AES

ISSN: 2096-3246
Volume 52, Issue 02, April, 2020

https://doi.org/10.5281/zenodo.12706987

FEDERATED CLOUD DATA ANALYSIS WITH A ROBUST REPUTATION MANAGEMENT

JANGILI RAVI KISHORE, K BALAJI SUNIL CHANDRA, VIJAYA BHASKAR MADGULA

Assistant Professor [1,2,3]

jangiliravi.kishore1@gmail.com, hod.cse@svitatp.ac.in, vijaya.bhaskar2010@gmail.com,

Department of Computer Science and Engineering, Sri Venkateswara Institute of Technology, N.H 44,
Hampapuram, Rapthadu, Anantapuramu, Andhra Pradesh 515722

Keywords:

Federated cloud, virtual network embedding, trust, reputation, multi-tenancy

ABSTRACT

Computing resources are made available for rent under the Infrastructure as a Service (IaaS) model of cloud computing. Users are hesitant to utilise it because they do not have faith in the leased computing resources, even if it provides a cost-effective answer to virtual network needs. There is multi-tenancy, which allows computing resources to be pooled in order to save costs. Security and privacy concerns arise due to the sharing of communication channels and other computing resources. Users remain anonymous, so it's impossible to tell which of them may be a reliable flatmate. The responsibility of assigning reliable co-tenants to the user rests on the Cloud Provider (CP). However, making the most of its resources is what the CP wants. Users' actions have no effect on the maximum co-tenancy it permits. In order to prevent resource sharing in a federated cloud, we provide a strong reputation management method that motivates CPs to identify fraudulent users and allocate resources accordingly. Through analytical and experimental investigation, we demonstrate that the suggested reputation management method is both accurate and efficient.

https://doi.org/10.5281/zenodo.12706987

Introduction

There is multi-tenancy in the Infrastructure as a Service (IaaS) model of cloud computing, which allows users to share computer resources in order to lower the rental cost. Concerns about privacy and security arise due to the sharing of communication channels and other resources. Such issues include, but are not limited to, probing assaults, side-channel attacks, etc. [1, 2], and 3]. Because of these concerns about security, not all consumers are willing to embrace cloud computing. Users might have more faith in Cloud Providers (CPs) if they take their reputation into account while selecting one [4, 5]. One goal of reputation management mechanisms (RMMs) is to make sure that CPs' negative and self-serving actions show up on their reputation [6]. With an emphasis on multi-tenancy, this article presents a strong RMM for the federated cloud. Users rely on the CP to find reliable co-tenants in a cloud with several tenants. In order to motivate CPs to pair up excellent users with good co-tenants, we provide a new reputation management approach in this work. Standing inside the Federated Cloud: A virtual network request may be met by more than one cloud provider in a federated cloud, which is built by contributions from numerous cloud providers. A CP's credibility could take a hit in a federated cloud since it shares resources with other CPs. This is because a virtual network request might include resources from many CPs. Issue with a virtual CPs' physical resources could be the starting point for the network. We may utilise this kind of input to assess CPs' reputation: One sort of feedback that is difficult to implement is input on other CPs. This is because CPs need to communicate information about their own resources. 2) Consumer Perceptions of CPs: It's easier to get their opinions. However, this data can be inaccurate or even harmful. It will be difficult for customers to identify the exact CP accountable for a defect in a virtual network as the resources may be held by many CPs. 3) User-centric CP input: This is the most accessible kind of user-centric feedback. One way CPs may tell whether their clients are malicious is by keeping tabs on their activity.In this research, we assess the CPs' credibility using the third kind of feedback. Such input might be reported incorrectly. In this article, we provide a system that CPs may use to ensure they are reporting accurate client feedback. Multi-tenancy and CP's reputation: Cloud computing RMMs that are now available collect user reviews and aggregate them to get CP reputation scores. In addition,As far as the CPs' performance is concerned, it attempts to distinguish between fair and unfair user comments [7]. Distinguishing between physical network problems and CP-initiated disruptions is another important feature. Hence, CPs' reputations are unaffected by flaws, as they are considered to be beyond of their control [8]. Unlike other RMMs, our proposed RMM in this research prioritises many tenants. The major worry of users is sharing computing resources with others, especially when considering the possibility of harmful co-tenants. Keep in mind that, 1) A user's roommates remain anonymous. This means that users have no control over who their computing resources are shared with.

2) The CP's ability to pair users with suitable roommates is upon the user. Users will have greater faith in a CP that distinguishes itself from the crowd if the latter emphasises co-tenancy. in between legitimate and bad users, and whether it prevents them from sharing resources. Therefore, the capacity and willingness to distinguish between legitimate and malevolent users is the primary factor that determines a CP's reputation. In comparison to another CP that does differentiate, a CP that does not should be regarded with low esteem. We provide an RMM in this study that takes into account the CP's capacity and desire to differentiate its consumers in this way. Making the most of its resources is what the CP wants to do. Therefore, it permits maximal co-tenancy independent of user behaviour. In this research, we focus on federated clouds, where the physical network is a linked graph that is supplied by numerous stakeholders. Virtual network requests in a federated cloud are routed to specific physical network segments that are controlled by different CPs. So, in order to meet a need for a virtual network, the CPs may function together. Keep in mind that 1) when the CPs work together to fulfil requests from the virtual network, it is possible for one CP, CP1, to fail to distinguish between legitimate and malicious users, while another CP, CP2, does the exact opposite. If CP1 and CP2 work together, CP2 may wind up letting a good user move in with a bad user—something it definitely doesn't want to happen. Thus, a CP's actions have consequences for those who work with it. As a result, we presume the following: 1) The knowledge on multi-tenancy is shared by CPs. It is also unable to change this data. 2) Nevertheless, there is a chance that they may portray user conduct inaccurately.In a nutshell, here's how our RMM operates:1) First, every CP should identify good users from bad ones and provide resources

accordingly, such that the following is true: a) It shouldn't let a bad user become a co-tenant with a good user. b) Malevolent users could be able to exchange resources with one other. The CPs then proceed to discuss multi-tenancies.
Thirdly, the RMM receives information on user activity from each CP. 4) The reputation of a CP is enhanced when the reputations of the users in each group of multi-tenant users remain stable, meaning that their reputations either go up or down. If the changes in the users' reputations are comparable, it suggests the CPs did a decent job of separating the good users from the bad ones and preventing them from sharing resources. The following is the rationale for users to inflate their reported conduct in the aforementioned RMM model:

- A CP's reputation improves when there is consistency in the reputation changes of each set of multi-tenant users.
- In order to ensure consistency in the reputation modifications of each group of multi-tenant users, it is in its interest to report user reputations incorrectly. Our CP behavioural models are as follows: 1) Reasonable CP: Users' actual conduct is consistently reported by a logical CP. 2) Irrational CP: This kind of CP falsely claims that a collection of users with different tenants are all good or malicious, regardless of how they really behave. 3) Opportunistic CP: This kind of CP claims that a group of users who have several tenants are excellent users if the majority of those tenants are decent, and it claims the reverse otherwise. We demonstrate that, when these three classes of CPs are present, The first is robustness. We examine the RMM's resilience. Based on the work of [9], we apply the concept of resilience to a normative framework. This robustness concept presupposes that certain agents in a normative multi-agent system constantly act contrary to the rules. The multi-agent system is resilient if it functions correctly, considering the proportion of non-compliant agents, as long as other agents stay compliant. A related concept of robustness is used in this work. We demonstrate the distribution of rational and irrational agents, where the former constitutes the majority, and for which the suggested RMM continues to operate. 2) CPs' Reputation: We demonstrate that CPs' reputations improve when they distinguish between benign and malevolent users and deny the former access to shared resources. 3. User Reputation: We demonstrate that, in comparison to malevolent users, good users get a superior reputation. Section I.1.1 1.1 Structure The article is structured as follows: Section 2 covers the literature review. The suggested RMM is detailed in Section 3. An examination of the proposed RMM is presented in section 4. The experimental evaluation of the suggested RMM is presented in Section 5. In part 6, we wrap up the paper.

2 RELATED WORK

Three study issues come together in RMMs research under the IaaS paradigm of cloud computing: (a) virtual networks, (b) reputation management in cloud computing, and (c) online reputation management. set up. Here we provide a high-level overview of where certain areas of study now stand. Managing one's online reputation To detect unfair feedback, there are two main methods, as stated in [10]. Only feedback that is unjust may be used to decide via the endogenous processes [11] [12]. The statistical features of the feedbacks form the basis of these systems. These algorithms often make the assumption that most feedbacks are fair based on the history of feedbacks. To ascertain the fairness or unfairness of feedback, the exogenous mechanisms use data from outside sources. The trustworthiness of the purchasers is one example of this kind of data. In order to determine how trustworthy a suggestion is, [13] use a personalised similarity metric. Within this system, the reliability of an evaluator is decided by its contemporaries who have dealt with it. In [14], a similar method is used to determine the reliability. The legitimacy of the feedback is determined by using the service trust as the parameter [15]. However, when dealing with service providers that are in competition, this technique might be prone to sending unfair comments about those providers. In [16], the weighted majority algorithm (WMA) is suggested, which uses weights to enhance the relative weight of successful advisers and lower the relative weight of failed ones. [11] finds the buyer agent's closest neighbours according to how similar their preferences are. They are considered to have similar preferences if they have a high number of comparable ratings for sellers. Unfair rating is detected via cluster filtering once the buyer agent's closest neighbours have been identified. To further eliminate unjust ratings, [17] builds on reputation management techniques established in [18] by including iterative filtering. In order to determine the seller's reputation, [12] has taken into account the reputations of the purchasers. TRAVOS is a concept for agent-based virtual organisations that focuses on trust and reputation [19]. Based on the quantity of comparable past advice, both correct and wrong, this approach first calculates the accuracy of the present reputation advise. The next step is that it calibrates reputational recommendations based on its precision. This task's objective is to lessen the impact of bad advise. But this model presupposes constant behaviour on the part of selling agents, which isn't always the case. A reputation management system incentive design algorithm is one of many available. Agents who provide more useful input are rewarded more handsomely according to the incentive system defined by [20] utilising a payment game. One way to incentivize honest input is via a payment plan that was suggested in [21]. When an agent's feedback on a target agent is consistent with subsequent feedback on the same target agent, the agent receives payment under this method. The incentive model put out in reference [22] is predicated on Problems faced by inmates. Agents who provide honest feedback under this approach are more useful. A penalising system was suggested in order to collect honest feedback [23]. In this arrangement, one party reports the other in every transaction. Both parties are subject to penalties in the event that the reports in the various transactions are inconsistent. Investigated in [24] was the viability of a payment scheme for online auction platforms to solicit honest feedback. Reputation management is presented in [25] using an iterative probabilistic approach. Weights of the edges and vertices embedded in the virtual network. In a physical network G = (N, E, W) (where N, E, and C are nodes, edges, and weights of the communication channels, respectively), the vertex weight indicates the portion of CPU or compute utilisation, while the edge weights indicate the bandwidth of the channels. the edges and vertices) in a way that meets certain criteria (a) The requested weights of the vertices and edges must be equal to or greater than what is actually mapped into the physical network. (b) There must be exactly one mapped vertex (or set of vertices) for every requested edge, and exactly one mapped path (or set of vertices) against each requested vertex. Always keep in mind that many VNEs might share the same vertex and edge. In addition to the heristics described in [30], [31], and approximation methods developed in [32], it is known that the VNE issue is NP-complete [27], [28], and [29].

Because VNE makes it possible to rent a computing infrastructure, it is a cost-effective option. However, services dependent on the physical network are interrupted when it is unavailable owing to problems or maintenance. When we extend the VNE issue to include the allocation of a set of extra resources in the physical network, we get the survivable virtual network embedding (SVNE) [33] problem. The physical resources are guaranteed to be available continuously with the help of these additional resources. One or more vertex and edge failures, or both, may occur in a physical network. According to [34], maintenance accounts for 20% of failures, router issues for 53%, and single link failures for 70%. According to [35], the failure of a connection is ten times more often than that of a node. The SVNE challenges may be addressed in two ways [36](a)Preventative measures: These include making sure there are backups or other resources available in case the physical network fails in any way, and (b)Reactive measures: These involve identifying VNEs with the resources that are accessible after the physical network has collapsed. Neither of these methods ensures that data will not be lost, but the first one also addresses the optimisation issue of determining the most cost-effective backups. In order to address the issue of single link failure with shared backups, a reactive approach was introduced in [37] to solve SVNE difficulties. Instead of backing up each main connection, the authors of [38] suggested keeping pathways as backups. Reactive solutions were investigated in [39] for SVNEs that adhere to certain metrics for quality assurance. In order to improve the request with more redundant nodes, single node failure SVNE was examined in [40]. Graph decomposition was used to examine SVNEs with a single node failure in [41]. Researchers in [42] looked at the effects of a single regional failure (the failure of a single linked subgraph of the physical network) and in [43] they looked at the same issue with geographical limitations (backup or replacement conditions). The SNVE issue was investigated for failures at the edges as well as the nodes [44]. In addition, it takes backup minimization into account. Using multi-agent systems, [45] suggested a distributed approach to solving the SVNE challenge. Assuming n=1 as the number of vertices in the SVNE issue, an online solution was suggested in [46] with a competitive ratio. the actual network device. When the physical network makes it impossible to differentiate between live and backup mapping, this becomes an embedding issue [47]. Therefore, it incorporates a series of redundant pathways to ensure that in the event of an edge failure, there would be a backup. Both the NP-completeness of the issue and the provision of heuristics are shown. Keeping a positive reputation in the cloud In order to differentiate between legitimate and unjust criticisms of cloud service providers, [48] put forth a multi-pronged trust management methodology. Cloud service providers may be assessed by users based on a number of criteria, as suggested in [49], which is a multi-faceted reputation management paradigm. A method for gauging cloud providers' reliability was suggested in [50] by looking at how often they broke the SLA. [51] [7] suggested a method for distinguishing between malicious and unfair trust feedback in the cloud. To lessen the blow to cloud providers' credibility in the event of a system outage, [8] suggested a reputation management strategy. The privacy and security concerns raised by multi-tenancy have been investigated in [1, 2], and [3]. For a more comprehensive overview of cloud computing trust mechanisms,

REPUTATIONMANAGEMENT MECHANISM

Unofficial framework This is the RMM informally:

1) Both the quantity of CPs and the number of users are limited. All users' virtual network requests are expected to be hosted by each CP. The first kind of CP is the logical one; the second is the irrational kind; and the third is the opportunistic kind. Both good and bad users exist; the former does not compromise security or privacy, while the latter does just the opposite. Security issues, such as side channel attacks, may be caused by a malicious co-tenant [53]. the assault dependent on the network's physical implementation), denial-of-service attack [54], and network probing attack [55] (an attack to determine the network's topology). We take it as read that a CP may detect fraudulent or benign user activity if it hosts a user.

2) The first step is for each CP to classify users as excellent or bad.

b) It allots people virtual stuff.

b) They are multi-tenant, meaning that users are divided into groups and within each group, all users share resources.

d) The multi-tenancy information is announced to the RMM by each CP, who then broadcasts partitions over the users.

Afterwards, CPs keep tabs on user activity and notify the RMM of any suspicious activity. Any CP may cast a vote for any user, favourable or bad. It is expected that the RMM will be able to communicate with the individual CPs via the federated cloud architecture. On a regular basis, CPs inform the RMM of their thoughts (a positive or negative vote on the users). According to the CP, a harmful user is indicated by a negative vote; otherwise, the individual is deemed good. Here is the model of user-CP interaction that we use:

e) Users' actions, whether good or bad, are deduced from the events they produce at each stage.

f) In each stage, after the observation of user-generated events, each CP records the following user behaviour:

The CP views the user favourably when they provide a positive vote for them.

• A negative vote: This means the CP thinks the user is bad.

g) The RMM determines a user's reputation in the following way after receiving their votes:

i) A user's reputation will improve if the number of positive votes exceeds the number of negative votes.

ii) A user's reputation will take a hit if the number of positive votes is lower than the number of negative votes.

iii) A user's reputation will not change if the amount of positive votes and negative votes are equal.

After the users' reputations are updated in each stage, the RMM changes the CPs' reputations in the following way:

a) in the case of each set of users who are tenants in more than one building, is the reputation of all users to rise or If everyone's reputation takes a hit, the CP's will take a boost. b) A CP's reputation takes a hit if, among a group of users who are also tenants, certain users' reputations take a hit while the remainder of the users' reputations take a hit. Take note that, for a CP to be reputable, it must correctly segment its users. This means dividing them into groups, with each group consisting of multiple tenants who share resources. A good user should be grouped with similarly minded individuals, while a bad user should be grouped with similarly minded individuals. When users' reputations alter, it shows how accurate a CP's user segmentation was. Reputation of users in a group will rise if it has only assigned good users to that group, and fall if it has

ISSN: 2096-3246
Volume 52, Issue 01, Jan, 2020

https://doi.org/10.5281/zenodo.12706987

assigned only bad users to that group (because other CPs vote for good and bad users). Any change in the reputation of all members of a group has the effect of raising or lowering the reputation of the CP. While some users' reputations will rise and others' will fall, this is only possible if the CP has treated both good and bad users equally. Therefore, when the reputations of certain group members rise and those of other members fall, the reputation of a CP falls. 3.2 Hypothesis U = {u1,..., um} represents a collection of users, whereas C = {c1,..., cn} represents a set of n cloud providers. Our presumption is as follows: • The host, or the CP that holds its virtual network, is able to identify any user misbehaviour. R(ui) ∈ [0, 1] represents the users' initial reputation. A user's reputation may be as high as 1, with 0 being the lowest possible reputation and 1 the highest possible. The starting reputation values of all users are known to all CPs. At the outset, all CPs have the same reputation. A positive real number, R(ci) ∈ R>0, is used to represent the CPs' reputation. The host should notify the RMM if the user exhibits inappropriate conduct. The users are divided into k partitions, indicated as π = {π1,..., πk}, by each CP. In the physical network, the group of users in each πi share at least one vertex or edge. We presume that the partition π, which each CP reports to the RMM, cannot be changed, meaning that the report on the partition is always accurate. • User conduct may be inaccurately reported by a CP. A function that links physical resources to virtual network requests is defined first; this is known as virtual network embedding. First Definition: (Embedding a virtual network) An embedding f from a virtual network may be used to transform a virtual network request GR = (V R, ER) into a physical network, given a vertex weight function WR 1: V R 7→ R>0 and an edge weight function WR 2: ER 7→ R>0. Vertex weight function with G = (V, E) Such that the following is true for the weight function W1: V7→ R>0 and the edge weight function W2: E7→ R>0: 1) A subset S ⊂ V exists for every vertex v1 ∈ V R such that f(v1) = S. 2) For every vertex v1 ∈ V R, The constraints WR 1 P (v1) < v2∈f(v1) W1(v2) hold. 3) An edge e1 in ER is considered linked if and only if there is a subset S in E such that f(v1) = S and |S| > 1. For every edge e1 in ER, the weight of WR 2(e1) is less than or equal to P e2 in f(e1) W1(e2). Keep in mind that users do not have access to exclusive computing resources via the aforementioned virtual network embedding procedure; instead, they share these resources. Based on the distribution of available computing resources, we categorise the users. If two users share a resource, then they are considered to be in the same group according to this partition. Second Definition: (User Partition) We shall refer to the set of users hosted by a CP ci as U(ci) ⊂ U. For any set πi that contains virtual network requests G1 = (V 1, E1) and G2 = (V 2, E2), one of the following must be true: either f(V 1) ∩ f(V 2) 6= Ø or f(E1) ∩ f(E2) 6= Ø. This is represented as π = {π1,.., πk}. For any i and j, the intersection of πi and πj is empty. Here is how we describe the user's reputation moving forward: Third Definition: (Reputation of the User) Figure 2 is referenced. Imagine a circle C with a radius r and a centre cu. The users' circle will be referred to as C. The x-axis and the y-axis are represented by lines Lh and Lv, respectively. As shown in Figure 2, two points on the user's circle's diameter, a and b, are given to each user. The following is true for the users' reputation, which is supplied by the angle ∦(a, cu, b): At the outset, for every user, ∦(a, cu, o) = Ø(b, cu, o) > 0, with o being the intersection point of the circumferences of C and Lv. • The range of each user's reputation is (a, cu, o)/2) − (b, cu, o)//2. ∠(a, cu, o) or ∠(b, cu, o) is limited to the interval [0◦, 90◦]. As a result, the reputation of every user falls somewhere between -1 and 1. • The user's points are a and b. The procedure for editing a user's reputation is subsequently detailed. We modify a user's standing by referring to CP's report of its actions. Fourth Definition (Reputational Shift:) Figure 2 is referenced. When a host CP reports good things about a user, their reputation changes.

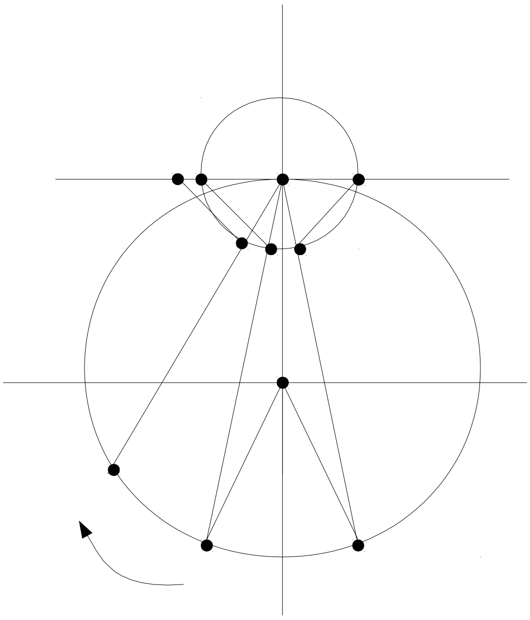
Fig. 2. The credibility of content providers and audiences. or criticisms levelled against it. Either a positive or negative vote may be cast by a CP. Find a tiny positive rational number greater than zero. The CP's

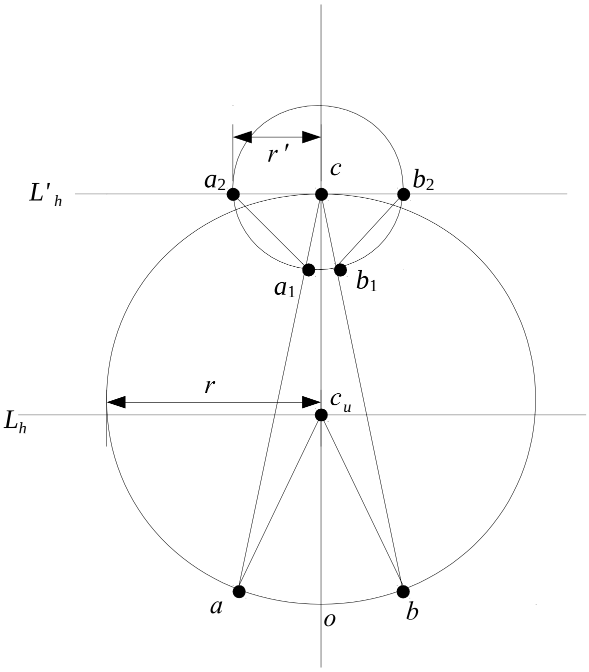
ISSN: 2096-3246
Volume 52, Issue 02, April, 2020

https://doi.org/10.5281/zenodo.12706987

positive vote increases the angle ∠(a, cp, o) by, whereas a negative vote increases the angle ∠(b, cp, o) by. Therefore, a user's reputation is enhanced when it gets good votes and diminished when it receives negative votes. We will now use figure 2 to determine the CP's repute. Fifth Definition: (CP's standing:) Figure 2 is referenced. Every CP ci is given a circle Ci whose centre is cp (the place where the lines Lv and L 0 h cross) and whose radius is r 0 based on the user's circle C. Circumference of the circle Ci represents the CP ci's repute. A CP circle is what this is known as. We will now outline the relationships between a CP's and a user's reputations.

Definition 6. (Association between CP’s and user’s reputation) (Refer to Figure 3) If a CP ci has hosted the user uj with user’s points a and b then we assign twolines to the CP ci as follows: • Lines are (a1, a2) and (b1, b2). • Find the point a1 (b1) on the circumference of the CP circle Ci as the point of intersection between the line (a, cp) ((b, cp)) and the circumference of the CP circle Ci
. • These lines (a1, a2) and (b1, b2) will act as theassociation between the CP’s and the user’s reputation. Using this association we will change a CP’s reputation from the change in a user’s reputation (discussed in next definition). From the definition of relation between aCP’s reputation and the user’s reputation, next we define the change in a



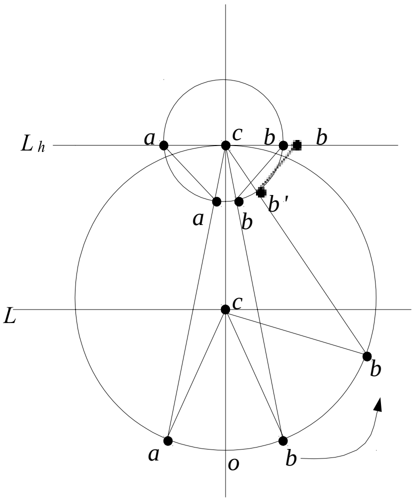


Relative importance of the user's and the CP's reputations (Fig. 3). CP's standing as a result of shifts in user reputations. Meaning 7. (Reputational shift for a single user at CP) Figure 4 is referenced. As the user uj's reputation changes, the change in the CP ci's reputation is denoted by δ(uj, ci). What follows is the calculation of δ(uj, ci): The angle ∠(a, cu, o) is raised to angle ∠(a 0, cu, o) when uj has gotten affirmative votes. The place where the line (a 0, cp) and the circumference of the CP circle of ci meet is denoted as a 0 1. The line L 0 h crosses the line 0 2 at a 0 2, and we construct a parallel line from a 0 1 w.r.t. the line (a1, a2). The line segment (a 0 2, a2) has a length denoted as δ(uj, ci). Likewise, if uj has been voted against, δ(uj, ci) is determined. As shown in Figure 4, the reputation of ci may alter when a user's reputation changes. The user's behaviour changes in response to good and negative votes, as shown in Figure 4 (a) and (b), respectively. Keep in mind that the user's CP reputation grows in direct proportion to the number of good (or negative) votes they get. The eighth definition concerns the shift in CP's standing with regard to user partitioning. Figure 4 is referenced. Allow CP ci to divide the users into the following groups: π is equal to the set of all actual values from 1 to πk. Here is how we determine the change in the CP's repute for every set πi: In this context, S denotes people who have gotten positive votes and T denotes users who have received negative votes, where S **8** T = πi. Every time uj is an element of πi, we determine δ(uj, ci).

ISSN: 2096-3246
Volume 52, Issue 01, Jan, 2020

AES
https://doi.org/10.5281/zenodo.12706987

Figure 4 shows how CP's reputation might be affected by hosting only one user. | P $u_j \in S$ $\delta(u_j, c_i) - $ P $u_j \in T$ $\delta(u_j, c_i)$| is the change in ci's reputation as a result of the change in the reputation of the users in πi, represented as $\Delta(\pi i, ui)$. And lastly, $\pi i \in \pi$ $\Delta(\pi i, ci)$ represents the overall change in CP ci P's repute. Keeping with the previous process of modifying the CP's reputation based on its user partition, we note: Prove that the following is true for any two partitions $\pi = (\pi 1,.., \pi k)$ and $\pi 0 = (\pi 0 1,.., \pi 0 k)$: For every set ($\pi i \in \pi$), the majority of users who have earned positive votes (or negative votes) are shown. In any set ($\pi 0 i \in \pi 0$), the quantity of users who have been voted for positively is about equal to the quantity of users who have been voted against. Compared to partition 0, the increase in the CP's repute from partition π is greater. The kinds of CPs are now defined. A CP might be reasonable, illogical, or motivated by opportunity. No. 9: (Adversary) Here are three categories of CP behaviours: • Rational CP: These CPs accurately distinguish between trustworthy and malicious users. In addition, they reveal users' actual conduct; for example, it gives a negative vote to a user whose behaviour is inappropriate, and a positive vote to a user whose behaviour is appropriate. • Irrational CP: These CPs falsely categorise users as either good or evil. In addition, they exaggerate the positive or negative votes cast by users inside a division, rather than reflecting their actual conduct. • Opportunistic CP: These CPs also make the mistake of classifying users as either good or evil. What follows is an inaccurate depiction of the users' actual actions: Assume that $\pi = \{\pi 1,..., \pi k\}$, and let the opportunistic CP to divide the users into k sets. It will report a good vote for all users in set πi if the majority of users in set πi act properly. If most people in πi act inappropriately, it will record a bad vote for all users in πi. 4 a review It is estimated in Lemma 1 how much a change in a user's reputation affects a CP's reputation. Theorem 1. If user uj with user points A and B is hosted by CP ci and receives positive votes, then the following holds: $\delta(uj, ci) = r \prod \cos \theta 2 \prod (1 - \sin \theta 1)\cos \theta 1 - r + r \prod \sin (\theta 2)$ (1), where r is the initial reputation of ci, 2θ1 and 2θ2 are the positive vote angles of uj before and after it receives the positive votes, respectively.



Proof. The scenario is illustrated in figure 5. The share angle changes from 4θ1 to 4θ2 as the share points are changed from (A, B) to (A1, B1). Note that the angles ∠A1, cp, o and ∠A, cp, o are θ2 and θ1 respectively. The share linesare changed from (z, a) to (z1, a1). Hence the change in the radius of the buyer’s circle is the length of the line segment zz1. We calculate the length of the line segmentzz1 as follows: Note that, ∠cp, z1, a1 = ∠cp, z, a = θ. In the

https://doi.org/10.5281/zenodo.12706987

ISSN: 2096-3246
Volume 52, Issue 02, April, 2020

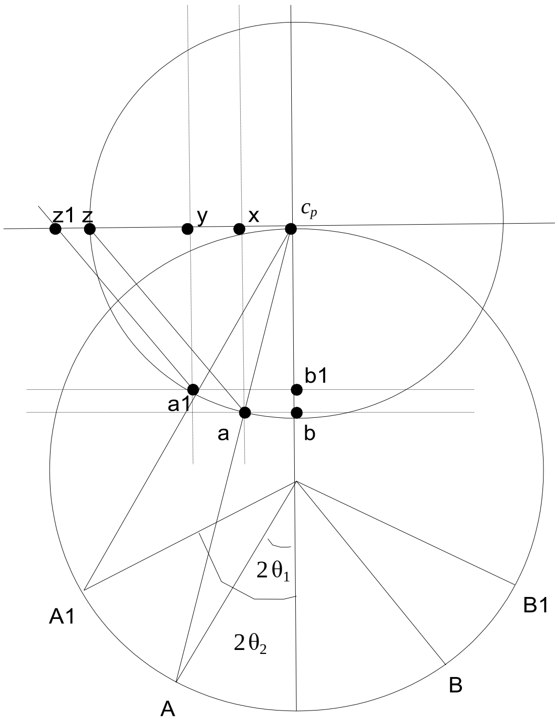
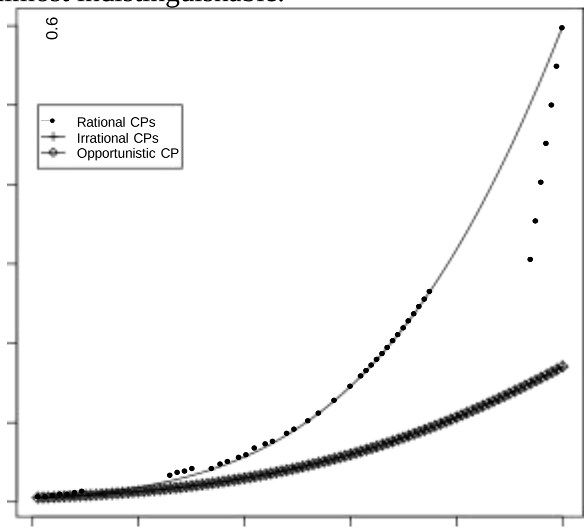


Fig. 5. The triangles Δabcp and Δacpz, where ab acp = sin θ1, are proven in Lemma 1. The equation (3) may be rewritten as xcp = ab = r * sin{θ1. The equation xz = r − r ∏ sin θ1 (4) stands. In addition, cpb acp = cos θ1 (5) The equation for xa is given by cpb = r cos θ1 (6). The equation Δ(xaz) tan θ = xa xz = r cos θ1 r − r ∏ sin θ1 (in the triangle) may be simplified to cos θ1 1 − sin θ1 (7). In the triangle, the angle Δ(a1, b1, cp) cos θ2 is equal to ya1 r, where ya1 is equal to r all multiplied by cos θ2 (8) times tan θ, which is equal to r all. angle θ2 The equation yz1 = cos ξ1 1 − sin θ1 (9). The equation yz1 = r/cos θ2 + (1 - sin θ1)/cos θ1 (10). Tan(90 − θ2) = r ∏ cos θ2 ycp (11) in the triangle Δ(a1, y, cp). You may write it as ycp = r * cos θ2 tan (90 − θ2). Thus, zy is equal to r − r ∏ cos θ2 tan (90 − θ2). (12) the equation zz1 = yz1 − yz (13) The first power, zz1, is equal to the product of r, cos θ2, and (1 - sin θ1). The expression is (cos θ1 − r + r ∏ cos θ2 tan (90 − θ2)) The equation zz1 = r * cos θ2 * (1 - sin) sine of θ1 minus r plus r squared here is the equation: cos θ2 multiplied by sin (θ2) "zz1 = r ∏ cos (θ2)" (1 - sin θ1) cos(θ2) equation (14): cos θ1 − r + r ∏ sin (θ2) Rational CPs are more highly regarded than irrational and opportunistic CPs, as shown by Lemma 1 in Theorem 1. Assumption 1. Assume that there are m users (u1,... um) and n CPs (c1,..., cn). • There are three types of CPs: K1 rational, K2 irrational, and K3 majority irrational. There are two types of users: good users and bad users. The number of buckets is k. The upper limit is represented by kmax. Each bucket stands for a group of renters who are also users. The capacity of a resource is the highest possible number of people that can share it. Assumption: k*kmax >= m. The rational CPs will be renowned more highly than the illogical or opportunistic CPs if K1 is greater than m − 2 2 K2 − K3. The evidence. This setup is being used by the CPs: Unreasonable CPs: Z-1 buckets are available. With m being the total number of users, there are kmax users in each bucket. • (Comprehensive CPs:) The number of buckets is z 2 = z 12 + z 22. It stores positive CPs in z-12 buckets and bad CPs in z-22 buckets. The maximum value of z at 12 and 22 was found to be l1 and l2, respectively. • (CPs that are substantially irrational:) Three buckets are available. With m being the total number of users, there are kmax users in each bucket. At each given point in time, for every decent user, the following is true: • Every reasonable CP gives a yes vote. As a result, it receives K1 votes in favour. • In the bucket selected by the majority of irrational CPs, there will be l1∏zmax m good CPs and l2∏zmax m harmful users. The estimated number of irrational CPs providing negative and positive feedbacks is K2/2 and K2/2, respectively. The outcome is determined

by whether l1∏zmax m > l2∏zmax m; if not, the outcome is negative. The following is true at every point in time for each malevolent user: • All reasonable CPs cast negative votes. This results in K1 votes against it. In the bucket selected by the majority of irrational CPs, there are projected to be l1∏zmax m good CPs and l2∏zmax m harmful users. The predicted number of irrational CPs providing negative and positive feedbacks is K2/2 and K2/2, respectively. The outcome is determined by whether l1∏zmax m > l2∏zmax m; if not, the outcome is negative. Let us pretend that l1 is greater than l2. In a group of excellent users, the predicted positive and negative votes for rational CP are: The sum of p–votes equals kmax.The rational CPs z}|{ K1 + irrational CPs z}|{ K2 2 + opportunistic CPs z}|{ K3] become 3. The sum of all votes, minus one, equals kmax.|{ K2 2][Irrational CPs z} (15) Here are the predicted favourable and negative votes for rational CP in a group of malevolent users: "p - votes" equals kmax|{ K2 2][Irrational CPs z} The sum of all votes, minus one, equals kmax.The rational CPs z}|{ K1 + irrational CPs z}|{ K2 2 + opportunistic CPs z}|{ K3] become 3. sixteen (16) P − Change = kmax[K1 + K3] is the equation that represents the change in the CP's reputation from the buckets with excellent users. (17) Hence, the reduction in the CP's credibility caused by the buckets containing malevolent users may be expressed as: n − Change = kmax[K1 + K3]. (18) Consequently, its repute may be expressed as: z 2 ∏ (kmax[K1 + K3]) (19) ∏[r ∏ cos θ2 ∏ (1 − sin θ1) cos θ1 − r + r ∏ sin (θ2)]. (20) The anticipated number of good users is kmaxl1 m in every irrational CP bucket, while the expected number of malevolent users is kmaxl2 m. In any given bucket for excellent users, the predicted number of positive and negative votes for an irrational CP is p − votes = kmaxl1 m. |{ K1 + Irrational CPs z}|{ K2 2 +] Potentially lucrative CPs z}|{ K3] The sum of all votes cast, minus one, is equal to kmaxl1 m. |{ K2 2][Irrational CPs z} (21) The situation is same for malevolent users: The sum of P— votes is equal to kmaxl2 m. |{ K2 2][Irrational CPs z} A total of n votes equals k multiplied by the square of m. The rational CPs z}|{ K1 + irrational CPs z}|{ K2 2 + opportunistic CPs z}|{ K3] become 3. (22) Therefore, the illogical CP receives the following amount of affirmative votes for every bucket: p − votes equals kmaxl2 m [K2 2] plus kmaxl1 m [K1 + K2 2 + K3] The number of votes is equal to the sum of kmaxl1 multiplied by m [K2 2] plus kmaxl2 multiplied by m**2. All of the following: [K1 + K2 2 + K3] (23) So, its repute is: z 1 ∏ [kmax(l1 − l2) m [K2 2] + kmax(l1 − l2) m. The equation is written as [[K1 + K2 + K3]] ∏[r ∏ cos θ2 ∏ (1 − sin θ1) cos θ1 − r + r ∏ sin (δ2)]. (24) The opportunistic CPs are also included in Equation 24. Keep in mind that, z 1 ∏ [kmax(l1 − l2) m [K2 2] + kmax(l1 − l2) m [K1 + K2 2 + K3]] is correct. = z 1 * kmax(l1 − l2) m [[K2 2] + [K1 + K2 2 + K3]] z1 multiplied by the square of the maximum value between l1 and l2 plus or minus the sum of K1, k2, and K3 (25) The inequality z2 ∏ (kmax[K1 + K3]) > z1 ∏ kmax(l1 − l2) m [K1 + k2 + K3] must be shown. > z1 ∏ (l1 → l2) m, where K1 and K3 are equations. [K1 plus k2 plus K3] (26) Please take note that z 2 is greater than z 1. The rational CP requires just one additional bucket to distinguish between good and bad users, unlike the illogical or opportunistic CP, which requires more than that. So, we have to make sure we meet these requirements: <[K1 + K3] > (m) in the direction of (l1 → l2) (K1, K2, and K3) After adding K2 and K3, the result is K3 K1[1 → (l1 − l2) m] > (l1 − l2) m, which is more than the product of (l1 − l2) and (K1[1 → (l1 → l2) m]. M = K2 − (1 − (l1 − l2)).The equation K3 K1 > m may be rewritten as m − (l1 → l2) (l1 → l2) m. Assuming there is a malevolent user, K1 > m − 2 m − (m − 2), and K2 → K3 > l1.The equation (28) states that K1 > m − 2 2 K2 − K3, where K2 is different from K3. We conclude by demonstrating that a trustworthy user will have a higher reputation than an evil one. Second Theorem: A trustworthy user will have a higher reputation than an evil one. The evidence. In each phase, the good user's reputation is changed by [K1 + K3] using equation 15. (29) The reputation of the bad actors is altered by subtracting K1 from K3 using equation 30. (30) Therefore, the reputation of trustworthy users improves while that of dishonest people declines. 5 Findings from Experiment Here is how we model the federated cloud: Thirty CPs are available. One kind of CP is an opportunistic CP, while another is an illogical CP. There are two hundred users. A user's intentions might be good or bad. • It is presumed that all users are hosted by each CP. Users are divided into ten groups, with each group representing a set of co-tenants, by each CP. Reasonable CPs would never combine a trustworthy user alongside an evildoer. On the other hand, CPs that are illogical and selfish tend to group people at random. Here is how we model the RMM and how CPs and users communicate: 1) Each CP starts by dividing the users into different categories. It is taken for granted that users who are part of the same group share the available computing resources. 2) Every CP keeps an eye on how the users are behaving at all times. b) Additionally, it notifies the RMM of their actions as votes, whether good or negative. c) The RMM uses Definition 3 to determine a user's reputation once it has received votes from the CPs. d) The RMM determines the CPs' reputations after receiving their votes by using Definition 8 and Lemma 2. Figures 6 and 7 show the results of the simulation with ten opportunistic CPs, ten irrational CPs, and one rational CP. An equal amount of people utilise it for good and harm. Pictured in Figure 6 are the typical reputations of the opportunistic, illogical, and reasonable CPs. This proves beyond a reasonable doubt that logical CPs are more reputable than their illogical and opportunistic counterparts.

Figure 7 shows the reputation of the good and the malicious users. It clearly shows that, the good users get better reputation than the malicious users. Note that, the differencebetween the mean reputation of the irrational and the opportunistic CPs is very small and almost indistinguishable.



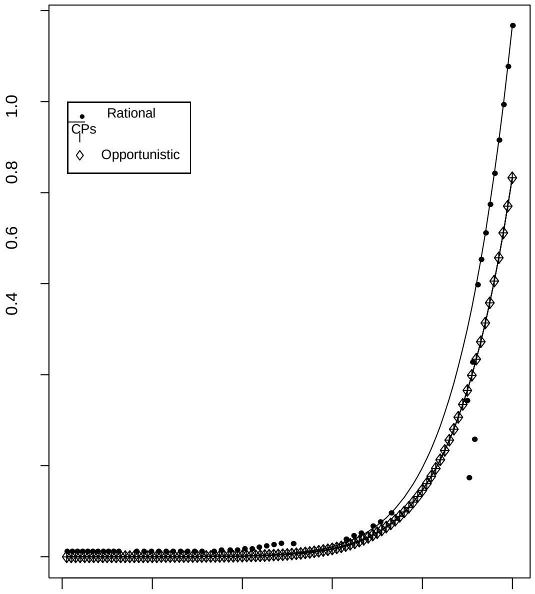


Fig. 6. Ten CPs are opportunistic, ten are illogical, and ten are reasonable. An equal amount of people utilise it for good and harm. The plot reveals the CPs' repute. The CPs' repute is shown on the Y-axis. The next two experiments involve an increase in the number of rational agents. The results with 12 logical, 8 irrational, and 10 opportunistic CPs are shown in Figures 8 and 9. An equal amount of people utilise it for good and harm. Additionally, it demonstrates that illogical and opportunistic CPs have worse reputations than reasonable ones, and that good users have higher reputations than bad ones. With 15 rational, 5 irrational, and 10 opportunistic CPs with an equal number of good and bad users, we get the same results (Figures 10 and 11). The data, however, reveals that as the quantity of rational CPs grows, the reputational gap between them and other CP types widens. Reputational differences between rational and irrational (or opportunistic) CPs become insignificant when the number of rational actors is reduced below 10. Therefore, we state in this simulation that the RMM is still usable if there are rational CPs that make up at least 1/3 of the total CP population.

https://doi.org/10.5281/zenodo.12706987

6 CONCLUSION:

Co-tenancy makes cloud computing affordable but it also introduces new risk from malicious co-tenants. A user de

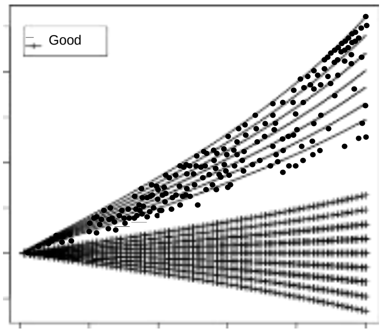


Fig. 7. There are 10 rational, 10 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputa- tion of the Users

Fig. 8. There are 12 rational, 8 irrational and 10 opportunisticCPs. There are equal number of good and bad users. Plot shows the reputation of the CPs.

pends on the CP for allocation of safe co-tenants. Our objec-tive in this paper is to develop a RMM that encourages CPs to make correct segmentation among good and malicious users, i.e., a good user gets only other good users as co- tenants. The existing RMMs for cloud computing do not consider this criteria to evaluate reputation of the CPs. The

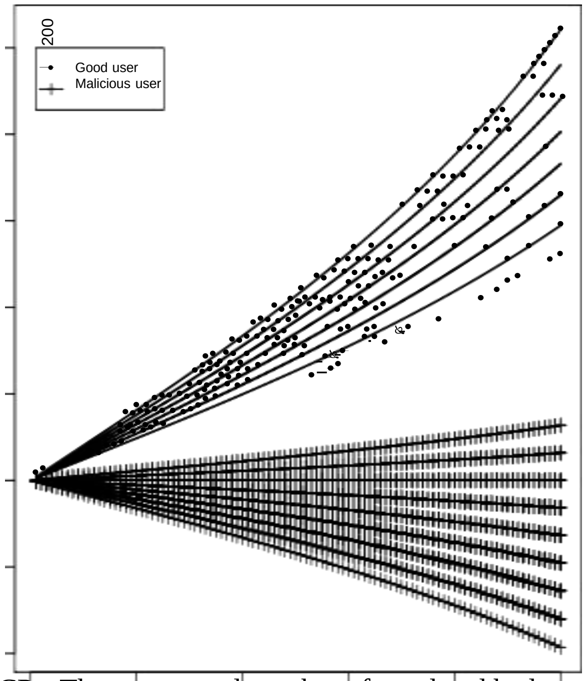
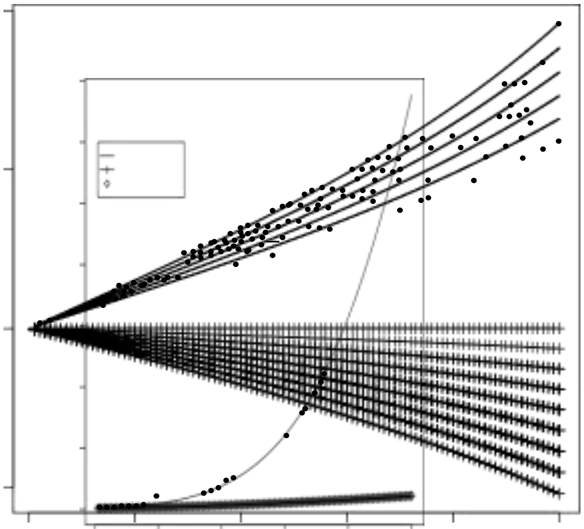


Fig. 9. There are 12 rational, 8 irrational and 10 opportunisticCPs. There are equal number of good and bad users. Plot shows the reputationoftheUsers



REFERENCES

Fig. 10. An equal number of opportunistic CPs (ten) and irrational ones (fifteen) exist. An equal amount of people utilise it for good and harm. The plot displays the CPs' reputation.

currently available RMMs for cloud computing rate CPs using the tried-and-true method of user input aggregated in a conventional manner. In this research, we provide a novel RMM that motivates CPs to distinguish between benign and malevolent users and distribute resources accordingly, preventing resource sharing.
Our results demonstrate, using both theoretical and practical analyses, the

[1] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, “On detecting co-resident cloud instances using network flow watermarking techniques,” *Int. J. Inf. Secur.*, vol. 13, no. 2, pp. 171–189, Apr. 2014.

[2] Y. Azar, S. Kamara, I. Menache, M. Raykova, and B. Shepard, “Co- location-resistant clouds,” in *Proceedings of the 6th Edition of the ACM Workshop on Cloud Computing Security*, ser. CCSW ’14. New York, NY, USA: ACM, 2014, pp. 9–20.

[3] F. Koeune and F.-X. Standaert, “Foundations of security analysis and design iii,” A. Aldini, R. Gorrieri, and F. Martinelli, Eds. Berlin, Heidelberg: Springer-Verlag, 2005, ch. A Tutorial on Physi- cal Security and Side-channel Attacks, pp. 78–108.

[7] T. Noor and Q. Sheng, “Credibility-based trust management for services in cloud environments,” in *Service-Oriented Computing*, ser. Lecture Notes inComputer Science, G. Kappel, Z. Maamar, and H.Motahari-Nezhad, Eds. Springer Berlin Heidelberg,2011, vol. 7084, pp. 328–343.

[8] M. Macas and J. Guitart, “Trust-aware operation of providers in cloud markets,” in *Distributed Applications and Interoperable Systems*, ser. Lecture Notes in Computer Science, K. Magoutis and P. Pietzuch, Eds. Springer Berlin Heidelberg, 2014, vol. 8460, pp. 31–37.

[9] T. Å gotnes, W. van der Hoek, and M. Wooldridge, “Robust normative systems,” in *Normative Multi-Agent Systems,15.03. - 20.03.2009*, 2009.

[10] A. Whitby, A. Jsang, and J. Indulska, “Filtering out unfair ratings in bayesian reputation systems,” in *AAMAS04*, 2004.

C. Dellarocas, “Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior,” in *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, ser. EC ’00. New York, NY, USA: ACM, 2000, pp. 150–157.

[11] M. Chen and J. P. Singh, “Computing and using reputations for internet ratings,” in *Proceedings of the 3rd ACM Conference on Electronic Commerce*, ser. EC ’01. New York, NY, USA: ACM, 2001, pp. 154–162.

[12] A. Das and M. Islam, “Securedtrust: A dynamictrust computa- tion model for secured communication in multiagent systems,” *Dependableand Secure Computing, IEEE Transactions on*, vol. 9, no. 2, pp. 261–274, March 2012.

[13] A. K. Despotovic Z, “Maximum likelihoodestimation of peers? performances in p2p networks,”in *Proceedings of the 2nd workshop on the economics of peer-to-peer systems*, 2004.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia- Molina, “The eigen- trust algorithm for reputation management in p2p

https://doi.org/10.5281/zenodo.12706987

networks," in *Proceedings of the 12th International Conference on World Wide Web*, ser. WWW '03. New York, NY, USA: ACM, 2003, pp. 640–651.

[15] B. Yu, M. Singh, and K. Sycara, "Developing trust

in large-scale peer-to-peer systems," in *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, Aug 2004, pp. 1–10.

[16] A. Whitby, A. Jsang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," 2004.

[17] B. E. Commerce, A. Jsang, and R. Ismail, "The beta reputation system," in *In Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

[18] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck, "Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model," in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '05. New York, NY, USA: ACM, 2005, pp. 997–1004.

[19] H. Zhao, X. Yang, and X. Li, "An incentive mechanism to reinforce truthful reports in reputationsystems," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp.951–961, May 2012.

[20] R. Jurca and B. Faltings, "An incentive compatible reputation mechanism," in *Proceedings of the Second International Joint Con- ference onAutonomous Agents and Multiagent Systems*, ser. AAMAS '03. New York, NY, USA: ACM, 2003, pp.1026–1027.

[21] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proceedings of the 5th ACM Conference on Electronic Commerce*, ser. EC '04. New York, NY, USA: ACM, 2004, pp. 102–111.

[22] T. G. Papaioannou and G. D. Stamoulis, "An incentives' mech- anism promoting truthful feedback in peer-to-peer systems," in *Proceedingsof the Fifth IEEE International Symposium on Cluster Computing and the Grid - Volume 01*, ser. CCGRID '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 275–283.

[23] J. Witkowski, "Truthful feedback for sanctioning reputation mech- anisms," *CoRR*, vol. abs/1203.3527, 2012.

[24] E. Ayday and F. Fekri, "Robust reputation management using probabilistic message passing,"in *Proceedings of the Global Commu- nications Conference, GLOBECOM 2011, 5-9 December2011, Houston, Texas, USA*, 2011, pp. 1–5.

[25] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and

A. Yumerefendi, "Embedding virtual topologies in networked clouds," in *Proceedings of the 6thInternational Conference on Future Internet Technologies*, ser. CFI '11. New York, NY, USA: ACM, 2011, pp. 26–29.

[26] Y. Zhu and M. Ammar, "Algorithms for assigning substrate net- work resources to virtual network components," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communica-tions. Proceedings*, April 2006, pp. 1–12.

[27] D. G. Andersen, "Theoretical approaches to node assignment," 2002.

[28] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism

detection," in *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems andArchitec- tures*, ser. VISA '09. New York, NY, USA:ACM, 2009, pp. 81–88.

[29] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.

R. Ricci, C. Alfeld, and J. Lepreau, "A solver for thenetwork testbed mapping problem," *SIGCOMM Comput.*