A robust reputation management mechanism in the federated cloud

Subhasis Thakur
John G. Breslin

Abstract—In the Infrastructure as a Service (IaaS) paradigm of cloud computing, computational resources are available for rent. Although it offers a cost efficient solution to virtual network requirements, low trust on the rented computational resources prevents users from using it. To reduce the cost, computational resources are shared, i.e., there exists multi-tenancy. As the communication channels and other computational resources are shared, it creates security and privacy issues. A user may not identify a trustworthy co-tenant as the users are anonymous. The user depends on the Cloud Provider (CP) to assign trustworthy co-tenants. But, it is in the CP's interest that it gets maximum utilization of its resources. Hence, it allows maximum co-tenancy irrespective of the behaviours of users. In this paper, we propose a robust reputation management mechanism that encourages the CPs in a federated cloud to differentiate between good and malicious users and assign resources in such a way that they do not share resources. We show the correctness and the efficiency of the proposed reputation management system using analytical and experimental analysis.

Index Terms—Virtual network embedding, Federated cloud, Reputation, Trust, Multi-tenancy

1 Introduction

In the IaaS paradigm of cloud computing, computational resources are shared to reduce the cost of renting them, i.e., there exists multi-tenancy. As the communication channels and other resources are shared, this creates security and privacy issues. Examples of such problems are side-channel attacks, probe attacks, etc. [1], [2], [3]. These security issues prevent some users from adopting cloud computing. To increase user's trust on Cloud Providers (CP), the reputation of the CPs can be used [4], [5] as it helps users to choose an appropriate CP. A reputation management mechanism (RMM) aims to take account of the malicious and selfish behaviours of CPs and reflect this on their reputation [6]. In this paper, we propose a robust RMM in the federated cloud with focus on multi-tenancy. In a multi-tenant cloud, a user depends on the CP for trustworthy co-tenants. In this paper we propose a novel reputation management mechanism that encourages the CPs to assign good co-tenants to a good user.

Reputation in the Federated Cloud: A federated cloud is constructed by contributions from several cloud providers and a virtual network request may be fulfilled by more than one cloud provider. In a federated cloud, a CP risks its own reputation as it shares its resources with other CPs (a virtual network request may span over the resources owned by several CPs). The problem in a virtual network may originate from the physical resources owned by other CPs. To evaluate the reputation of CPs, we can use the following form of feedback:

1) Feedback from CPs about other CPs: This form of feedback is hard to implement as CPs needs to share information about their own resources.
2) Feedback from the customers about CPs: It can be more easily obtained. But such information may be malicious and faulty. Also, as a virtual network may span over the resources owned by several CPs, it will be difficult for a customer to accurately identify the CP that is responsible for a fault.
3) Feedback from the CPs about the users: This form of feedback is easy to obtain. A CP can monitor the activities of its customers and decide on whether or not a customer has bad intentions.

In this paper we use the third kind of feedback to evaluate the reputation of the CPs. It is possible to misreport such feedback. In this paper we propose a mechanism that encourages CPs to report correct feedback about the customers.

CP's reputation and multi-tenancy: Existing RMMs for cloud computing gather feedback from users and aggregate them to obtain reputations for the CPs. Also,

1) It attempts to differentiate between fair feedback from unfair feedback provided by the users [7] about the performance of the CPs.
2) It also differentiates between faults in the physical networks and the intentional activities of CPs that lead to disruption in the physical network. Therefore, faults (which are assumed to be beyond the control of the CP) do not impact reputations of CPs [8].

In a contrast with existing RMMs, in this paper we propose a RMM with a focus on multi-tenancy. Sharing computational resources with others is the main concern of users as other co-tenants may be malicious. Note that,

- S. Thakur is with Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, Università degli Studi dell'Aquila, Via Vetoio 1, Coppito, 67100 L'Aquila, Italy, E-mail: subhasis.thakur@univaq.it
- J.G. Breslin is with National University of Ireland, Galway, Ireland, E-mail: john.breslin@nuigalway.ie

Manuscript received April 19, 2005; revised September 17, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2689020, IEEE Transactions on Cloud Computing

1) The co-tenants of a user are anonymous. Hence, a user can not choose with whom it will share computational resources.
2) The user depends on the CP to assign good co-tenants.

Thus, from a user's perspective, with the focus on co-tenancy, it will have more trust in a CP if it differentiates between good and malicious users and if it does not allow them to share resources. Thus the capability and willingness of such differentiation between good and malicious users is the main parameter that decides the reputation of a CP. If a CP does not make such differentiation, then it should receive a low reputation when compared with another CP who makes such a differentiation. In this paper, we propose a RMM that considers the CP's capability and willingness to make such differentiation among its users.

It is in the CP's interest that it gets maximum utilization of its resources. Hence, it allows maximum co-tenancy irrespective of the behaviours of the users. In this paper, we work on the federated cloud, where the physical network is contributed by multiple stakeholders and it is a connected graph. In the federated cloud, virtual network requests are mapped to the parts of the physical network owned by multiple CPs. Thus the CPs may collaborate to satisfy a virtual network requirement. Note that,

1) as the CPs collaborate to satisfy virtual network requests, it may happen that a CP, say CP_1, does not differentiate between good and malicious users but another CP, say CP_2, does the opposite. If CP_1 and CP_2 collaborate then, although CP_2 does not intend, it may have to allow a good user to become a co-tenant with a malicious user as part of this collaboration with CP_1.
2) Thus, the behaviour of a CP affects its collaborators.

Hence, we make the following assumptions:

1) CPs share the information about multi-tenancy. Also this information can not be manipulated.
2) However, they may misreport the actual behaviour of users.

Briefly, our RMM works as follows:

1) First, each CP distinguishes malicious users from good users and it should assign resources to them such that the following holds:
    a) It must not allow any malicious user to become a co-tenant of a good user.
    b) It may allow malicious users to share resources among themselves.
2) Next, the CPs share information about multi-tenancies.
3) Each CP reports the behaviour of users to the RMM.
4) A CP's reputation is increased if the reputations of the users in each group of multi-tenant users are consistent, i.e., either their reputations increase or decrease. This means that if changes in the reputation of the users are similar, then the CPs must have correctly partitioned the good users from the malicious users and did not allow them to share resources.

In the above model of our RMM, the motivation for misreporting the behaviour of users is as follows:

- A CP gets better reputation if changes in the reputations of the users in each group of multi-tenant users are consistent.
- Hence, it is in its interest to misreport the reputations of users in such a way that the changes in the reputations of the users in each group of multi-tenant users become consistent.

We use the following behavioural models of the CPs:

1) *Rational CP:* A rational CP always reports the true behaviour of the users.
2) *Irrational CP:* An irrational CP reports that a group of multi-tenant users are all good users or all malicious users irrespective of the actual behaviour of its users.
3) *Opportunistic CP:* An Opportunistic CP reports that a group of multi-tenant users are good users if the majority of them are actually good, otherwise it reports the opposite.

In the presence of these three types of CPs, we show that,

1) *Robustness:* We analyse the robustness of the RMM. We use the notion of robustness in a normative system, as developed in [9]. In this notion of robustness, it is assumed that a subset of agents in a normative multi-agent system always violate the norms. Given the fraction of such non-compliant agents, the multi-agent system is robust if it works properly as other agents remain compliant. In this paper, we use a similar notion of robustness. We show the demography of rational and irrational agents (with a majority of irrational agents) for which the proposed RMM remains functional.
2) *Reputation of the CPs:* We show that the reputations of the CPs who differentiate between good and malicious users, and do not allow them to share resources, increase compared with the CPs who do not make such differentiation.
3) *Reputation of the users:* We show that, a good user gets better reputation than a malicious user.

1.1 Organization

The paper is organized as follows: in section 2, we discuss the related literature. In section 3, we describe the proposed RMM. In section 4, we present an analytical analysis of the proposed RMM. In section 5, we show experimental analysis of the proposed RMM. We conclude the paper in section 6.

2 Related work

The research in RMMs in the IaaS paradigm of cloud computing is a conjunction of three research themes, (a) online reputation management, (b) reputation management in cloud computing and (c) virtual network embedding. In this section we briefly discuss the state of the art in these research themes.

2.1 Online reputation management

As mentioned in [10], there are two types of mechanisms to identify unfair feedbacks. The endogenous mechanisms [11] [12] only use the feedback to determine an unfair feedback. These mechanisms are based on statistical properties of the feedbacks. Often these mechanisms use the history of feedbacks and assume that majority of feedbacks are fair. The exogenous mechanisms incorporate external information to determine whether a feedback is fair or unfair. Examples of such information includes the credibility of the buyers. [13] uses personalized similarity measure to rate the recommendation credibility. In this mechanism, the credibility of an evaluator is determined by its peers whose have interacted with it. Similar approach to determine the credibility is used in [14]. [15] uses the service trust as the parameter to determine the feedback credibility. But this mechanism is vulnerable in the situations where the service provider faces competition and may send unfair feedbacks about its competitors. [16] proposes the weighted majority algorithm (WMA) that assigns weights in such a way that the relative weight assigned to the successful advisors is increased and the relative weight assigned to the unsuccessful advisors is decreased. [11] identifies the nearest neighbors of a buyer agent based on their preference similarity. Preference similarity is calculated using the number of their similar ratings for commonly rated sellers. After identifying the nearest neighbours of the buyer agent, cluster filtering is used to identify unfair rating. [17] extends the reputation management systems developed in [18] to filter out unfair ratings using the iterated filtering. [12] has used the buyers reputations in the calculation of the sellers reputation. [19] propose the TRAVOS model, which is a trust and reputation model for agent-based virtual organizations. This mechanism first estimates the accuracy of the current reputation advice based on the amount of accurate and inaccurate previous advices which are similar to the current reputation advice. Next, it adjusts reputation advice according to its accuracy. The aim of this task is to reduce the effect of inaccurate advice. However, this model assumes that seller agents act consistently, which might not be true in many cases.

There are several algorithms for designing incentives for reputation management system. [20] has modelled the incentive system using a payment game in such a way that the agents who provide truthful feedbacks get more utility. [21] proposed a payment scheme for feedback submission that encourages truthful feedback. In this mechanism, an agent gets paid if its feedback about a target agent matches the next feedback about the same target agent. The incentive model proposed in [22] is based on prisoners dilemma. In this model agents with truthful feedbacks get better utility. [23] proposed a sanctioning mechanism to obtain truthful feedback. In this model, in every transaction both parties submit a report about each other. If the reports in each transactions are not consistent then both parties are punished. [24] studied the feasibility of payment system for eliciting truthful feedbacks for online auction systems. [25] introduces an iterative probabilistic method for reputation management.

2.2 Virtual network embedding

In virtual network embedding problem [26], a customer requests a network $G^V = (N^V, E^V, W^V)$ (N^V is the set of nodes, E^V is the set of edges, and W^V denotes the weights of the vertices and the edges. The vertex weight denotes the share of CPU or computation usage and weights on the edges denote the bandwidth for the communication channels) to be embedded into a physical network $G = (N, E, W)$ (N, E, C are nodes, edges and weights of the vertices and the edges) such that certain conditions are satisfied (a) weights of the vertices and the edges mapped into the physical network must be at least the requested weights of the vertices and the edges and (b) for each requested vertex there must be one mapped vertex (or a set of vertices) and for each requested edge there must be one mapped edge (or a mapped path). Note that a vertex (and an edge) can be shared by multiple VNEs. The VNE problem is known to be NP-complete [27], [28], [29] and heristics are developed in [30], [31], and approximation algorithms are developed in [32].

VNE is a cost efficient solution as it allows to rent a computational infrastructure. Yet, the unavailability of the physical network due to faults or maintenance disrupts the services based on it. Survivable virtual network embedding (SVNE) [33] is an extension of the VNE problem where we also need to allocate on the set of supplementary resources in the physical network to the VNE solution. These supplementary resources insure continuous availability of the physical resources. The physical network failure can be single (or multiple) vertex and edge failure or both. [34] showed that 20% of all failure is due to maintenance, 53% due to router related and 70% is due to single link failure. [35] showed that link failure is 10 times more than node failure. There are two approaches to the SVNE problems [36] (a) proactive approaches: in this approach we need to maintain backups or alternative resources on standby in case parts of the physical network fails and (b) reactive approaches: in this approach we need to find the VNEs with the available resources once parts of the physical network collapses. While both measures do not guarantee data loss, the first method is also concerned with the optimization problem of finding the most cost efficient backups. [37] presented reactive mechanism for solving SVNE problems for single link failure with shared backups. [38] studied the same problem but it proposed to keep paths as backups instead of backing up each primary link. [39] studied reactive solutions for SVNEs with certain quality control measures. [40] studied single node failure SVNE as it enhance the request with additional redundant nodes. Similar single node failure SVNEs were analysed in [41] using graph decomposition. [42] studied single regional failure (connected subgraph of the physical network fails) while [43] studied the similar problem with location constraints (conditions on the backup or replacement). [44] studied the SNVE problem for both node and edge failures. Also it considers minimizing the backup. [45] proposed a distributed solution for the SVNE problem using multi agent systems. [46] proposed an online algorithm for SVNE problem with competitive ratio $(n-1)$ where n is the number of the vertices in the physical network. [47] considered embedding problem

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2689020, IEEE Transactions on Cloud Computing

where the physical network is such that it is not possible to distinguish between the actual mapping and the backups. Hence it includes a set of additional paths such that if an edge fails then there will be at least one backup. It proves that this problem is NP-complete and provides heuristics.

2.3 Reputation management in cloud computing

[48] proposed a multi-faceted trust management model with the intention to distinguish between fair and unfair feedbacks about the cloud providers. [49] also proposed a multi-faceted reputation management model that allows the users to evaluate the cloud providers using various features. [50] proposed a trust evaluation of the cloud providers based on the violation of contracts described in the service level agreement. [51] [7] proposed a mechanism to isolate unfair and malicious trust feedback in cloud computing. [8] proposed a policy on reputation management that minimizes the impact of system failure on the reputation of the cloud providers. [1], [2], [3] studied the security and privacy issues due to multi-tenancy. We refer to [52], [4], [5] for more detailed survey on trust mechanisms for cloud computing.

3 Reputation management mechanism

3.1 Informal model

Informally the RMM is as follows:

1) There is a finite number of CPs and a finite number of users. It is assumed that each CP hosts virtual network request from all users. There are three types of CPs, (a) rational CP, (b) irrational CP and (c) opportunistic CP. There are two types of users, (a) good user: one who does not cause any security or privacy issues and (b) malicious user: one who causes security and privacy issues. A malicous cotenant may create various security problems such as side channel attack [53] (attack based on the physical implementation of the network), DOS attack [54], Network probe attack [55] (attack to find the topology of the network). We assume that if a CP hosts a user then it can monitor the user's activities and recognize whether it is malicious or not.
2) First,
    a) each CP labels each user as either a good user or a malicious user.
    b) It assigns virtual resources to the users.
    c) The users are partitioned in groups such that in each group all users share resources with each other, i.e., they are multi-tenant.
    d) Each CP announces partitions over the users, i.e., they announce the multi-tenancy information to the RMM.
3) Next,CPs monitor activities of the users and report it to the RMM. A CP can either provide a positive or a negative vote for a user. It will be assumed that the federated cloud infrastructure will provide the RMM with the means of communication with the individual CPs and using such communication channels CPs regularly provide feedback (i.e., positive or negative vote about the users) to the RMM. A negative vote indicates that the user is malicious according to the CP (who has provided a negative vote for it) otherwise it is a good user. We use the following interaction model between the CP and the users:
    a) At each step, the users generate certain events which are interpreted as indications of their good or malicious behaviours.
    b) At each step, after observing the events generated by the users, each CP reports the behaviour of a user as follows:
        - Positive vote: It indicates that the CP perceives the user as a good user.
        - Negative vote: It indicates that the CP perceives the user as a malicious user.
    c) After the RMM received the votes for a user, it calculates the user's reputation as follows:
        i) If a user receives more positive votes than the negative votes then its reputation is increased.
        ii) If a user receives less positive votes than the negative votes then its reputation is decreased.
        iii) If a user receives equal number of positive votes and the negative votes then its reputation remains the same.
4) In each step, after updating the reputation of the users, the RMM updates the reputation of the CPs as follows:
    a) for each group of multi-tenant users, if reputation of all users are increased or reputation of all users are decreased then, the CP's reputation is increased.
    b) For each group of multi-tenant users, if reputations of some users are increased (decreased) and the same for the rest of the users are decreased (increased) then, the CP's reputation is decreased.

Note that, a CP's reputation depends on its correct segmentation of the users, i.e., partitioning the users into groups (each group is a set of multi-tenants, i.e., share resources among themselves) where a good user should be in a group with only other good users and a malicious user should be in a group with other bad users. The correctness of a CP's segmentation of the users is determined by the change of reputation of the users. If it has placed only good users in a group then the reputation of the users in that group will increase (as other CPs vote for the bad and good users) and if it has placed only bad users in a group then the reputation of the users in this group will decrease. Thus a CP's reputation increases when the reputations of all users in a group either increase or decrease. But if the CP has placed both the good and the bad users in the same group then the reputation of some users will increase and the reputation of other users will decrease. Hence, a CP's reputation decreases when the reputations of some users in a group increase and reputations of other users in the same group decrease.

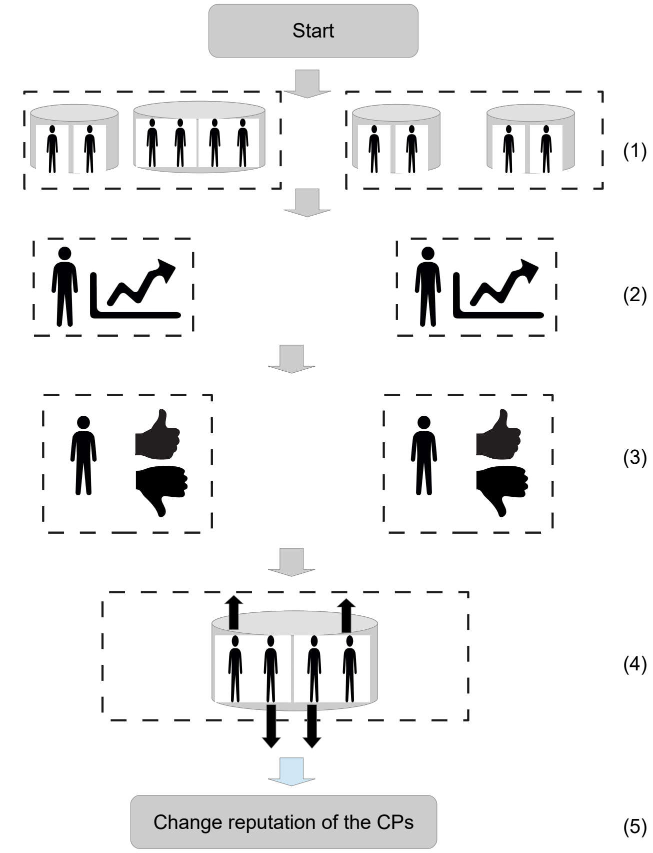


Fig. 1. The RMM is as follows:(1) each CP groups the users into sets of multi-tenants, (2) performance and activities of all users are monitored, (3) based on these observations the CPs vote (positive or negative) on each user and (4) finally, based on the change in the reputation of each user in a group the reputation of the CP's are changed.

3.2 Formal model

We denote a set of n cloud providers as $C = \{c_1, \ldots, c_n\}$ and a set of m users as $U = \{u_1, \ldots, u_m\}$. We assume the following:

- Any misbehaviour of a user is detected by its host, i.e., the CP that hosts its virtual network.
- There is an initial reputation of the users, denoted as $R(u_i) \in [0, 1]$. Reputation 0 indicates that the reputation of the user is the minimum and reputation 1 indicates that the reputation of the user is the maximum. All CPs know this initial reputation values of the users.
- There is an uniform initial reputation for all CPs. Reputation of the CPs is denoted as $R(c_i) \in \mathbb{R}_{>0}$ (positive real number).
- If a user misbehaves then, its host should report such misbehaviour to the RMM.
- Each CP generates a $k \geq 1$ partition over the users, denoted as $\pi = \{\pi_1, \ldots, \pi_k\}$. The set of users in each π_i shares at least an edge or a vertex in the physical network.
- Each CP informs the partition π to the RMM and we assume that it can not be manipulated, i.e., the report about the partition is always true.
- A CP can misreport the behaviour of the users.

First we define the virtual network embedding which is a function that associates physical resources with the virtual network request.

Definition 1. *(Virtual network embedding) Given a virtual network request $G^R = (V^R, E^R)$, vertex weight function $W_1^R : V^R \mapsto \mathbb{R}_{>0}$ and edge weight function $W_2^R : E^R \mapsto \mathbb{R}_{>0}$, a virtual network embedding f, maps G^R to a physical network $G = (V, E)$ with vertex weight function $W_1 : V \mapsto \mathbb{R}_{>0}$ and edge weight function $W_2 : E \mapsto \mathbb{R}_{>0}$ such that the following holds:*

1) *For each vertex $v_1 \in V^R$ there exists a subset $S \subset V$ such that $f(v_1) = S$.*
2) *For each vertex $v_1 \in V^R$, $W_1^R(v_1) \leq \sum_{v_2 \in f(v_1)} W_1(v_2)$.*
3) *For each edge $e_1 \in E^R$ there exists a subset $S \subset E$ such that $f(v_1) = S$ and if $|S| > 1$ then S is a connected path.*
4) *For each edge $e_1 \in E^R$, $W_2^R(e_1) \leq \sum_{e_2 \in f(e_1)} W_1(e_2)$.*

Note that, the above virtual network embedding process does not offer computational resources exclusive to the users, rather they share computational resources. We partition the users based on sharing of computational resources. According to this partition, if two users share a resource then they reside in the same group.

Definition 2. *(Partition over the users) The set of users hosted by a CP c_i will be denoted as $U(c_i) \subset U$. A partition over $U(c_i)$ into $k \geq 1$ sets will be denoted as $\pi = \{\pi_1, \ldots, \pi_k\}$ such that the following holds:*

- *for each pair of users u_1 and u_2 in any set π_i with virtual network requests $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$, either $f(V^1) \cap f(V^2) \neq \emptyset$ or $f(E^1) \cap f(E^2) \neq \emptyset$.*
- *For any $\pi_i \cap \pi_j = \emptyset$ for any $i \neq j$.*

Now, we define the user's reputation as follows:

Definition 3. *(User's reputation:) (Refer to Figure 2) Let $\mathbb{C}$ be a circle with center c_u and radius r. $\mathbb{C}$ will be called the users circle. The lines L_h and L_v are the x and the y axis respectively. Each user is assigned two points on the circumference of the user's circle, for example in Figure 2 two points are a and b. Reputation of the users is given by the angle $\angle(a, c_u, b)$ such that the following holds:*

- *Initially, for all users, $\angle(a, c_u, o) = \angle(b, c_u, o) > 0$ where o is the point of intersection between the circumference of $\mathbb{C}$ and L_v.*
- *The reputation of each user is in the range $\tan(\angle(a, c_u, o)/2) - \tan(\angle(b, c_u, o)//2)$.*
- *We restrict $\angle(a, c_u, o)$ or $\angle(b, c_u, o)$ in the range $[0°, 90°]$. Hence, the reputation is of each user is in the range $[-1, 1]$.*
- *The points a and b are user's point.*

Next we define the process to change a user's reputation. We change a user's reputation using CP's report about its behaviour.

Definition 4. *(Change in user's reputation:) (Refer to Figure 2) A user's reputation is changed as its host CP reports positive*

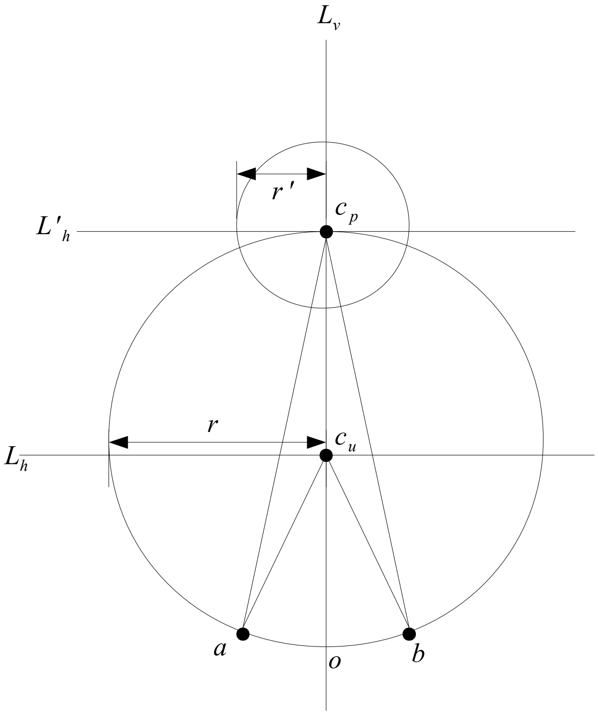


Fig. 2. Reputation of CPs and users.

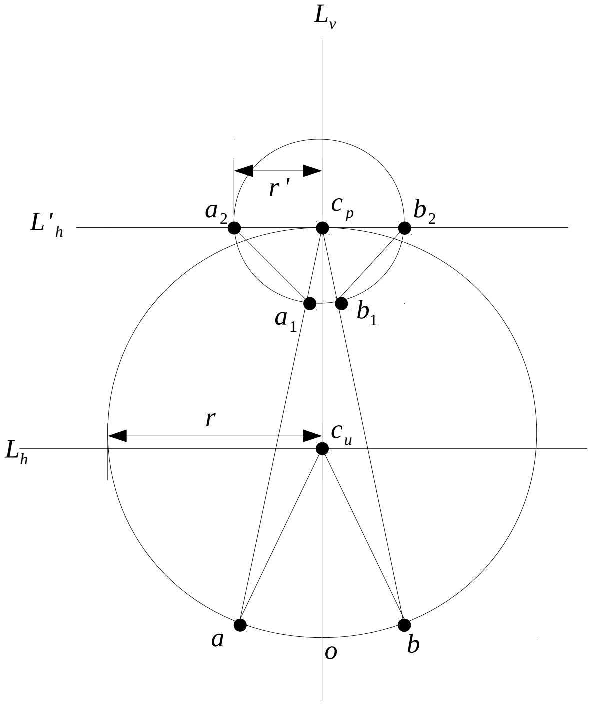


Fig. 3. Association between the user's reputation and the CP's reputation.

or negative feedback about it. A CP can either vote positive or vote negative. Let $\epsilon > 0$ be a small positive rational number. If the CP has provided a positive vote then the angle $\angle(a, c_p, o)$ is increased by ϵ and if it has provided a negative vote then the angle $\angle(b, c_p, o)$ is increased by ϵ. Hence if a user received positive vote then its reputation is increased and if it receives negative votes then its reputation is decreased.

Now we define the CP's reputation using figure 2.

Definition 5. *(CP's reputation:) (Refer to Figure 2) Given the user's circle $\mathbb{C}$, each CP c_i is assigned a circle $\mathbb{C}_i$ such that its center is c_p (point of intersection between the line L_v and L'_h) and its radius is r'. The reputation of the CP c_i is the radius of the circle $\mathbb{C}_i$. This circle is called CP circle.*

Now we define the dependencies between a CP's reputation and the user's reputation.

Definition 6. *(Association between CP's and user's reputation) (Refer to Figure 3) If a CP c_i has hosted the user u_j with user's points a and b then we assign two lines to the CP c_i as follows:*

- *Lines are (a_1, a_2) and (b_1, b_2).*
- *Find the point a_1 (b_1) on the circumference of the CP circle $\mathbb{C}_i$ as the point of intersection between the line (a, c_p) $((b, c_p))$ and the circumference of the CP circle $\mathbb{C}_i$.*
- *These lines (a_1, a_2) and (b_1, b_2) will act as the association 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 a CP's reputation and the user's reputation, next we define the change in a CP's reputation due to changes of the user's reputations.

Definition 7. *(Change in CP's reputation for one user) (Refer to Figure 4) Let $\delta(u_j, c_i)$ denotes the change in reputation of the CP c_i as the reputation of the user u_j changes. We calculate $\delta(u_j, c_i)$ as follows:*

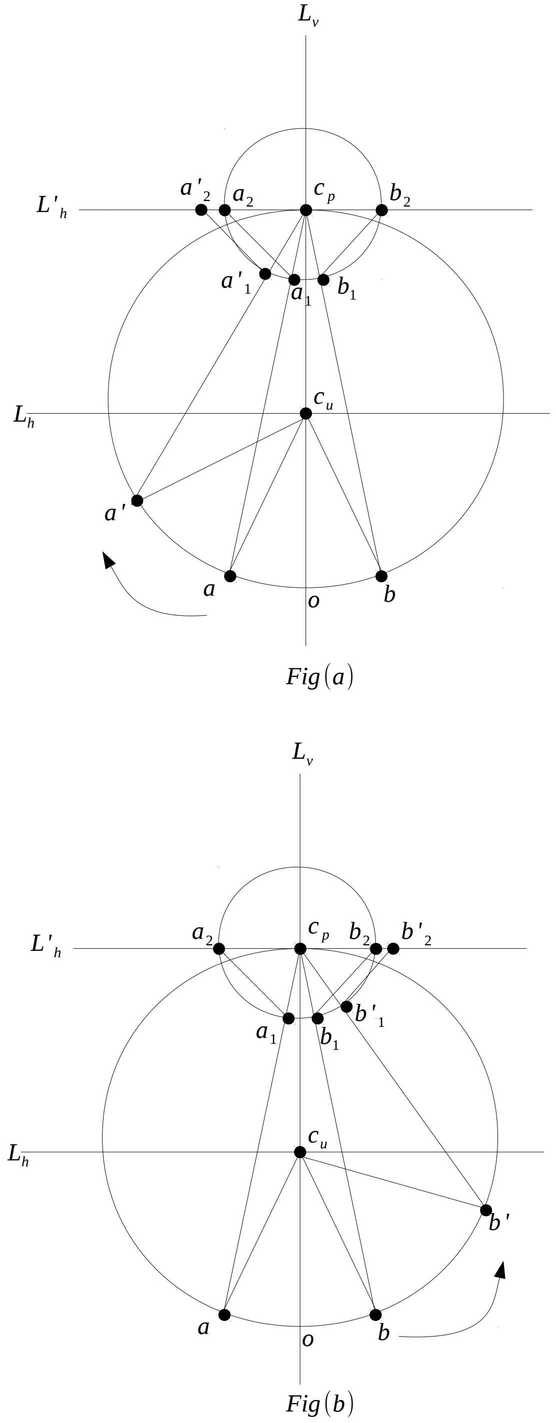
- *Let u_j has received positive votes and the angle $\angle(a, c_u, o)$ is increased to angle $\angle(a', c_u, o)$.*
- *Let a'_1 be the point of intersection between the line (a', c_p) and circumference of the CP circle of c_i.*
- *We draw a parallel line from a'_1 w.r.t the line (a_1, a_2) and let this line intersects the line L'_h at a'_2.*
- *$\delta(u_j, c_i)$ is the length of the line segment (a'_2, a_2).*

Similarly, $\delta(u_j, c_i)$ is calculated if u_j has received negative votes.

We illustrate the change in c_i's reputation as the result of changes in a user's reputation in Figure 4. Figure 4 (a) shows the change as the user receives positive votes and Figure 4 (b) shows the change as the user receives negative votes. Note that, as the user receives more positive (or negative) votes, the reputation of the CP is increased more.

Definition 8. *(Change in CP's reputation for partition over the users) (Refer to Figure 4) Let the CP c_i partitions the users into the sets as follows $\pi = (\pi_1, \ldots, \pi_k)$. For each set π_i, we calculate the change in the CP's reputation as follows:*

- *Let $S \subset \pi_i$ are the users who have received positive votes and $T \subset \pi_i$ is the set of users who have received negative votes ($S \cup T = \pi_i$).*
- *For each $u_j \in \pi_i$, we calculate $\delta(u_j, c_i)$.*



$Fig(a)$

$Fig(b)$

Fig. 4. Change in CP's reputation if it hosts one user.

- *The change in c_i's reputation due to change in reputation of the users in π_i, denoted as $\Delta(\pi_i, u_i)$, is $|\sum_{u_j \in S} \delta(u_j, c_i) - \sum_{u_j \in T} \delta(u_j, c_i)|$.*

Finally, the total change in CP c_i's reputation is $\sum_{\pi_i \in \pi} \Delta(\pi_i, c_i)$.

Following the above procedure of changing the CP's reputation from its partition over the users, we observe the following:

Lemma 1. *Let there are two partitions $\pi = (\pi_1, \ldots, \pi_k)$ and $\pi' = (\pi'_1, \ldots, \pi'_k)$ such that the following holds:*

1) *In each set ($\pi_i \in \pi$), most of the users who have received positive votes (or most of the users who have received negative votes).*
2) *In each set ($\pi'_i \in \pi'$), the number of users who have received positive votes is almost equal to the number of users who have received negative votes).*

The increment in the CP's reputation from partition π is more than the same for the partition π'.

Now we define the types of CPs. A CP may be a rational CP or an irrational CP or an opportunistic CP.

Definition 9. *(Adversary) We define three types of behaviours of the CPs as follows:*

- *Rational CP: These CPs correctly separate the good users from the bad users. Also, they report the true behaviour of the users, i.e., if a user misbehaves then its provides a negative vote for it otherwise it provides a positive vote.*
- *Irrational CP: These CPs incorrectly separate the good users from the bad users. Also, they misreport the true behaviour of the users, i.e., either they report positive vote for all users or negative vote for all users in a partition.*
- *Opportunistic CP: These CPs also, incorrectly separate the good users from the bad users. They misreport the true behaviour of the users as follows: let the opportunistic CP partitions the users into k sets as $\pi = \{\pi_1, \ldots, \pi_k\}$. For any set π_i*
    1) *If the majority of users in π_i behave well, then it reports positive vote for all users in π_i.*
    2) *If the majority of users in π_i misbehave, then it reports negative vote for all users in π_i.*

4 Analysis

In Lemma 1 we estimate the change in a CP's reputation due to the change of a user's reputation.

Lemma 1. *If the CP c_i hosts the user u_j with user points A and B and if u_j receives positive votes then, the following holds*

$$\delta(u_j, c_i) = r * \cos\theta_2 * (1 - \sin\theta_1)\cos\theta_1 - r + r * \sin(\theta_2) \tag{1}$$

where r is the initial reputation of c_i, $2\theta_1$ and $2\theta_2$ are the positive vote angle of u_j before and after it receives the positive votes.

Proof. The scenario is illustrated in figure 5. The share angle changes from $4\theta_1$ to $4\theta_2$ as the share points are changed from (A, B) to $(A1, B1)$. Note that the angles $\angle A1, c_p, o$ and $\angle A, c_p, o$ are θ_2 and θ_1 respectively. The share lines are 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 segment $zz1$ as follows: Note that, $\angle c_p, z_1, a_1 = \angle c_p, z, a = \theta$. In the

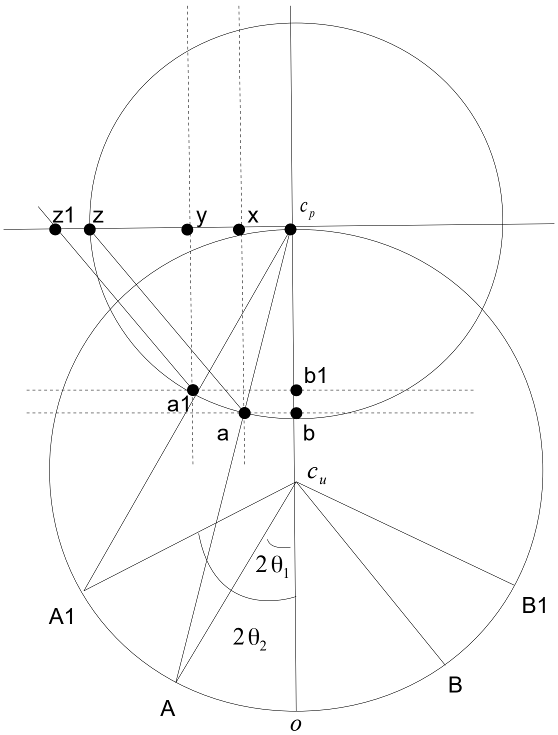


Fig. 5. Proof of Lemma 1

triangles Δabc_p and Δac_pz,

$$\frac{ab}{ac_p} = \sin\theta_1 \tag{2}$$

$$xc_p = ab = r * \sin\theta_1 \tag{3}$$

$$xz = r - r * \sin\theta_1 \tag{4}$$

Also,

$$\frac{c_pb}{ac_p} = \cos\theta_1 \tag{5}$$

$$xa = c_pb = r\cos\theta_1 \tag{6}$$

In the triangle, $\Delta(xaz)$

$$\tan\theta = \frac{xa}{xz} = \frac{r\cos\theta_1}{r - r * \sin\theta_1} = \frac{\cos\theta_1}{1-\sin\theta_1} \tag{7}$$

In the triangle, $\Delta(a1, b1, c_p)$

$$\cos\theta_2 = \frac{ya_1}{r}, ya_1 = r * \cos\theta_2 \tag{8}$$

$$\tan\theta = \frac{r * \cos\theta_2}{yz_1} = \frac{\cos\theta_1}{1-\sin\theta_1} \tag{9}$$

$$yz_1 = \frac{r * \cos\theta_2 * (1-\sin\theta_1)}{\cos\theta_1} \tag{10}$$

In triangle $\Delta(a_1, y, c_p)$,

$$\tan(90-\theta_2) = \frac{r * \cos\theta_2}{yc_p} \tag{11}$$

$$yc_p = \frac{r * \cos\theta_2}{\tan(90-\theta_2)}$$

Hence,

$$zy = r - \frac{r * \cos\theta_2}{\tan(90-\theta_2)}. \tag{12}$$

$$zz_1 = yz_1 - yz \tag{13}$$

$$zz_1 = \frac{r * \cos\theta_2 * (1-\sin\theta_1)}{\cos\theta_1} - r + \frac{r * \cos\theta_2}{\tan(90-\theta_2)}$$

$$zz_1 = \frac{r * \cos\theta_2 * (1-\sin\theta_1)}{\cos\theta_1} - r + \frac{r * \cos\theta_2 * \sin(\theta_2)}{\cos(\theta_2)}$$

$$zz_1 = \frac{r * \cos\theta_2 * (1-\sin\theta_1)}{\cos\theta_1} - r + r * \sin(\theta_2) \tag{14}$$

□

Using Lemma 1, in Theorem 1 we prove that rational CPs get better reputation than irrational and opportunistic CPs.

Theorem 1. *Let,*

- *there are n CPs $c_1, \ldots, c_n$ and m users $u_1, \ldots, u_m$.*
- *Among the CPs, there are K_1 rational CPs, K_2 irrational CPs and K_3 majority irrational CPs.*
- *There are l_1 good users and l_2 malicious users.*
- *There are k buckets. The maximum capacity is k_{max}. Each bucket represents a set of users who are co-tenants. Capacity indicates the maximum number of users that can share the resources. It is assumed that, $k * k_{max} \geq m$.*

If,

$$K_1 > \frac{m-2}{2}K_2 - K_3$$

then, the rational CPs will get higher reputations than the irrational or opportunistic CPs.

Proof. The CPs are using the following configuration:

- (Irrational CPs:) There are z^1 buckets. Each bucket has k_{max} users and $z^1 * k_{max} = m$.
- (Rational CPs:) There are $z^2 = z^{12} + z^{22}$ buckets. It is using z^{12} buckets to accommodate good CPs and z^{22} buckets to hold malicious CPs. $z^{12} * z_{max} = l_1$ and $z^{22} * z_{max} = l_2$.
- (Majority irrational CPs:) There are z^3 buckets. Each bucket has k_{max} users and $z^1 * k_{max} = m$.

At any step, for any good user the following holds:

- All rational CPs provides a positive vote. Thus it gets K_1 positive votes.
- The expected number of irrational CPs that provide negative and positive feedbacks are $K_2/2$ and $K_2/2$.
- In its bucket chosen by a majority irrational CP, the expected number of good CPs is $\frac{l_1 * z_{max}}{m}$ and the expected number of malicious users is $\frac{l_2 * z_{max}}{m}$. It gets a positive vote if $\frac{l_1 * z_{max}}{m} > \frac{l_2 * z_{max}}{m}$ otherwise it gets a negative vote.

At any step, for any malicious user the following holds:

- All rational CPs provides negative votes. Thus it gets K_1 negative votes.
- The expected number of irrational CPs that provide negative and positive feedbacks are $K_2/2$ and $K_2/2$.
- In its bucket chosen by a majority irrational CP, the expected number of good CPs is $\frac{l_1 * z_{max}}{m}$ and the expected number of malicious users is $\frac{l_2 * z_{max}}{m}$. It gets a positive vote if $\frac{l_1 * z_{max}}{m} > \frac{l_2 * z_{max}}{m}$ otherwise it gets a negative vote.

Assume that, $l_1 > l_2$.

For rational CP, the expected positive and negative votes in a bucket consisting good users are as follows:

$$\begin{aligned} p - votes =& k_{max}[\overbrace{K_1}^{\text{Rational CPs}} + \overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}} + \overbrace{K_3}^{\text{Opportunistic CPs}}] \\ n - votes =& k_{max}[\overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}}] \end{aligned} \tag{15}$$

For rational CP, the expected positive and negative votes in a bucket consisting malicious users are as follows:

$$\begin{aligned} p - votes =& k_{max}[\overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}}] \\ n - votes =& k_{max}[\overbrace{K_1}^{\text{Rational CPs}} + \overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}} + \overbrace{K_3}^{\text{Opportunistic CPs}}] \end{aligned} \tag{16}$$

Thus the change in the CP's reputation from the buckets with good users is:

$$P - Change = k_{max}[K_1 + K_3] \tag{17}$$

Thus the change in the CP's reputation from the buckets with malicious users is:

$$n - Change = k_{max}[K_1 + K_3] \tag{18}$$

Hence its reputation in one step becomes:

$$z^2 * (k_{max}[K_1 + K_3]) \tag{19}$$

$$*[\frac{r * \cos\theta_2 * (1 - \sin\theta_1)}{\cos\theta_1} - r + r * \sin(\theta_2)]. \tag{20}$$

In any bucket of the irrational CP, the expected size of good users is $\frac{k_{max} l_1}{m}$ and the expected size of malicious users is $\frac{k_{max} l_2}{m}$. For an irrational CP, the expected positive and negative votes in any bucket for the good users is

$$\begin{aligned} p - votes =& \frac{k_{max} l_1}{m}[\overbrace{K_1}^{\text{Rational CPs}} + \overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}} + \overbrace{K_3}^{\text{Opportunistic CPs}}] \\ n - votes =& \frac{k_{max} l_1}{m}[\overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}}] \end{aligned} \tag{21}$$

The same for the malicious users is:

$$\begin{aligned} P - votes =& \frac{k_{max} l_2}{m}[\overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}}] \\ n - votes =& \frac{k_{max} l_2}{m}[\overbrace{K_1}^{\text{Rational CPs}} + \overbrace{\frac{K_2}{2}}^{\text{Irrational CPs}} + \overbrace{K_3}^{\text{Opportunistic CPs}}] \end{aligned} \tag{22}$$

Thus for any bucket, the irrational CP gets the following number of positive votes:

$$\begin{aligned} p - votes &= \frac{k_{max} l_2}{m}[\frac{K_2}{2}] + \frac{k_{max} l_1}{m}[K_1 + \frac{K_2}{2} + K_3] \\ n - votes &= \frac{k_{max} l_1}{m}[\frac{K_2}{2}] + \frac{k_{max} l_2}{m}[K_1 + \frac{K_2}{2} + K_3] \end{aligned} \tag{23}$$

Hence its reputation becomes:

$$\begin{aligned} z^1 * [\frac{k_{max}(l_1 - l_2)}{m}[\frac{K_2}{2}] + \frac{k_{max}(l_1 - l_2)}{m}[K_1 + \frac{K_2}{2} + K_3]] \\ *[\frac{r * \cos\theta_2 * (1 - \sin\theta_1)}{\cos\theta_1} - r + r * \sin(\theta_2)]. \end{aligned} \tag{24}$$

Equation 24 also holds for the opportunistic CPs. Note that,

$$\begin{aligned} z^1 * [\frac{k_{max}(l_1 - l_2)}{m}[\frac{K_2}{2}] + \frac{k_{max}(l_1 - l_2)}{m}[K_1 + \frac{K_2}{2} + K_3]] \\ = z^1 * \frac{k_{max}(l_1 - l_2)}{m}[[\frac{K_2}{2}] + [K_1 + \frac{K_2}{2} + K_3]] \\ = z^1 * \frac{k_{max}(l_1 - l_2)}{m}[K_1 + k_2 + K_3] \end{aligned} \tag{25}$$

We need to show,

$$\begin{aligned} z^2 * (k_{max}[K_1 + K_3]) > z^1 * \frac{k_{max}(l_1 - l_2)}{m}[K_1 + k_2 + K_3] \\ z^2[K_1 + K_3]) > z^1 * \frac{(l_1 - l_2)}{m}[K_1 + k_2 + K_3] \end{aligned} \tag{26}$$

Note that,

$$z^2 > z^1.$$

This is because, as the rational CP needs to separate the good users from the malicious users it needs at most one more bucket than the irrational or opportunistic CP. Thus, we need to satisfy the following:

$$\begin{aligned} [K_1 + K_3] &> \frac{(l_1 - l_2)}{m}[K_1 + K_2 + K_3] \\ K_1[1 - \frac{(l_1 - l_2)}{m}] &> \frac{(l_1 - l_2)}{m}[K_2 + K_3] - K_3 \\ K_1[1 - \frac{(l_1 - l_2)}{m}] &> \frac{(l_1 - l_2)}{m}K_2 - (1 - \frac{(l_1 - l_2)}{m})K_3 \\ K_1 &> \frac{m}{m - (l_1 - l_2)}\frac{(l_1 - l_2)}{m}K_2 - K_3 \\ K_1 &> \frac{l_1 - l_2}{m - (l_1 - l_2)}K_2 - K_3 \end{aligned} \tag{27}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2689020, IEEE Transactions on Cloud Computing

Assuming there is at least one malicious user,

$$K_1 > \frac{m-2}{m-(m-2)} K_2 - K_3$$
$$K_1 > \frac{m-2}{2} K_2 - K_3 \quad (28)$$

□

Finally, we prove that a good user gets better reputation than a malicious user.

Theorem 2. *A good user will get better reputation than a malicious user.*

Proof. Using equation 15, the reputation of the good user in each step is changed by:

$$[K_1 + K_3] \quad (29)$$

Using equation 30 the reputation of the malicious users is changed by:

$$-[K_1 + K_3] \quad (30)$$

Hence, reputation of good users get better than the reputation of the malicious users. □

5 Experimental results

We simulate the federated cloud as follows:

- There are 30 CPs. A CP can be (a) a rational CP, (b) an irrational CP or (c) an opportunistic CP.
- There are 200 users. A user can be either a good user or a malicious user.
- We assume that each CP hosts all users.
- Each CP partitions the users into 10 groups (each group represents a set of co-tenants). The rational CPs do not place a good user in the same group with a malicious user. But irrational and opportunistic CPs group the users randomly.

We simulate the RMM and the interaction between the CPs and the users as follows:

1) First, each CP partitions the users into groups. It is assumed that users belonging to the same group share computational resources.
2) At every step,
    a) Each CP observes the behaviours of the users.
    b) And, it reports their behaviours to the RMM as positive votes or negative votes.
    c) After receiving the votes from the CPs, the RMM calculates the reputation of the users using Definition 3.
    d) After receiving the votes from the CPs, the RMM calculates the reputation of the CPs using Definition 8 and Lemma 2.

In Figure 6 and 7 we show the outcome of the simulation with 10 rational, 10 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Figure 6 shows the average reputations of the rational, irrational and opportunistic CPs. It clearly shows that the reputation of the rational CPs are better than irrational and opportunistic CPs. 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 difference between the mean reputation of the irrational and the opportunistic CPs is very small and almost indistinguishable.

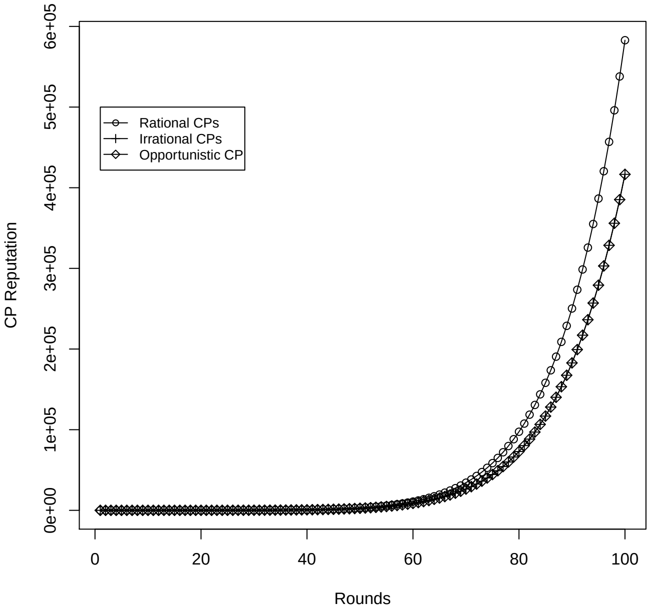


Fig. 6. There are 10 rational, 10 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputation of the CPs. Y-axis shows the reputation of the CPs.

Next, we increase the number of rational agents in next two experiments. Figures 8 and 9 shows the outcome with 12 rational, 8 irrational and 10 opportunistic CPs. There are equal number of good and bad users. It also shows that the reputation of the rational CPs is better than the same for irrational and opportunistic CPs and the reputation of good users is better than the reputation of malicious users.

We get the same results (Figures 10 and 11) with 15 rational, 5 irrational and 10 opportunistic CPs and equal number of good and bad users. But it shows that, the difference between the reputation of rational CPs and other types of CPs gets bigger as the number of rational CPs is increased.

By decreasing the number of rational agents beyond 10 we found that the difference between reputation of rational and irrational (or opportunistic) CPs is negligible. Hence in this simulation, we claim that the RMM remains functional if the number of rational CPs is at least one third of the entire population of the CPs.

6 Conclusion

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2689020, IEEE Transactions on Cloud Computing

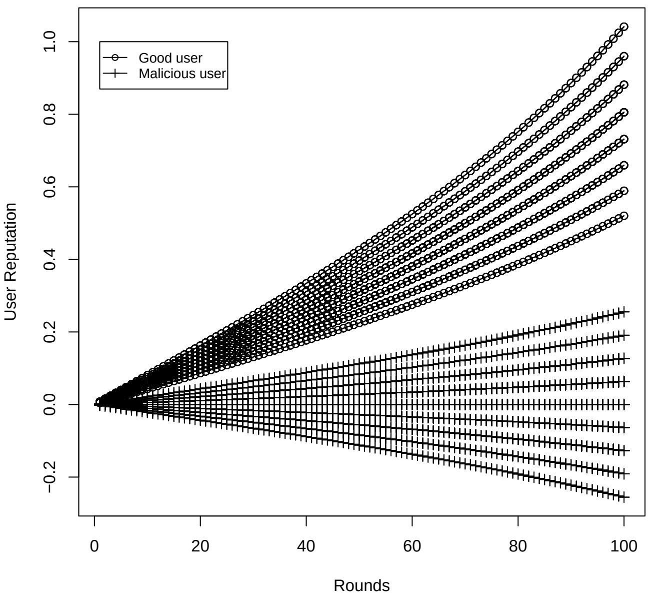


Fig. 7. There are 10 rational, 10 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputation of the Users.

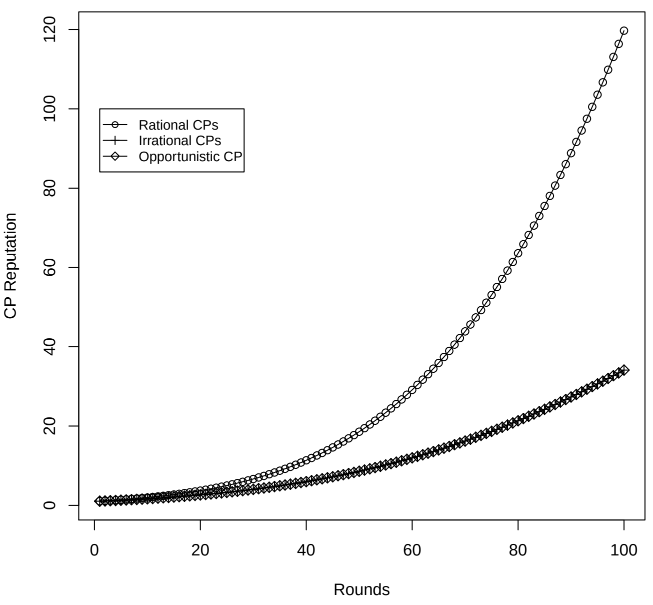


Fig. 8. There are 12 rational, 8 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputation of the CPs.

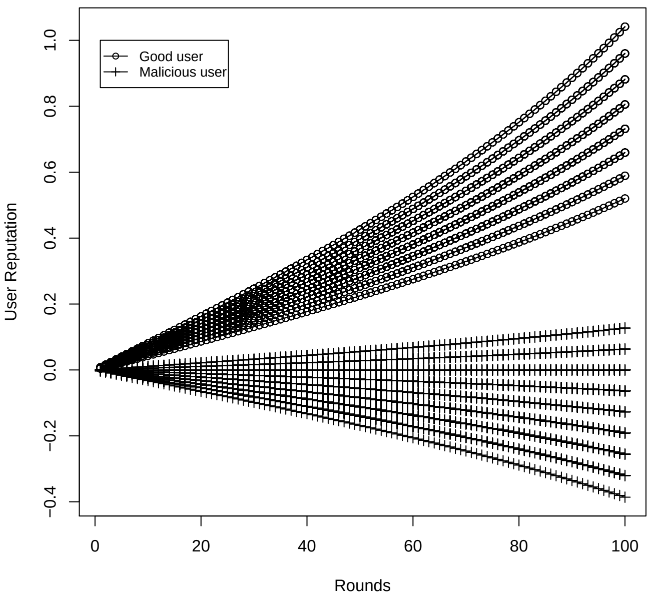


Fig. 9. There are 12 rational, 8 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputation of the Users.

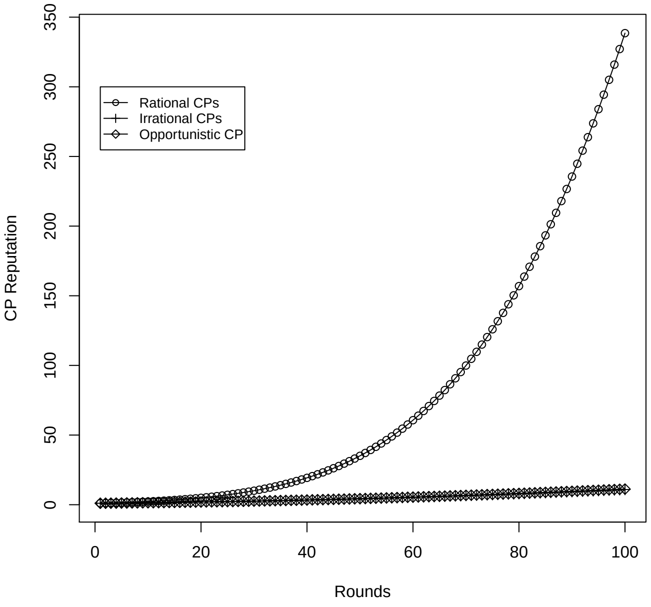


Fig. 10. There are 15 rational, 5 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputation of the CPs.

pends on the CP for allocation of safe co-tenants. Our objective 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 existing RMMs for cloud computing use traditional aggregation of feedback from users to rate the CPs. In this paper we have developed a unique RMM that encourages CPs to differentiate between good and malicious users and assign resources in such a way that they do not share resources. Using analytical and experimental evaluations we show the

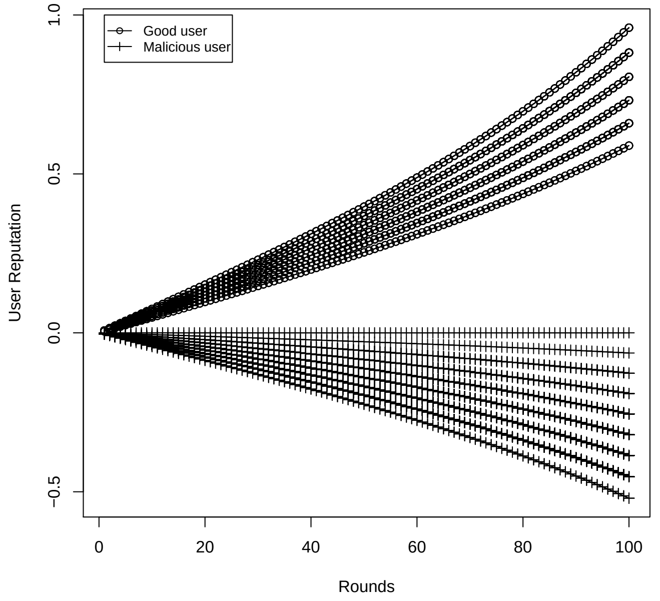


Fig. 11. There are 15 rational, 5 irrational and 10 opportunistic CPs. There are equal number of good and bad users. Plot shows the reputation of the Users.

correctness of the proposed RMM.

References

[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 Physical Security and Side-channel Attacks, pp. 78–108.

[4] S. Habib, S. Hauke, S. Ries, and M. Mhlhuser, "Trust as a facilitator in cloud computing: a survey," *Journal of Cloud Computing*, vol. 1, no. 1, 2012.

[5] J. Huang and D. Nicol, "Trust mechanisms for cloud computing," *Journal of Cloud Computing*, vol. 2, no. 1, 2013.

[6] R. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "Trustcloud: A framework for accountability and trust in cloud computing," in *Services (SERVICES), 2011 IEEE World Congress on*, July 2011, pp. 584–588.

[7] T. Noor and Q. Sheng, "Credibility-based trust management for services in cloud environments," in *Service-Oriented Computing*, ser. Lecture Notes in Computer 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. Ågotnes, 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.

[11] 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.

[12] 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.

[13] A. Das and M. Islam, "Securedtrust: A dynamic trust computation model for secured communication in multiagent systems," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 2, pp. 261–274, March 2012.

[14] A. K. Despotovic Z, "Maximum likelihood estimation of peers? performances in p2p networks," in *Proceedings of the 2nd workshop on the economics of peer-to-peer systems*, 2004.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th International Conference on World Wide Web*, ser. WWW '03. New York, NY, USA: ACM, 2003, pp. 640–651.

[16] 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.

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

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

[19] 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.

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

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

[22] 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.

[23] T. G. Papaioannou and G. D. Stamoulis, "An incentives' mechanism promoting truthful feedback in peer-to-peer systems," in *Proceedings of 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.

[24] J. Witkowski, "Truthful feedback for sanctioning reputation mechanisms," *CoRR*, vol. abs/1203.3527, 2012.

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

[26] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi, "Embedding virtual topologies in networked clouds," in *Proceedings of the 6th International Conference on Future Internet Technologies*, ser. CFI '11. New York, NY, USA: ACM, 2011, pp. 26–29.

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

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

[29] 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 and Architectures*, ser. VISA '09. New York, NY, USA: ACM, 2009, pp. 81–88.

[30] 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.

[31] R. Ricci, C. Alfeld, and J. Lepreau, "A solver for the network testbed mapping problem," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 65–81, Apr. 2003.

[32] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *INFOCOM 2009, IEEE*, April 2009, pp. 783–791.

[33] X. A. Sandra Herker, Ashiq Khan, "Survey on survivable virtual network embedding problem and solutions," *ICNS 2013 : The Ninth International Conference on Networking and Services*, 2013.

[34] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an ip backbone," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, March 2004, pp. 2307–2317 vol.4.

[35] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 350–361, Aug. 2011.

[36] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable wdm mesh networks," *Lightwave Technology, Journal of*, vol. 21, no. 4, pp. 870–883, April 2003.

[37] T. Guo, N. Wang, K. Moessner, and R. Tafazolli, "Shared backup network provision for virtual network embedding," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[38] H. Yu, V. Anand, C. Qiao, and H. Di, "Migration based protection for virtual infrastructure survivability for link failure," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, March 2011, pp. 1–3.

[39] J. Shamsi and M. Brockmeyer, "Qosmap: Achieving quality and resilience through overlay construction," in *Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on*, May 2009, pp. 58–67.

[40] H. Yu, V. Anand, C. Qiao, and G. Sun, "Cost efficient design of survivable virtual infrastructure to recover from facility node failures," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–6.

[41] C. Qiao, B. Guo, S. Huang, J. Wang, T. Wang, and W. Gu, "A novel two-step approach to surviving facility failures," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, March 2011, pp. 1–3.

[42] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec 2010, pp. 1–6.

[43] Q. Hu, Y. Wang, and X. Cao, "Location-constrained survivable network virtualization," in *Sarnoff Symposium (SARNOFF), 2012 35th IEEE*, May 2012, pp. 1–5.

[44] W.-L. Yeow, C. Westphal, and U. Kozat, "Designing and embedding reliable virtual infrastructures," in *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, ser. VISA '10. New York, NY, USA: ACM, 2010, pp. 33–40.

[45] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive virtual network provisioning," in *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, ser. VISA '10. New York, NY, USA: ACM, 2010, pp. 41–48.

[46] M. Rahman and R. Boutaba, "Svne: Survivable virtual network embedding algorithms for network virtualization," *Network and Service Management, IEEE Transactions on*, vol. 10, no. 2, pp. 105–118, June 2013.

[47] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: A new approach to the design of wdm-based networks," *IEEE J.Sel. A. Commun.*, vol. 20, no. 4, pp. 800–809, Sep. 2006.

[48] X. Sun, G. Chang, and F. Li, "A trust management model to enhance security of cloud computing environments," in *Networking and Distributed Computing (ICNDC), 2011 Second International Conference on*, Sept 2011, pp. 244–248.

[49] M. Wang, G. Wang, J. Tian, H. Zhang, and Y. Zhang, "An accurate and multi-faceted reputation scheme for cloud computing," *Procedia Computer Science*, vol. 34, pp. 466 – 473, 2014, the 9th International Conference on Future Networks and Communications (FNC'14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops.

[50] J. Sidhu and S. Singh, "Compliance based trustworthiness calculation mechanism in cloud environment," *Procedia Computer Science*, vol. 37, pp. 439 – 446, 2014, the 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)/ The 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2014)/ Affiliated Workshops.

[51] M. Macías and J. Guitart, "Cheat-proof trust model for cloud computing markets," in *Proceedings of the 9th International Conference on Economics of Grids, Clouds, Systems, and Services*, ser. GECON'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 154–168.

[52] T. H. Noor, Q. Z. Sheng, S. Zeadally, and J. Yu, "Trust management of services in cloud environments: Obstacles and solutions," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 12:1–12:30, Jul. 2013.

[53] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 199–212. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653687

[54] T. Zhang, Y. Zhang, and R. B. Lee, "Memory dos attacks in multi-tenant clouds: Severity and mitigation," *CoRR*, vol. abs/1603.03404, 2016. [Online]. Available: http://arxiv.org/abs/1603.03404

[55] H. AlJahdali, A. Albatli, P. Garraghan, P. Townend, L. Lau, and J. Xu, "Multi-tenancy in cloud computing," in *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, April 2014, pp. 344–351.

Subhasis Thakur Dr. Subhasis Thakur received his Ph.D from Griffith University, Australia in 2013. He has worked as research fellow at the University of Liverpool, the University of L'Aquila and the National University of Ireland. His research interest includes multi-agent systems, game theory and cloud computing.

John G. Breslin Dr John Breslin is a Senior Lecturer in Electronic Engineering and Director of TechInnovate at NUI Galway. He is also Co-Principal Investigator at the Insight Centre for Data Analytics at NUI Galway (formerly DERI), where he leads the Unit for Social Semantics. He created the SIOC framework, implemented in hundreds of applications on tens of thousands of websites. He has written 175 peer-reviewed publications and co-authored the books The Social Semantic Web and Social Semantic Web Mining. He is co-founder of boards.ie, adverts.ie, and the Galway City Innovation District / PorterShed. He is an advisor to AYLIEN, BuilderEngine and Pocket Anatomy. He has won various best paper awards (SEMANTICS, ICEGOV, ESWC, PELS) and two IIA Net Visionary Awards. He is Vice Chair of IFIPs Working Group 12.7 on Social Networking Semantics and Collective Intelligence. Dr Breslin is a Senior Member of the IEEE.