Robust Reputation Management Using Probabilistic Message Passing

Erman Ayday
School of Electrical and Comp. Eng.
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: eayday@gatech.edu

Faramarz Fekri
School of Electrical and Comp. Eng.
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: fekri@ece.gatech.edu

***Abstract*—In a typical reputation management system, after each transaction, the buyer (who receives a service or purchases a product) provides its report/rating about the quality of the seller for that transaction. In such a system, the problem of reputation management is to compute two sets of variables: 1. the (global) reputation parameters of entities who act as sellers, and 2. the trustworthiness parameters of the entities who act as the raters (i.e., buyers). In this paper, for the first time, we introduce an iterative probabilistic method for reputation management. The proposed scheme, referred to as RPM, relies on a probabilistic message passing algorithm in the graph-based representation of the reputation management problem on an appropriately chosen factor graph. In the graph representation of the problem, the sellers and buyers are arranged as two sets of variable and factor nodes, respectively, that are connected via some edges. Then, the reputation and trustworthiness parameters are computed by a fully iterative and probabilistic message passing algorithm between these nodes in the graph. We provide a detailed evaluation of RPM via computer simulations. We observe that RPM iteratively reduces the error in the reputation estimates of the sellers due to the malicious raters. Finally, comparison of RPM with some well-known and commonly used reputation management techniques (e.g., Averaging Scheme, Bayesian Approach and Cluster Filtering) indicates the superiority of the proposed scheme both in terms of robustness against attacks (e.g., ballot-stuffing, bad-mouthing) and computational efficiency.**

I. Introduction

Reputation management is a crucial requirement for most environments wherein entities participate in various transactions and protocols among each other. The consumer (buyer) of the service (or product) often has insufficient information about the service quality of the service provider (seller) before the transaction. Hence, the consumer should take a prior risk before receiving the actual service. This risk puts the consumer into an unprotected position since he has no opportunity to try the service before he receives it. A reputation management mechanism is a promising method to protect the consumer by forming some foresight about the providers before using their services. By using a reputation management scheme, an individual peer's reputation can be formed by the combination of received reports (ratings). After each transaction between the service providers and consumers, the consumers provide (to the central authority) feedbacks in the form of ratings about the service providers. The central authority collects the reports and updates the reputations of the service providers. Reputation management mechanisms, on the other hand, open up new vulnerabilities as the consumers may provide unreliable or malicious feedbacks, demonizing the reputations of the service providers unfairly. Therefore, a reputation management mechanism has two main goals: 1. computing the quality of the peers (referred to as the *service providers* or SPs hereafter) who provide a service or sell a product by using the feedbacks from the peers (referred to as the *raters* hereafter) who used the service or purchased the product, and 2. determining the trustworthiness of the raters by analyzing their feedback about the SPs. Hence, the success of a reputation scheme depends on the robustness of the mechanism to accurately evaluate the reputations of the SPs and the trustworthiness of the raters.

Current reputation management schemes are vulnerable to sophisticated attacks since none of these schemes are designed considering the noise and the incomplete information in the system. The objective of this work is to introduce the first application of iterative probabilistic algorithms in the design and evaluation reputation management systems. Our work on the reputation systems is inspired by earlier work on graph-based iterative probabilistic decoding of turbo codes and low-density parity-check codes, the most powerful error-control codes known. These probabilistic and iterative decoding algorithms are shown to perform at error rates near what can be achieved by the optimal scheme, maximum likelihood decoding, while requiring far less computational complexity (i.e., linear in the length of the code). We believe that the significant benefits offered by the probabilistic message passing algorithms [1] can be tapped in to benefit the field of reputation systems. The reputation management problem can be viewed as finding the marginal probability distributions of the variables representing the global reputations of the SPs, given the observed data (i.e., evidence). This problem, however, cannot be solved in a large-scale reputation systems, because the number of terms grow exponentially with the number of raters and SPs. The key role of the probabilistic message passing algorithm is that we can use it to compute those marginal distributions in the complexity that grows only linearly with the number of nodes. Therefore, we introduce the "Robust Reputation Management Using Probabilistic Message Passing" (RPM).

The proposed RPM relies on a graph-based representation of an appropriately chosen factor graph for reputation systems. In this representation, SPs and raters (consumers) are arranged as two sets of variable and factor nodes that are connected via some edges. The reputation values of the SPs are computed by message passing between nodes in the graph until the scheme converges. We show that RPM iteratively reduces the error in the reputation values of SPs due to the malicious raters with a high probability. Although we present the proposed algorithm as a global reputation system, it can be applied to various applications from personalized reputation systems to ad-hoc networks. The main contributions of our work are summarized in the following.

- We introduce the first application of graph-based iterative probabilistic algorithms in the design and evaluation of reputation management systems. We use an iterative and probabilistic message passing algorithm as the core of our proposed scheme.
- The proposed iterative algorithm computes the reputation values of the SPs with a small error in a short amount of time in the presence of attackers. Thus, it is a robust and efficient methodology for detecting and filtering out malicious ratings.
- The proposed RPM significantly outperforms the existing and commonly used reputation management techniques in the presence of attackers.

The rest of this paper is organized as follows. In the rest of this section, we summarize the related work. In Section II, we describe the proposed RPM in detail. Next, in Section III, we evaluate RPM via computer simulations and compare RPM with the existing and commonly used reputation management schemes. Finally, in Section IV, we conclude our paper.

A. Related Work

Several works in the literature have focused so far on building reputation management mechanisms [2]–[5][1]. The most famous and primitive global reputation system is the one that is used in eBay. Other well-known web sites such as Amazon, Epinions, and AllExperts use a more advanced reputation mechanism than eBay. Their reputation mechanisms mostly compute the average (or weighted average) of the ratings received for a product (or a peer) to evaluate the global reputation of a product (or a peer). Hence, these schemes are vulnerable to collaborative attacks by malicious peers. Use of the Bayesian Approach is also proposed in [6]. In these systems, the *a posteriori* reputation value of a peer is computed combining its *a priori* reputation values with the new ratings received for the peer. In [7], authors proposed to use the *Cluster Filtering* method for reputation systems to distinguish between the reliable and unreliable raters. Finally, in our previous work, we proposed an algebraic iterative algorithm [8] for reputation systems (referred to as ITRM) and showed the benefit of using iterative algorithms for reputation management. Here, we expand this work and introduce a fully probabilistic, message passing based approach. We compare our proposed scheme with the existing schemes (including ITRM [8]) in Section III-B and show its superior performance (i.e., accuracy and robustness against attacks).

II. Robust Reputation Management Using Probabilistic Message Passing

We assume two different sets[2] in the system: i) the set of SPs, and ii) the set of raters. We let TR_j be the global reputations of the SPs. Further, TR_{ij} represents the i^{th} rater's rating about the j^{th} SP, and R_i denotes the (rating) trustworthiness of the i^{th} rater (i.e., the amount of confidence that the central authority has about the correctness of any rating provided by the rater i).

To describe the reputation system, we arrange the collection of raters and SPs together with their associated relations (i.e., the ratings) as a bipartite (or factor) graph, as in Fig. 1. In this representation, each rater corresponds to a *check vertex* (or factor node), shown as a square and each SP is represented by a *bit vertex* (or variable node) shown as a hexagon. Further, each rating is represented by an edge from a check-vertex to a bit-vertex. Hence, if a rater i has a rating about the j^{th} SP, we place an edge with value TR_{ij}[3] from the i^{th} check-vertex to the bit-vertex representing the j^{th} SP.

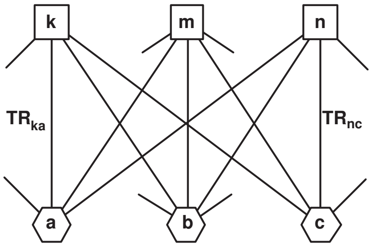

Fig. 1: Setup of the scheme.

In each iteration, the probabilities are exchanged over the edges of the bipartite graph in Fig. 1, estimating probabilistically the reputation values of the SPs as well as the confidence (i.e., trustworthiness) on the raters. For simplicity of presentation, we assume that the rating values are either 0 or 1. The extension in which rating values can take any real number can be developed similarly (we implemented RPM for both cases and illustrate its performance in Section III-B). We consider slotted time throughout this discussion. At each time-slot, the iterative algorithm is executed using the input parameters R_i and TR_{ij} to obtain the global reputation values of the SPs (e.g., TR_j).

Details of RPM may be described by the following procedure at the L^{th} time-slot. Let $TR_j^{(\nu)}$ be the value of the bit-vertex at the iteration ν of the RPM algorithm. For simplicity, we consider the network in Fig. 1 and describe message exchange between rater k and SP a. We represent the set of neighbors of SP a and rater k as $\mathbf{N_a}$ and $\mathbf{N_k}$, respectively (neighbors of a SP are the raters who rated it and neighbors of a rater are the SPs whom it rated). Further, let $\Xi = \mathbf{N_a}\backslash\{k\}$ and $\Delta = \mathbf{N_k}\backslash\{a\}$, where $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{|\Xi|}\}$ and $\Delta = \{\delta_1, \delta_2, \ldots, \delta_{|\Delta|}\}$ (the notation $\mathbf{N_a}\backslash\{k\}$ denotes the set of all neighbors of SP a except the neighbor k).

We denote the messages from SPs to raters and from raters to SPs as μ and λ, respectively. The message $\mu_{a\to k}^{(i)}$ is a vector $[\mu_{a\to k}^{(i)}(0), \mu_{a\to k}^{(i)}(1)]$ denoting the probability of TR_a being zero or one at the i^{th} iteration. Obviously, $\mu_{a\to k}^{(i)}(1) = 1 - \mu_{a\to k}^{(i)}(0)$. On the other hand, $\lambda_{k\to a}^{(i)}$ denotes the belief (confidence) of rater k (at the i^{th} iteration) that the TR_a value is equal to TR_{ka}. This resembles the belief/plausibility concept of the Dempster-Shafer Theory [9]. For example, given $TR_{ka} = 1$, we denote $\lambda_{k\to a}^{(i)}$ as the belief of the k^{th} rater (at the i^{th} iteration) that the TR_a value is one. Further, since there is no evidence contrary to the hypothesis $TR_a = 1$, the plausibility that $TR_a = 1$ is

[1] The list of references is not exhaustive due to the page limit.

[2] Sets are not necessarily disjoint.

[3] TR_{ij} value between rater i and SP j is the aggregate of all past and present ratings between these two peers. Further, one may include a fading factor while updating the values of the edges to account for the freshness of the ratings.

equal to one. Thus, $(1-\lambda^{(i)}_{k\to a})$ corresponds to the uncertainty in the belief of rater k. To remove this uncertainty, we distribute the uncertainty uniformly between two outcomes (one and zero). Hence, given $TR_{ka}=1$, in the eyes of the k^{th} rater, TR_a value is equal to one with probability $(\lambda^{(i)}_{k\to a}+(1-\lambda^{(i)}_{k\to a})/2)$, and zero with probability $(1-\lambda^{(i)}_{k\to a})/2$. We note that a similar statement holds for the case when $TR_{ka}=0$.

Therefore, for SP a, we calculate the probability of TR_a being one or zero by multiplying all probabilities it received from its neighbors excluding the rater k. Hence, the message $\mu^{(j)}_{a\to k}$ from SP a to rater k at the j^{th} iteration is given by

$$\mu^{(j)}_{a\to k}(1)=\frac{\prod\limits_{i\in\Xi}\Pr\left(TR_a=1|TR_{ia},\lambda^{(j-1)}_{i\to a}\right)}{\sum\limits_{\varsigma\in\{0,1\}}\prod\limits_{i\in\Xi}\Pr\left(TR_a=\varsigma|TR_{ia},\lambda^{(j-1)}_{i\to a}\right)}, \tag{1}$$

where

$$\Pr\left(TR_a=1|TR_{ia},\lambda^{(j-1)}_{i\to a}\right)=$$
$$TR_{ia}\left(\lambda^{(j-1)}_{i\to a}+\frac{1-\lambda^{(j-1)}_{i\to a}}{2}\right)+(1-TR_{ia})\left(\frac{1-\lambda^{(j-1)}_{i\to a}}{2}\right) \tag{2a}$$

$$\Pr\left(TR_a=0|TR_{ia},\lambda^{(j-1)}_{i\to a}\right)=$$
$$TR_{ia}\left(\frac{1-\lambda^{(j-1)}_{i\to a}}{2}\right)+(1-TR_{ia})\left(\lambda^{(j-1)}_{i\to a}+\frac{1-\lambda^{(j-1)}_{i\to a}}{2}\right) \tag{2b}$$

This computation is repeated for every neighbors of each SP. This finishes the first half of the j^{th} iteration. During the second half, we compute the messages $\lambda^{(j)}$ between each rater and its neighbors. For rater k, we calculate its confidence on its ratings by calculating the deviation in its ratings TR_{ki} ($i\in\Delta$) based on the messages $\mu^{(j)}_{i\to k}$ ($i\in\Delta$) it received from its neighbors except the neighbor a. Thus, the message from rater k to SP a at the j^{th} iteration is formed (by considering all realizations of the messages from its neighbors) as

$$\lambda^{(j)}_{k\to a}=1-\frac{1}{|\Delta|}\Big\{\sum_{\alpha_1\in\{0,1\}}\cdots\sum_{\alpha_{|\Delta|}\in\{0,1\}}[|TR_{k\delta_1}-\alpha_1|+ \\ \ldots+|TR_{k\delta_{|\Delta|}}-\alpha_{|\Delta|}|]\prod_{x\in\Delta}\mu^{(j)}_{x\to k}\left(h(x)\right)\Big\}, \tag{3}$$

where

$$h(x)=\alpha_i \text{ if } x=\delta_i \text{ for } \delta_i\in\Delta. \tag{4}$$

Equation (3) can be interpreted as one minus the average inconsistency of rater k obtained using the messages it received from its neighbors (excluding SP a). The algorithm proceeds to the next iteration in the same way as the j^{th} iteration. We clarify that the iterative algorithm starts by computing $\mu^{(1)}_{a\to k}$ in (1). However, the trustworthiness values R_k from the previous execution of RPM are used as initial values for $\lambda^{(0)}_{k\to a}$ in (1), (2a), and (2b).

At the end of each iteration the global reputations and the trustworthiness of raters are calculated using modified (1) and (3), respectively. That is, we use the set $\mathbf{N_a}$ instead of Ξ in (1) to compute $\mu^{(j)}_a(1)$ and $\mu^{(j)}_a(0)$. Then we set $TR^{(j)}_a=\sum_{i=0}^{1}i\mu^{(j)}_a(i)$. Likewise, we use the set $\mathbf{N_k}$ instead of Δ in (3) to compute $\lambda^{(j)}_k$, and then we set $R^{(j)}_k=\lambda^{(j)}_k$. We repeat this to compute the global reputations and trustworthiness of every SP and rater. The iterations stop when the TR_j values converge for every SP.

III. Security Evaluation of RPM

We list the frequently used notations in Table 1.

N_{SP}	The set of service providers (SPs)
N_M	The set of malicious raters
N_R	The set of reliable raters
r_h	Report (rating) given by a reliable rater
r_m	Report (rating) given by a malicious rater
d	Total number of newly generated ratings, per time-slot, per a reliable rater
b	Total number of newly generated ratings, per time-slot, per a malicious rater
$\hat{b}$	Total number of newly generated attacking/malicious ratings, per time-slot, by a malicious rater
Λ	$\hat{b}/b$ (i.e., fraction of attacking ratings per time-slot)

TABLE I: Notations and definitions.

A. Attack Models

We consider the following two major attacks that are common for any reputation management mechanisms:
Bad-mouthing: Malicious raters collude and attack the SPs with the highest reputation by giving low ratings to undermine them. It is also noted that in some applications, bad-mouthing may be originated by a group of selfish peers who attempt to weaken high-reputation providers in the hope of improving their own chances as providers.
Ballot-stuffing: Malicious raters collude to increase the reputation value of peers with low reputations. Just as in bad-mouthing, in some applications, this could be mounted by a group of selfish consumers attempting to favor their allies.

We make the following assumptions for modeling the adversary. We assumed that the malicious raters initiate bad-mouthing[4]. Further, all the malicious raters collude and attack the same subset Γ of SPs in each time-slot (which represents the strongest attack), by rating those SPs as r_m. In other words, we denote by Γ the set of size $\hat{b}$ in which every victim SP has one edge from each of the malicious raters. The subset Γ is chosen to include those SPs who have the highest reputation values but received the lowest number of ratings from the non-malicious raters (assuming that the attackers have this information)[5]. We note that this attack scenario also represents the RepTrap attack in [10] which is shown to be a strong attack. To the advantage of malicious raters, we assumed that a total of T time-slots had passed since the initialization of the system and a fraction of the existing raters change behavior and become malicious after T time-slots. In other words, malicious raters behaved like reliable raters and increased their trustworthiness values before mounting

[4] Even though we use the bad-mouthing attack, similar counterpart results hold for ballot-stuffing and combinations of bad-mouthing and ballot-stuffing.

[5] Although it may appear unrealistic for some applications, availability of such information for the malicious raters would imply the worst case scenario.

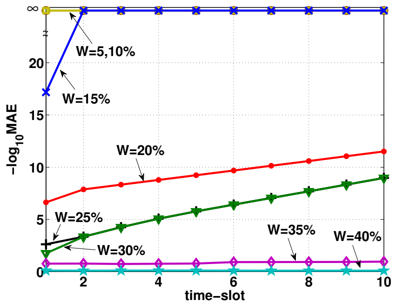

Fig. 2: MAE performance of RPM versus time when W of the existing raters become malicious in RepTrap.

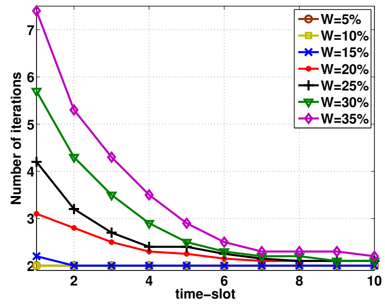

Fig. 3: The average number of required iterations for convergence versus time when W of the existing raters become malicious in RepTrap.

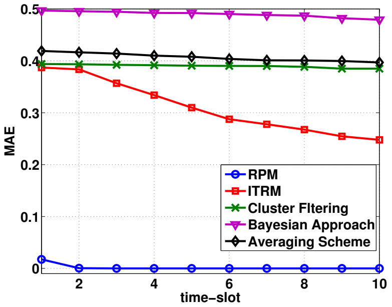

Fig. 4: MAE performance of various schemes when 30% of the existing raters become malicious in RepTrap.

their attacks at the $(T+1)^{th}$ time-slot. We will evaluate the performance for the time-slot $(T+1)$.

B. Simulations

We compared the performance of RPM with three well-known and commonly used reputation management schemes: 1) *The Averaging Scheme* (which is widely used as in eBay), 2) *Bayesian Approach* [6], and 3) *Cluster Filtering* [7]. Further, we compared RPM with our previous work on iterative trust and reputation management (referred to as ITRM) [8] to show the benefit of using probabilistic message passing algorithm.

Throughout the simulations, we adopted the following models for various peers involved in the reputation system. We acknowledge that although the models are not inclusive of every scenario, they are good illustrations to present our results. We assumed that the quality of each SP remains unchanged during time-slots. Ratings generated by the non-malicious raters are distributed uniformly among the SPs (i.e., their ratings/edges in the graph representation are distributed uniformly among SPs). Further, we assumed that d is a random variable with Yule-Simon distribution, which resembles the power-law distribution used in modeling online systems [11], with the probability mass function $f_d(d;\rho) = \rho B(d, \rho+1)$, where B is the Beta function. Finally, we assumed the adversary model in Section III-A. The parameters we used are $|N_M| + |N_R| = 100$, $|N_{SP}| = 100$, $\rho = 1$, $T = 50$ and $b = 5$. Let $\tilde{TR}_j$ be the actual value of the global reputation of the j^{th} SP. Then, we obtained the performance of RPM, for each time-slot, as the mean absolute error (MAE) $|TR_j - \tilde{TR}_j|$, averaged over all the SPs that are under attack. We note that we start our observations at time slot 1 after the initialization period.

Initially, we assumed that the rating values are either 0 or 1 (where 1 represents a good service quality), all the ratings provided by the malicious raters are malicious (i.e., $\hat{b} = b$), and $r_m = 0$. We further assumed that the rating r_h (provided by the non-malicious raters) is a random variable with Bernoulli distribution, where $Pr(r_h = \tilde{TR}_j) = 0.8$ and $Pr(r_h \neq \tilde{TR}_j) = 0.2$, and $\tilde{TR}_j$ is the actual value of the global reputation of the j^{th} SP. First, we evaluated the MAE performance of RPM for different fractions of malicious raters ($W = \frac{|N_M|}{|N_M|+|N_R|}$), at different time-slots (measured since the attack is applied) in Fig. 2[6]. We observed that the proposed RPM provides significantly low errors for up to $W = 30\%$ malicious raters. We further showed the average number of required iterations of RPM at each time-slot in Fig. 3. We concluded that the average number of iterations for RPM decreases with time and with decreasing fraction of malicious raters. Finally, we compared the MAE performance of RPM with the other schemes for the RepTrap attack. Figure 4 illustrates the comparison of RPM with the other schemes for bad-mouthing when the fraction of malicious raters (W) is 30%. It is clear that RPM outperforms all the other techniques significantly.

We also evaluated the performance of RPM when the malicious raters provide both reliable and malicious ratings to mislead the algorithm. We assumed binary rating values (0 and 1) and the adversary model in Section III-A with $r_m = 0$. In Fig. 5, we illustrate the MAE performance of RPM for this attack for $W = 30\%$ and different $\Lambda = \hat{b}/b$ values. We observed that as the malicious raters attack with less number of edges (for low values of $\hat{b}$), it requires more time slots to undo their impact using RPM. Further, in Fig. 6, we show the change in the average trustworthiness of malicious raters versus time for varying Λ when $W = 30\%$. We observed that as Λ decreases, the trustworthiness values of the malicious raters decrease at slower rates compared to higher values of Λ. Thus, the malicious raters stay under cover when they attack to small number of SPs. However, this type of an attack limits the malicious raters' ability to make a serious impact. To illustrate this, we observed the gain and loss of the adversary for different attack strategies. The gain of an adversary is proportional to the MAE and the number of victim SPs. Therefore, we defined the gain of the adversary as (MAE $\times\ \Lambda$). On the other hand, we defined the loss of the adversary as $(\Delta R_M/R_R)$, where R_R is the average trustworthiness value of the reliable raters, R_M is the average trustworthiness value of the malicious rates, and $\Delta R_M = R_R - R_M$. Thus, we illustrate the gain and loss of the adversary in Fig. 7 for varying Λ and various W. We observed that the adversary only has a positive gain (i.e., gain>loss) for $W = 40\%$, which is a significantly high adversary level. We note that for different values of Λ and W, we observed that RPM still keeps its superiority over the other schemes.

We also simulated the same attack scenario when ratings are integers from the set $\{1, \ldots, 5\}$ instead of binary values, and $\Lambda = 1$. We assumed that the rating r_h is a random variable

[6] The plots in Figs. 2, 3, 4, 5, 6, 7 and 8 are shown from the time-slot the adversary introduced its attack.

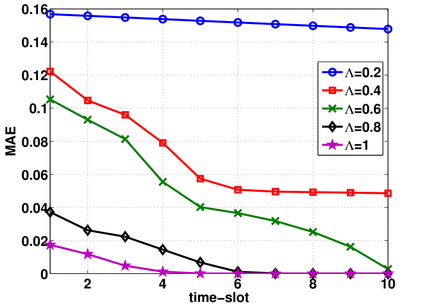

Fig. 5: MAE performance of RPM versus time for varying Λ when 30% of the existing raters become malicious in RepTrap.

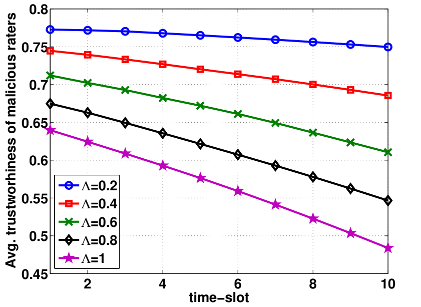

Fig. 6: Average trustworthiness of malicious raters versus time for varying Λ when 30% of the existing raters become malicious in RepTrap.

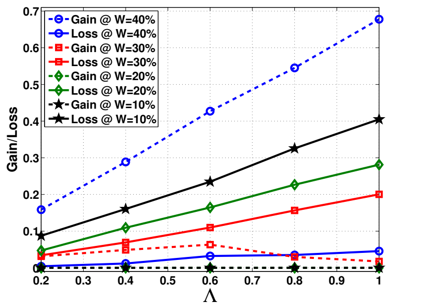

Fig. 7: Gain and loss of the adversary versus Λ when W of the existing raters become malicious in RepTrap.

with folded normal distribution (mean $\tilde{TR}_j$ and variance 0.5), however, it takes only discrete values from 1 to 5. The malicious raters do not deviate very much from the actual $\tilde{TR}_j = 5$ values to remain undercover (while still attacking) as many time-slots as possible. Hence, malicious raters choose SPs from Γ and rate them as $r_m = 4$[7]. We compared the MAE performance of RPM with the other schemes in Fig. 8 and observed that RPM outperforms all the other techniques significantly.

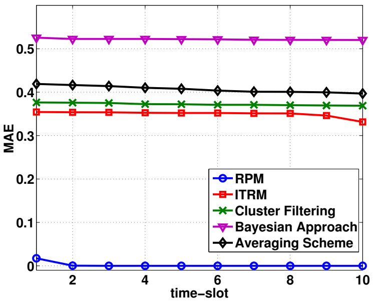

Fig. 8: MAE performance of various schemes when 30% of the existing raters become malicious and rating values are nonbinary, i.e., from $\{1, \ldots, 5\}$ in RepTrap.

From these simulation results, we conclude that RPM significantly outperforms the Averaging Scheme, Bayesian Approach and Cluster Filtering in the presence of attackers. We identify that ITRM (i.e., our algebraic iterative scheme) is the closest in performance to RPM. This emphasizes the robustness of using iterative algorithms for reputation management. Finally, assuming $K = |N_M| + |N_R|$ raters and $|N_{SP}|$ SPs, we obtained the computational complexity of RPM as $\mathbf{max}(\mathrm{O}(cK), \mathrm{O}(cN_{SP}))$ in the number of multiplications, where c is a small number representing the average number of ratings per rater. On the other hand, Cluster Filtering suffers quadratic complexity versus number of raters (or SPs).

IV. Conclusion

In this paper, we introduced the Robust Reputation Management Using Probabilistic Message Passing (RPM). Our work is motivated by the prior success of the probabilistic message passing algorithms on decoding of low-density parity-check codes. RPM is a graph-based reputation management system in which service providers and raters are arranged as two sets of variable and factor nodes and the reputation values of SPs are computed by message passing between these nodes in the graph until the convergence. The proposed RPM is a robust mechanism to evaluate the quality of the service of the SPs from the ratings received from the raters. Moreover, it effectively evaluates the trustworthiness of the raters. We showed the robustness of RPM using computer simulations. We also compared RPM with some well-known reputation management schemes and showed the superiority of our scheme both in terms of robustness against various attacks and efficiency.

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

[2] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems: facilitating trust in internet interactions," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[3] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," *CIKM '01: Proceedings of the 10th International Conference on Information and Knowledge Management*, pp. 310–317, 2001.

[4] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," *CCS '02: Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 207–216, 2002.

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

[6] A. Whitby, A. Josang, and J. Indulska, "Filtering out unfair ratings in Bayesian reputation systems," *AAMAS '04: Proceedings of the 7th International Workshop on Trust in Agent Societies*, 2004.

[7] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," *EC '00: Proceedings of the 2nd ACM Conference on Electronic commerce*, pp. 150–157, 2000.

[8] E. Ayday, H. Lee, and F. Fekri, "An iterative algorithm for trust and reputation management," *ISIT '09: Proceedings of IEEE International Symposium on Information Theory*, 2009.

[9] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, N.J., 1976.

[10] Y. Yang, Q. Feng, Y. L. Sun, and Y. Dai, "RepTrap: a novel attack on feedback-based reputation systems," *SecureComm '08: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pp. 1–11, 2008.

[11] F. Slanina and Y. C. Zhang, "Referee networks and their spectral properties," *Acta Physica Polonica B*, vol. 36, pp. 2797–+, Sep. 2005.

[7] We also tried higher deviations from the $\tilde{TR}_j$ value and observed that the malicious raters were easily contained by RPM.