically mischievous attacks compared with the uniformly distributed trust propagation (UDTP) kernel.

7 APPLICATION PROSPECT

Edge Computing Trust. One of the main attractions of edge computing is to improve the computation and processing cost and time by leveraging edge nodes instead of always connecting to the remote Cloud servers. Porambage et al. [2018] pointed out "trust" as a significant mechanism in critical 5G use cases like remote surgeries, emergency autonomous vehicles, factory automation and tele-operated driving (e.g., drones). Several recent efforts have articulated the need to design appropriate trust management for edge cloud. Yan et al. [2014] articulated the role of trust management for reliable data fusion and mining, qualified services with context-awareness, and for enhancing user privacy and information security. Dang and Hoang [2017] demonstrated the use of trust management for data protection and performance improvement at edge servers. Garcia et al. [2015] also presented the challenges for trust, security and privacy in edge-centric computing.

Trust management in Blockchain Systems. Blockchain technique [Nakamoto 2008] employs a decentralized P2P network to achieve consensus on a distributed public ledger of transactions through calculating the proofs of work for different peers (miners). The representative application of blockchain is the Bitcoin system. Users in the Bitcoin system are anonymous, and can use their public key hash as their pseudo-identity to interact with the system. However, the blockchain and the miner still confront some transactional risks at present, such as the double-spending problem and selfish mining problem [Eyal and Sirer 2018; Heilman et al. 2015; Karame et al. 2015; Zhang et al. 2019].

One approach to mitigate such risks is to incorporate trust management into the peer-to-peer network of the blockchain system. For instance, consider the private blockchain and consortium blockchain scenarios, assume a service consumer u needs to pay a certain number of bitcoins to a service provider v using its address a_u. If the transaction is successfully accomplished, i.e., user v receives bitcoins indeed from user u, then user v will provide a positive rating to user u's address a_u, otherwise a negative rating if the payment does not accomplish successfully, i.e., user v does not receive bitcoins from user u. Based on the feedback rating information, the blockchain system can produce a trust score for each anonymous address through employing our previously-introduced trust metrics, i.e., yield an overall estimation on the trustworthiness for each address. In subsequent transactions, service consumers can select the addresses with high trust scores as transacted targets, which can effectively block the transactions from the potentially-mischievous addresses with low trust scores, accordingly decline the risk of unsuccessful transactions.

The pull-in trust management can bring in two-facet advantages: (i) keep the anonymity and un-traceability of users; and (ii) guarantee the trustworthiness of transactional behaviors among users. The above is to rate the anonymous address, furthermore, we can also rate the user u with multiple addresses. Taking into account the linkability between addresses and users [Karame et al. 2015; Meiklejohn et al. 2013], i.e., identify the ownership of different addresses for users, we can

straightly accumulate the positive and negative ratings to multiple addresses offered by other users with which user u has had transactions, and aggregate a comprehensive trust score for user u.

Blockchain-based Trust Management. Trust scores and feedback ratings are important pieces of data that should be carefully protected in any trust management system. Blockchain technology can be utilized as an excellent mechanism to keep track of trust ratings and store them in the public and secure global ledger, such that a trust rating score once admitted into the blockchain, it will be absolutely secure from malicious modification and compromises. We argue that this is an interesting research and development project with high practical relevance.

8 CONCLUSIONS AND FUTURE WORK

We describe decentralized trust management models and their efficiency and robustness from three unique perspectives. First, we study the risk factors and adverse effects of six common threat models. Second, we review the representative trust aggregation models and trust metrics. Third, we present an in-depth analysis and comparison of these reference trust aggregation methods with respect to effectiveness and robustness. We argue that our comparative study advances the understanding of adverse effects of present and future threats and the robustness of different trust metrics. It may also serve as a guideline for research and development of next generation trust aggregation algorithms and services in the anticipation of risk factors and mischievous threats.

ACKNOWLEDGMENTS

The authors thank Prof. Dr. Fang and anonymous reviewers for their helpful suggestions and comments that significantly improved the presentation of the article.

REFERENCES

Ankit Agrawal and A. K. Verma. 2016. A review & impact of trust schemes in MANET. In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*. ACM, 26:1–26:7.

Usama Ahmed, Imran Raza, and Syed Asad Hussain. 2019. Trust evaluation in cross-cloud federation: Survey and requirement analysis. *ACM Comput. Surv.* 52, 1 (2019), 19:1–19:37.

P. G. Balaji and D. Srinivasan. 2010. An introduction to multi-agent systems. In *Proceedings of Innovations in Multi-Agent Systems and Applications-1*. Springer, Berlin, 1–27.

Hongji Bao, Tengjiao Wang, Hongyan Li, and Dongqing Yang. 2009. DCMR: A method for combining user-based and trust-based recommendation. In *Proceedings of the International Conference on Computational Intelligence and Software Engineering*. IEEE, 1–5.

Diego De Siqueira Braga, Marco Niemann, Bernd Hellingrath, and Fernando Buarque De Lima Neto. 2018. Survey on computational trust and reputation models. *ACM Comput. Surv.* 51, 5 (2018), 101:1–101:40.

Ahmet Burak Can and Bharat Bhargava. 2013. SORT: A self-organizing trust model for peer-to-peer systems. *IEEE Trans. Depend. Secure Comput.* 10, 1 (2013), 14–27.

James Caverlee, Ling Liu, and Steve Webb. 2010. The SocialTrust framework for trusted social information management: Architecture and algorithms. *Info. Sci.* 180, 1 (2010), 95–112.

Younghun Chae, Lisa Cingiser Dipippo, and Yan Lindsay Sun. 2015. Trust management for defending on-off attacks. *IEEE Trans. Parallel Distrib. Syst.* 26, 4 (2015), 1178–1191.

Aiguo Chen, Guoai Xu, and Yixian Yang. 2008. A cluster-based trust model for mobile ad hoc networks. In *Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 1–4.

Ing-Ray Chen, Fenye Bao, MoonJeong Chang, and Jin-Hee Cho. 2014. Dynamic trust management for delay tolerant networks and its application to secure routing. *IEEE Trans. Parallel Distrib. Syst.* 25, 5 (2014).

Ing Ray Chen, Fenye Bao, and Jia Guo. 2016. Trust-based service management for social internet of things systems. *IEEE Trans. Depend. Secure Comput.* 13, 6 (2016), 684–696.

Jin-Hee Cho, Kevin Chan, and Sibel Adali. 2015. A survey on trust modeling. *ACM Comput. Surv.* 48, 2 (2015), 28:1–28:40.

Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. 2011. A survey on trust management for mobile ad hoc networks. *IEEE Commun. Surveys Tutor.* 13, 4 (2011), 562–583.

Karen S. Cook, Toshio Yamagishi, and Robin Cooper. 2005. Trust building via risk taking: A cross-societal experiment. *Soc. Psychol. Quart.* 68, 2 (2005), 121–142.

Thanh Dat Dang and Doan Hoang. 2017. A data protection model for fog computing. In *Proceedings of 2nd International Conference on Fog and Mobile Edge Computing*. IEEE, 32–38.

Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. 2017. On deep learning for trust-aware recommendations in social networks. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 5 (2017), 1164–1177.

Ittay Eyal and Emin Gn Sirer. 2018. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM* 61, 7 (2018), 95–102.

Xinxin Fan, Mingchu Li, Yizhi Ren, and Jianhua Ma. 2010. Dual-EigenRep: A reputation-based trust model for P2P file-sharing networks. In *Proceedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*. IEEE, 358–363.

Xinxin Fan, Ling Liu, Mingchu Li, and Zhiyuan Su. 2012. EigenTrust^{++}: Attack resilient trust management. In *Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 416–425.

Xinxin Fan, Ling Liu, Mingchu Li, and Zhiyuan Su. 2017. GroupTrust: Dependable trust management. *IEEE Trans. Parallel Distrib. Syst.* 28, 4 (2017), 1076–1090.

Diego Gambetta (Ed.). 1998. *Can We Trust Trust?* Basil Blackwell, Oxford, New York, NY.

Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. 2015. Edge-centric computing: Vision and challenges. *SIGCOMM Comput. Commun. Rev.* 45, 5 (2015), 37–42.

Jennifer Golbeck. 2005. Personalizing applications through integration of inferred trust values in semantic web-based social networks. In *Proceedings of the Semantic Network Analysis Workshop*. 15–28.

Jennifer Golbeck. 2006a. Combining provenance with trust in social networks for semantic web content filtering. In *Proceedings of the Conference on Provenance and Annotation of Data (IPAW'06)*, L. Moreau and I. Foster (eds.). Springer, 101–108.

Jennifer Golbeck. 2006b. Trust on the world wide web: A survey. *Found. Trends Web Sci.* 1, 2 (2006), 131–197.

Kannan Govindan and Prasant Mohapatra. 2012. Trust computations and trust dynamics in mobile ad hoc networks: A survey. *IEEE Commun. Surveys Tutor.* 14, 2 (2012), 279–298.

Jones Granatyr, Vanderson Botelho, Otto Robert Lessing, Edson Emílio Scalabrin, Jean-Paul Barthès, and Fabrício Enembreck. 2015. Trust and reputation models for multiagent systems. *ACM Comput. Surv.* 48, 2 (2015), 27:1–27:42.

Mark S. Granovetter. 1973. The strength of weak ties. *Amer. J. Sociol.* 34, 3 (1973), 1360–1380.

R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*. ACM, New York, NY, 403–412.

Sheikh Mahbub Habib, Sebastian Ries, and Max Muhlhauser. 2011. Towards a trust management system for cloud computing. In *Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 933–939.

Yu Han, Zhiqi Shen, Cyril Leung, Chunyan Miao, and Victor R. Lesser. 2013. A survey of multi-agent trust management systems. *IEEE Access* 1 (2013).

Daojing He, Chun Chen, Sammy Chan, Jiajun Bu, and Athanasios V. Vasilakos. 2012. ReTrust: Attack-resistant and lightweight trust management for medical sensor networks. *IEEE Trans. Info. Technol. Biomed.* 16, 4 (2012), 623–632.

Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse attacks on Bitcoin's peer-to-peer network. In *Proceedings of the 24th USENIX Conference on Security Symposium*. USENIX Association.

Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. 2009. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.* 42, 1 (2009), 1:1–1:31.

Hao Hu, Rongxing Lu, Zonghua Zhang, and Jun Shao. 2017. REPLACE: A reliable trust-based platoon service recommendation scheme in VANET. *IEEE Trans. Vehic. Technol.* 66, 2 (2017), 1786–1797.

Jianli Hu, Quanyuan Wu, and Bin Zhou. 2008. Distributed and effective reputation mechanism in P2P systems. In *Proceedings of the International Conference on Computer Science and Software Engineering*. IEEE, 110–115.

Kai Hwang and Deyi Li. 2010. Trusted cloud computing with secure resources and data coloring. *IEEE Internet Comput.* 14, 5 (2010), 14–22.

Paul J. Jackson. 1999. *Virtual Working: Social and Organisational Dynamics*. Routledge, London, UK.

Mohsen Jamali and Martin Ester. 2009. TrustWalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 397–406.

Wenjun Jiang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Jie Wu. 2016a. Understanding graph-based trust evaluation in online social networks: Methodologies and challenges. *ACM Comput. Surv.* 49, 1 (2016), 10:1–10:35.

Wenjun Jiang, Jie Wu, Feng Li, Guojun Wang, and Huanyang Zheng. 2016b. Trust evaluation in online social networks using generalized network flow. *IEEE Trans. Comput.* 65, 3 (2016), 952–963.

Wenjun Jiang, Jie Wu, and Guojun Wang. 2015. On selecting recommenders for trust evaluation in online social networks. *ACM Trans. Internet Technol.* 15, 4 (2015), 14:1–14:21.

Audun Jøsang, Ross Hayward, and Simon Pope. 2006. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference*. Australian Computer Society, Inc., Hobart, Australia.

Audun Jøsang and Roslan Ismail. 2002. The beta reputation system. In *Proceedings of the 15th BLED Electronic Commerce Conference E-Reality: Constructing the E-Economy*. 1–14.

Audun Jøsang, Roslan Ismail, and Colin Boyd. 2007. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* 43, 2 (2007), 618–644.

Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. 2003. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International Conference on World Wide Web*. ACM, 640–651.

Ghassan O. Karame, Elli Androulaki, Marc Roeschlin, Arthur Gervais, and Srdjan Čapkun. 2015. Misbehavior in bitcoin: A study of double-spending and accountability. *ACM Trans. Info. Syst. Secur.* 18, 1 (2015), 2:1–2:32.

Cristobald De Kerchove and Paul Van Dooren. 2008. The PageTrust algorithm: How to rank web pages when negative links are allowed? In *Proceedings of Siam International Conference on Data Mining*. SIAM, 346–352.

Chaker Abdelaziz Kerrache, Carlos Tavares Calafate, Juan Carlos Cano, Nasreddine Lagraa, and Pietro Manzoni. 2016. Trust management for vehicular networks: An adversary-oriented overview. *IEEE Access* 4 (2016), 9293–9307.

Tomas B. Klos and Floortje Alkemade. 2005. Trusted intermediating agents in electronic trade networks. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 1249–1250.

Bernd Lahno. 1999. *Olli Lagerspetz: Trust. The Tacit Demand*. Kluwer Academic Publishers, Dordrecht, the Netherlands.

Lei Li and Yan Wang. 2008. A trust vector approach to service-oriented applications. In *Proceedings of the IEEE International Conference on Web Services*. IEEE, 270–277.

Lei Li, Yan Wang, and Vijay Varadharajan. 2009. Fuzzy regression-based trust prediction in service-oriented applications. In *Proceedings of the 6th International Conference on Autonomic and Trusted Computing*. Springer-Verlag, 221–235.

Min Li, Xiaoxun Sun, Hua Wang, Yanchun Zhang, and Zhang Ji. 2011. Privacy-aware access control with trust management in web service. *World Wide Web* 14, 4 (2011), 407–430.

Wenjia Li and Xianshu Zhu. 2014. Recommendation-based trust management in body area networks for mobile healthcare. In *Proceedings of 11th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 515–516.

Xiaoyong Li, Huadong Ma, Feng Zhou, and Wenbin Yao. 2010. T-broker: A trust-aware service brokering scheme for multiple cloud collaborative services. *IEEE Trans. Info. Forensics Secur.* 10, 7 (2010), 1402–1415.

Xiaoyong Li, Feng Zhou, and Xudong Yang. 2012. Scalable feedback aggregating (SFA) overlay for large-scale P2P trust management. *IEEE Trans. Parallel Distrib. Syst.* 23, 10 (2012), 1944–1957.

Guanfeng Liu, Yan Wang, and Mehmet A. Orgun. 2011. Trust transitivity in complex social networks. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*. AAAI, 1222–1229.

Ling Liu and Margaret Loper. 2018. Trust as a service: Building and managing trust in the Internet of Things. In *Proceedings of the IEEE International Symposium on Technologies for Homeland Security*. IEEE, 1–6.

Ling Liu, Margaret Loper, Yusuf Ozkaya, Abdurrahman Yasar, and Emre Yigitoglu. 2016. Machine to machine trust in the IoT era. In *Proceedings of the 18th International Conference on Trust in Agent Societies-Volume 1578*. CEUR-WS.org, 18–29.

Yue Liu, David Bild, Robert Dick, Zhuoqing Mao, and Wallach Dan. 2015. The mason test: A defense against sybil attacks in wireless networks without trusted authorities. *IEEE Trans. Mobile Comput.* 14, 11 (2015), 2376–2391.

Sergio Marti and Hector Garcia-Molina. 2004. Limited reputation sharing in P2P systems. In *Proceedings of the 5th ACM Conference on Electronic Commerce*. ACM, 91–101.

Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the Conference on Internet Measurement Conference*. ACM, 127–140.

Zeinab Movahedi, Zahra Hosseini, Fahimeh Bayan, and Guy Pujolle. 2016. Trust-distortion resistant trust management frameworks on mobile ad hoc networks: A survey. *IEEE Commun. Surveys Tutor.* 18, 2 (2016), 1287–1309.

Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *www.Bitcoin.Org*.

Michele Nitti, Roberto Girau, and Luigi Atzori. 2014. Trustworthiness management in the social Internet of Things. *IEEE Trans. Knowl. Data Eng.* 26, 5 (2014), 1253–1266.

Talal Noor, Quan Sheng, Lina Yao, Schahram Dustdar, and Anne Ngu. 2016. CloudArmor: Supporting reputation-based trust management for cloud services. *IEEE Trans. Parallel Distrib. Syst.* 27, 2 (2016), 367–380.

Talal H. Noor, Quan Z. Sheng, Sherali Zeadally, and Jian Yu. 2013. Trust management of services in cloud environments: Obstacles and solutions. *ACM Comput. Surv.* 46, 1 (2013), 12:1–12:30.

John O'Donovan and Barry Smyth. 2005. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*. ACM, 167–174.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report SIDL-WP-1999-0120. Stanford InfoLab, Palo Alto, CA.

Isaac Pinyol and Jordi Sabater-Mir. 2013. Computational trust and reputation models for open multi-agent systems: A review. *Artif. Intell. Rev.* 40, 1 (2013), 1–25.

Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. 2018. Survey on multi-access edge computing for internet of things realization. *IEEE Commun. Surveys Tutor.* 20, 4 (2018), 2961–2991.

Julian B. Rotter. 1967. A new scale for the measurement of interpersonal trust. *J. Personal.* 35, 4 (1967), 651–665.

Antesar M. Shabut, Keshav P. Dahal, Sanat Kumar Bista, and Irfan U. Awan. 2015. Recommendation-based trust model with an effective defence scheme for MANETs. *IEEE Trans. Mobile Comput.* 14, 10 (2015), 2101–2115.

Wanita Sherchan, Surya Nepal, and Cecile Paris. 2013. A survey of trust in social networks. *ACM Comput. Surv.* 45, 4 (2013), 47:1–47:33.

Shanshan Song, Kai Hwang, Runfang Zhou, and Yu-Kwong Kwok. 2005. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Comput.* 9, 6 (2005), 24–34.

Mudhakar Srivatsa, Li Xiong, and Ling Liu. 2005. TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, 422–431.

Zhiyuan Su, Ling Liu, Mingchu Li, Xinxin Fan, and Yang Zhou. 2013. ServiceTrust: Trust management in service provision networks. In *Proceedings of the IEEE International Conference on Services Computing*. IEEE, 272–279.

Zhiyuan Su, Ling Liu, Mingchu Li, Xinxin Fan, and Yang Zhou. 2015. Reliable and resilient trust management in distributed service provision networks. *ACM Trans. Web* 9, 3 (2015), 14:1–14:37.

Yan Lindsay Sun, Zhu Han, Wei Yu, and K. J. R Liu. 2006. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *Proceedings of the 25th IEEE International Conference on Computer Communications*. IEEE, 1–13.

Girish Suryanarayana and Richard N. Taylor. 2004. *A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications*. Technical Report UCI-ISR-04-6. University of California, Irvine, Irvine, CA.

Shuaishuai Tan, Xiaoping Li, and Qingkuan Dong. 2016. A trust management system for securing data plane of ad-hoc networks. *IEEE Trans. Vehic. Technol.* 65, 9 (2016), 7579–7592.

Shrikant S. Tangade and Sunilkumar S. Manvi. 2013. A survey on attacks, security and trust management solutions in VANETs. In *Proceedings of the 4th International Conference on Computing, Communications and Networking Technologies*. IEEE, 1–6.

Frank E. Walter, Stefano Battiston, and Frank Schweitzer. 2009. Personalised and dynamic trust in social networks. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. ACM, 197–204.

Guojun Wang, Felix Musau, Song Guo, and Muhammad Bashir Abdullahi. 2015. Neighbor similarity trust against sybil attack in P2P E-commerce. *IEEE Trans. Parallel Distrib. Syst.* 26, 3 (2015), 824–833.

Guojun Wang and Jie Wu. 2011. Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Gener. Comput. Syst.* 27, 5 (2011), 529–538.

Jianyong Wang, Yuling Li, Yuhua Liu, and Xiu Jing. 2009. Cluster and recommendation-based multi-granularity trust model in P2P network. In *Proceedings of the International Conference on Computational Intelligence and Security-Volume 02*. IEEE, 380–384.

Yan Wang and Lei Li. 2011. Two-dimensional trust rating aggregations in service-oriented applications. *IEEE Trans. Serv. Comput.* 4, 4 (2011), 257–271.

Li Xiong and Ling Liu. 2003. A reputation-based trust model for peer-to-peer eCommerce communities. In *Proceedings of the 4th ACM Conference on Electronic Commerce*. ACM, 228–229.

Li Xiong and Ling Liu. 2004. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.* 16, 7 (2004), 843–857.

Zheng Yan, Peng Zhang, and Athanasios V. Vasilakos. 2014. A survey on trust management for Internet of Things. *J. Netw. Comput. Appl.* 42 (2014), 120–134.

Zhen Yu, Xuefeng Zheng, Shao-Jie Wang, and Ming-Xiang Li. 2008. A P2P trust model based on preference. In *Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 1–4.

Giorgos Zacharia and Pattie Maes. 2000. Trust management through reputation mechanisms. *Appl. Artific. Intell.* 14, 9 (2000), 881–907.

Chi Zhang, Xiaoyan Zhu, Yang Song, and Yuguang Fang. 2010. A formal study of trust-based routing in wireless ad hoc networks. In *Proceedings of the 29th Conference on Information Communications*. IEEE, 2838–2846.

Haibin Zhang, Yan Wang, Xiuzhen Zhang, and Ee-Peng Lim. 2015. ReputationPro: The efficient approaches to contextual transaction trust computation in E-commerce environments. *ACM Trans. Web* 9, 1 (2015), 2:1–2:49.

Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and privacy on blockchain. *ACM Comput. Surv.* 52, 3 (2019), 51:1–51:34.

Xiuzhen Zhang, Lishan Cui, and Yan Wang. 2014. CommTrust: Computing multi-dimensional trust by mining E-commerce feedback comments. *IEEE Trans. Knowl. Data Eng.* 26, 7 (2014), 1631–1643.

Runfang Zhou and Kai Hwang. 2007. PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.* 18, 4 (2007), 460–473.

Received July 2018; revised September 2019; accepted September 2019