Iterative Trust and Reputation Management Using Belief Propagation

Erman Ayday, *Student Member*, *IEEE*, and Faramarz Fekri, *Senior Member*, *IEEE*

Abstract—In this paper, we introduce the first application of the belief propagation algorithm in the design and evaluation of trust and reputation management systems. We approach the reputation management problem as an inference problem and describe it as computing marginal likelihood distributions from complicated global functions of many variables. However, we observe that computing the marginal probability functions is computationally prohibitive for large-scale reputation systems. Therefore, we propose to utilize the belief propagation algorithm to efficiently (in linear complexity) compute these marginal probability distributions; resulting a fully iterative probabilistic and belief propagation-based approach (referred to as BP-ITRM). BP-ITRM models the reputation system on a factor graph. By using a factor graph, we obtain a qualitative representation of how the consumers (buyers) and service providers (sellers) are related on a graphical structure. Further, by using such a factor graph, the global functions factor into products of simpler local functions, each of which depends on a subset of the variables. Then, we compute the marginal probability distribution functions of the variables representing the reputation values (of the service providers) by message passing between nodes in the graph. We show that BP-ITRM is reliable in filtering out malicious/unreliable reports. We provide a detailed evaluation of BP-ITRM via analysis and computer simulations. We prove that BP-ITRM iteratively reduces the error in the reputation values of service providers due to the malicious raters with a high probability. Further, we observe that this probability drops suddenly if a particular fraction of malicious raters is exceeded, which introduces a *threshold* property to the scheme. Furthermore, comparison of BP-ITRM 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 in terms of robustness against attacks (e.g., ballot stuffing, bad mouthing). Finally, BP-ITRM introduces a linear complexity in the number of service providers and consumers, far exceeding the efficiency of other schemes.

Index Terms—Trust and reputation management, belief propagation, iterative algorithms, bad mouthing, ballot stuffing, online services, e-commerce.

1 Introduction

Trust and reputation are crucial requirements for most environments wherein entities participate in various transactions and protocols among each other. In most online service systems, the consumer of the service (e.g., the buyer) has no choice but to rely on the reputation of the service provider (e.g., the seller) based on the latter's prior performance. A reputation management mechanism is a promising method to protect the consumer (buyer) of the service by forming some foresight about the service providers (sellers) before using their services (or purchasing their products). By using a reputation management scheme, an individual peer's reputation can be formed by the combination of received reports (ratings). Hence, after each transaction, a party who receives the service or purchases the product (referred to as the rater) provides (to the central authority) its report about the quality of the service provided (or the quality of the product purchased) for that transaction. The central authority collects the reports and updates the reputations of the service providers (sellers). Therefore, the main goal of a reputation mechanism is to determine the service (product), qualities of the service providers (sellers), and the trustworthiness of the raters based on their reports about the service qualities. Hence, the success of a reputation scheme depends on the robustness of the mechanism to accurately evaluate the reputations of the service providers (sellers) and the trustworthiness of the raters.

Trust and reputation mechanisms have various application areas from online services to mobile ad-hoc networks (MANETs) [1], [2], [3], [4]. Most well-known commercial websites such as eBay, Amazon, Netflix, and Google use some types of reputation mechanisms. Hence, it is foreseeable that the social web is going to be driven by these reputation systems. Despite recent advances in reputation systems, there is yet a need to develop reliable, scalable, and dependable schemes that would also be resilient to various ways a reputation system can be attacked. Moreover, new and untested applications open up new vulnerabilities, and hence, requiring specific solutions for reputation systems.

As in every security system, trust and reputation management systems are also subject to malicious behaviors. Malicious raters may attack particular service providers (sellers) in order to undermine their reputations while they help other service providers by boosting their reputations. Similarly, malicious service providers (sellers) may provide good service qualities (or sell high-quality products) for certain customers (buyers) in order to keep their reputations high while cheating the other customers. Moreover, malicious raters (or service providers) may collaboratively mount sophisticated attacking strategies by

- *The authors are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta 30332, GA. E-mail: eayday@gatech.edu, fekri@ece.gatech.edu.*

Manuscript received 19 Feb. 2010; revised 25 Aug. 2011; accepted 14 Dec. 2011; published online 23 Dec. 2011.
Recommended for acceptance by V. Varadharajan.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2010-02-0028.
Digital Object Identifier no. 10.1109/TDSC.2011.64.

Published by the IEEE Computer Society

exploiting their prior knowledge about the reputation mechanism. Hence, building a resilient trust and reputation management system that is robust against malicious activities becomes a challenging issue.

In this paper, we introduce the first application of the belief propagation algorithm in the design and evaluation of trust and reputation management systems. In our previous work, inspired by our earlier work on iterative decoding of error-control codes in the presence of stopping sets [5], [6], [7], we proposed an algebraic iterative algorithm [8] for reputation systems (referred to as ITRM) and showed the benefit of using iterative algorithms for trust and reputation management. Here, we expand this work and introduce a fully probabilistic approach based on the belief propagation algorithm. Different from our previous work [8], in this paper, we view the reputation management problem as an inference problem and describe it as computing marginal likelihood distributions from complicated global functions of many variables. Further, we utilize the belief propagation algorithm to efficiently (in linear complexity) compute these marginal probability distributions. The work is inspired by earlier work on graph-based iterative probabilistic decoding of turbo codes and low-density parity-check (LDPC) codes, the most powerful practically decodable error-control codes known. These 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 iterative probabilistic algorithms can be also tapped in to benefit the field of reputation systems. In iterative decoding of LDPC, every check vertex (in the graph representation of the code) has some opinion of what the value of each bit vertex should be. The iterative decoding algorithm would then analyze the collection of these opinions to decide, in each iteration, what value to assign for the bit vertex under examination. Once the values of the bit vertices are estimated, in the next iteration, those values are used to determine the satisfaction of the check vertex values. The contribution of our research stems from the observation that a similar approach can be adapted to determine the reputations of the service providers (sellers) as well as the trustworthiness of the raters. Furthermore, the analysis of reputation systems resembles that of the code design problem. In LDPC, one of the goals is to find the decoding error for a fixed set of check constraints. Similarly, in the reputation system, our goal is to specify the regions of trust for the set of the system parameters. A region of trust is the range of parameters for which we can confidently determine the reputation values within a given error bound. We acknowledge, however, that we have a harder problem in the case of reputation systems as the adversary dynamics is far more complicated to analyze than the erasure channel in the coding problem.

We introduce the “Belief Propagation-based Iterative Trust and Reputation Management Scheme” (BP-ITRM). Belief propagation [9], [10], [11] is a message passing algorithm for performing interface on graphical models such as Bayesian networks or Markov random fields. It is used for computing marginal distributions of the unobserved nodes conditioned on the observed ones. Computing marginal distributions is hard in general as it might require summing an exponentially large number of terms. Hence, the belief propagation algorithm is usually described in terms of operations on factor graphs. The factor graph representation of the reputation systems turned out to be a bipartite graph, where the service providers (sellers) and consumers (buyers) are arranged as two sets of variable and factor nodes that are connected via some edges. The reputation can be computed by message passing between nodes in the graph. In each iteration of the algorithm, all the variable nodes (sellers), and subsequently all the factor nodes (buyers), pass new messages to their neighbors until the reputation value converges. We note that in the rest of this paper, we use the word “message” as virtual term. The exchange of messages are not between the actual sellers and buyers; all messages between the nodes in the graph (i.e., between the variable and factor nodes) are formed by the algorithm that is ran in the central authority. We show that the proposed iterative scheme is reliable (in filtering out malicious/unreliable reports). Further, we prove that BP-ITRM iteratively reduces the error in the reputation values of service providers due to the malicious raters with a high probability. We observe that this probability suddenly drops if the fraction of malicious raters exceeds a threshold. Hence, the scheme has a *threshold* property.

The proposed reputation management algorithm can be utilized in well-known online services such as eBay or Epinions. In eBay, each seller-buyer pair rate each other after a transaction. Thus, BP-ITRM can be used in eBay to compute the reputation values of the sellers and buyers along with the trustworthiness values of the peers in their ratings. Epinions, on the other hand, is a product review site in which users can rate and review items. Users can also give ratings to the reviews. Hence, the ratings of members on a review and on a product are considered separately. BP-ITRM can be utilized in such an environment to compute the reputations of the reviewers based on the ratings given by the users on the reviews. Although we present the proposed algorithm as a centralized approach, it can also be applied to decentralized systems such as ad hoc networks and P2P systems to compute the reputations of the nodes in the network. As an example, we applied ITRM, our algebraic but iterative reputation management system, to delay tolerant networks [12] in a decentralized environment.

The rest of this paper is organized as follows: in the rest of this section, we summarize the related work, list the contributions of this work and describe the belief propagation algorithm. In Section 2, we describe the proposed BP-ITRM in detail. Next, in Section 3, we mathematically model and analyze BP-ITRM. Further, we support the analysis via computer simulations, compare BP-ITRM with the existing and commonly used trust management schemes, and discuss the computational complexity of the proposed scheme. Finally, in Section 4, we conclude our paper.

1.1 Related Work

Several works in the literature have focused so far on building reputation-management mechanisms [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. We may classify reputation mechanisms for centralized systems as 1) global reputation systems, where the reputation of a service provider (seller) is based on the reports form general users [26], [27], and 2) personalized reputation systems, where the reputation of a service provider (seller) is determined based on the reports of a group of particular users, which may be different in the eyes of different users [28], [29]. We note that our work falls under the category of global reputation systems. The most famous and primitive global reputation system is the one that is used in eBay. In eBay, each seller-buyer pair rate each other after a transaction, and the total rating of a peer is the sum of the individual ratings it received from the other peers. It is shown in [30] that, even this simple reputation mechanism provides the sellers with high reputation to sell their items more than the other sellers. On the other hand, since eBay's reputation scheme weights all individual ratings equally, the unfair ratings (the ones coming from the unreliable peers) are not filtered, effecting the reputation values of the sellers significantly. 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. Google's PageRank algorithm [26] can also be considered as a global reputation systems. This algorithm does not require the participation of the users to rank the web pages. Basically, the web page with more back links (links that point to it) is considered to be more important (has higher rank) than the one with fewer back links. PageRank algorithms is also modified and used in social networks for the reputation of the peers [31], [32]. Use of the Bayesian Approach is also proposed in [27], [33]. 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. Further, a threshold method is used to determine and update the report reliability of the rater peers. Finally, [29] proposed to use the *Cluster Filtering* method [34] for reputation systems to distinguish between the reliable and unreliable raters. We compare our proposed scheme with the existing schemes (in Section 3.3) and show its superior performance (i.e., accuracy and robustness against attacks).

Personalized reputation systems are also widely studied for different purposes. In Histos [28], the central node (server) keeps all the ratings between the peers and generates a graph to calculate the ratings of each peer for the other peers. However, each update of this graph requires a lot of computations. Hence, this scheme has high-computational complexity. The most well-known method that is used to build personal reputations is the *Collaborative Filtering* [35], [36]. Using this method, the predicted rating of a peer i for another peer j (that i has not directly rated) is calculated by the main server using a memory-based algorithm (such as similarity testing [37]) or a model-based algorithm (such as matrix factorization [38]). However, these types of systems have cold start and data sparseness problems which cause them to be vulnerable against malicious behavior.

1.2 Contributions of the Paper

The main contributions of our work are summarized in the following.

1. We introduce the first application of the belief propagation algorithm on trust and reputation management systems.
2. As the core of our trust and reputation management system, we use the belief propagation algorithm which is proven to be a powerful tool on decoding of turbo codes and LDPC codes. Therefore, we introduce a graph-based trust and reputation management mechanism that relies on an appropriately chosen factor graph and computes the reputation values of service providers (sellers) by a message passing algorithm.
3. The proposed iterative algorithm computes the reputation values of the service providers (sellers) accurately (with a small error) in a short amount of time in the presence of attackers. The scheme is also a robust and efficient methodology for detecting and filtering out malicious ratings. Further, the scheme detects the malicious raters with a high accuracy, and updates their trustworthiness accordingly enforcing them to execute low-grade attacks to remain undercover.
4. The proposed BP-ITRM significantly outperforms the existing and commonly used reputation management techniques such as the Averaging Scheme, Bayesian Approach as in [27] and [33], and Cluster Filtering in the presence of attackers.

1.3 Belief Propagation

Belief propagation [9], [10], [11] is a message passing algorithm for performing interface on graphical models (Bayesian networks, Markov random fields). It is a method for computing marginal distributions of the unobserved nodes conditioned on the observed ones. Computing marginal distributions is hard in general as it might require summing an exponentially large number of terms. Hence, belief propagation algorithm is usually described in terms of operations on a factor graph. A factor graph is a bipartite graph containing nodes corresponding to variables and factors with edges between them. A factor graph has a variable node for each variable, a factor node for each function, and an edge connecting a variable node to a factor node if and only if the variable is an argument of function corresponding to the factor node. The marginal distribution of an unobserved node can be computed accurately using the belief propagation algorithm if the factor graph has no cycles. However, the algorithm is still well defined and often gives good approximate results even for the factor graphs with cycles (as it has been observed in decoding of LDPC codes).

Belief propagation is commonly used in artificial intelligence and information theory. It has demonstrated empirical success in numerous applications including LDPC codes, turbo codes, free energy approximation, and satisfiability. In iterative decoding of LDPC, for example, every check vertex (in the graph representation of the code) has some opinion of what the value of each bit vertex should be. The iterative decoding algorithm would then analyze the collection of these opinions to decide, in each iteration, what value to assign for the bit vertex under examination. Once the values of the bit vertices are estimated, in the next iteration, those values are used to determine the satisfaction of the check-vertex values. While the optimal decoding technique of LDPC codes, maximum likelihood (ML) decoding, is an NP problem, belief propagation algorithm provides a very efficient decoding that gets close to the bit error rate (BER) performance of the ML decoding when the code length becomes large. In other words, belief propagation performs at error rates near what can be achieved by the optimal scheme while requiring far less computational complexity. Here, we propose to exploit such benefits in trust and reputation management systems.

2 Belief Propagation for Iterative Trust and Reputation Management

As in every reputation management mechanism, we have two main goals: 1) computing the service quality (reputation) of the peers who provide a service (henceforth referred to as Service Providers or SPs) by using the feedbacks from the peers who used the service (referred to as the raters), and 2) determining the trustworthiness of the raters by analyzing their feedback about SPs. We assume two different sets in the system: a) the set of service providers, $\mathbb{S}$ and b) the set of service consumers (hereafter referred as raters), $\mathbb{U}$. We note that these two sets are not necessarily disjoint. Transactions occur between SPs and raters, and raters provide feedbacks in the form of ratings about SPs after each transaction.

Let G_j be the reputation value of SP j ($j \in \mathbb{S}$) and T_{ij} be the rating that rater i ($i \in \mathbb{U}$) reports about SP j ($j \in \mathbb{S}$), whenever a transaction is completed between the two peers. Moreover, let R_i denote the trustworthiness of the peer i ($i \in \mathbb{U}$) as a rater. In other words, R_i represents the amount of confidence that the reputation system has about the correctness of any feedback/rating provided by rater i. All of these parameters may evolve with time. However, for simplicity, we omitted time dependencies from the notation. We assume there are u raters and s SPs in the system (i.e., $|\mathbb{U}| = u$ and $|\mathbb{S}| = s$). Let $\mathbb{G} = \{G_j : j \in \mathbb{S}\}$ and $\mathbb{R} = \{R_i : i \in \mathbb{U}\}$ be the collection of variables representing the reputations of the SPs and the trustworthiness values of the raters, respectively. Further, let $\mathbb{T}$ be the $s \times u$ SP-rater matrix that stores the rating values (T_{ij}), and $\mathbb{T}_i$ be the set of ratings provided by rater i. We consider slotted time throughout this discussion. At each time-slot (or epoch), the iterative reputation algorithm is executed using the input parameters $\mathbb{R}$ and $\mathbb{T}$ to obtain the reputation parameters (e.g., $\mathbb{G}$). After completing its iterations, the BP-ITRM scheme outputs new global reputations of the SPs as well as the trustworthiness ($\mathbb{R}$ values) of the raters. For simplicity of presentation, we assume that the rating values are from the set $\Upsilon = \{0, 1\}$. The extension in which rating values can take any real number can be developed similarly (we implemented the proposed scheme for both cases and illustrate its performance in Section 3.3).

The reputation management problem can be viewed as finding the marginal probability distributions of each variable in $\mathbb{G}$, given the observed data (i.e., evidence). There are s marginal probability functions, $p(G_j|\mathbb{T}, \mathbb{R})$, each of which is associated with a variable G_j; the reputation value of SP j. Loosely speaking, the present Bayesian approaches [27], [33] solve for these marginal distributions separately, leading to poor estimates as they neglect the interplay of the entire evidence. In contrast, we formulate the problem by considering the global function $p(\mathbb{G}|\mathbb{T}, \mathbb{R})$, which is the joint probability distribution function of the variables in $\mathbb{G}$ given the rating matrix and the trustworthiness values of the raters. Then, clearly, each marginal probability function $p(G_j|\mathbb{T}, \mathbb{R})$ may be obtained as follows:

$$p(G_j|\mathbb{T}, \mathbb{R}) = \sum_{\mathbb{G}\backslash\{G_j\}} p(\mathbb{G}|\mathbb{T}, \mathbb{R}), \tag{1}$$

where the notation $\mathbb{G}\backslash\{G_j\}$ implies all variables in $\mathbb{G}$ except G_j.

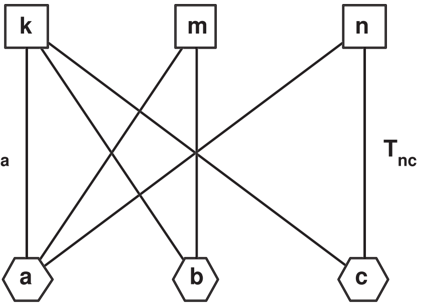

Fig. 1. Factor graph between the SPs and the raters in (3).

Unfortunately, the number of terms in (1) grows exponentially with the number of variables, making the computation infeasible for large-scale systems even for binary reputation values. However, we propose to factorize (1) to local functions f_i using a factor graph and utilize the belief propagation algorithm to calculate the marginal probability distributions in linear complexity. A factor graph is a bipartite graph containing two sets of nodes (corresponding to variables and factors) and edges incident between two sets. Following [10], we form a factor graph by setting a variable node for each variable G_j, a factor node for each function f_i, and an edge connecting variable node j to the factor node i if and only if G_j is an argument of f_i. We note that computing marginal probability functions is exact when the factor graph has no cycles. However, the belief propagation algorithm is still well defined and empirically often gives good approximate results for the factor graphs with cycles.

To describe the reputation system, we arrange the collection of the raters and the SPs together with their associated relations (i.e., the ratings of the SPs by the raters) as a bipartite (or factor) graph, as in Fig. 1. In this representation, each rater peer corresponds to a factor node in the graph, shown as a square. Each SP is represented by a variable node shown as a hexagon in the graph. Each report/rating is represented by an edge from the factor node to the variable node. Hence, if a rater i ($i \in \mathbb{U}$) has a report about SP j ($j \in \mathbb{S}$), we place an edge with value T_{ij} from the factor node i to the variable node representing SP j. We note that the T_{ij} value between rater i and SP j is the aggregation of all past and present ratings between these two peers as described in the following. If any new rating arrives from rater i about SP j, our scheme updates the value T_{ij} by averaging the new rating and the old value of the edge multiplied with the fading factor. The factor $\gamma_{ij}(t)$ is used to incorporate the fading factor of the SPs' reputation (service quality). We use a known factor $\gamma_{ij}(t) = \vartheta^{t-t_{ij}}$ where ϑ and t_{ij} are the fading parameter and the time when the last transaction between rater i and SP j occurred, respectively. The parameter ϑ is chosen to be less than one to give greater importance to more recent ratings.

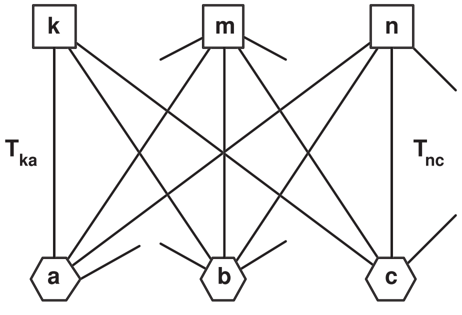

Fig. 2. Setup of the scheme.

Next, we suppose that the global function $p(\mathbb{G}|\mathbb{T},\mathbb{R})$ factors into products of several local functions, each having a subset of variables from $\mathbb{G}$ as arguments as follows:

$$p(\mathbb{G}|\mathbb{T},\mathbb{R}) = \frac{1}{Z}\prod_{i\in\mathbb{U}} f_i(\mathcal{G}_i, \mathbb{T}_i, R_i), \qquad (2)$$

where Z is the normalization constant and $\mathcal{G}_i$ is a subset of $\mathbb{G}$. Hence, in the graph representation of Fig. 1, each factor node is associated with a local function and each local function f_i represents the probability distributions of its arguments given the trustworthiness value and the existing ratings of the associated rater. As an example, the factor graph in Fig. 1 corresponds to

$$\begin{aligned} p(G_a, G_b, G_c|\mathbb{T},\mathbb{R}) = \frac{1}{Z} & f_k(G_a, G_b, G_c, T_{ka}, T_{kb}, T_{kc}, R_k) \\ & \times f_m(G_a, G_b, T_{ma}, T_{mb}, R_m) \\ & \times f_n(G_a, G_c, T_{na}, T_{nc}, R_n). \end{aligned} \qquad (3)$$

We note that using (3) in (1), one can attempt to compute the marginal distributions. However, as discussed before, this can get computationally infeasible. Instead, we utilize the belief propagation algorithm to calculate the marginal distributions of the variables in $\mathbb{G}$.

We now introduce the messages between the factor and the variable nodes to compute the marginal distributions using belief propagation. We note that all the messages are formed by the algorithm that is ran in the central authority. To that end, we choose an arbitrary factor graph as in Fig. 2 and describe message exchanges between rater k and SP a. We represent the set of neighbors of the variable node (SP) a and the factor node (rater) k as $\mathbf{N_a}$ and $\mathbf{N_k}$, respectively (neighbors of a SP are the set of raters who rated the SP while neighbors of a rater are the SPs whom it rated). Further, let $\Xi = \mathbf{N_a}\setminus\{k\}$ and $\Delta = \mathbf{N_k}\setminus\{a\}$. The belief propagation algorithm iteratively exchanges the probabilistic messages between the factor and the variable nodes in Fig. 2, updating the degree of beliefs on the reputation values of the SPs as well as the confidence of the raters on their ratings (i.e., trustworthiness values) at each step, until convergence. Let $\mathbb{G}^{(\nu)} = \{G_j^{(\nu)} : j \in \mathbb{S}\}$ be the collection of variables representing the values of the variable nodes at the iteration ν of the algorithm. We denote the messages from the variable nodes to the factor nodes and from the factor nodes to the variable nodes as μ and λ, respectively. The message $\mu_{a\to k}^{(\nu)}(G_a^{(\nu)})$ denotes the probability of $G_a^{(\nu)} = \ell$, $\ell \in \{0,1\}$, at the νth iteration. On the other hand, $\lambda_{k\to a}^{(\nu)}(G_a^{(\nu)})$ denotes the probability that $G_a^{(\nu)} = \ell$, for $\ell \in \{0,1\}$, at the νth iteration given T_{ka} and R_k.

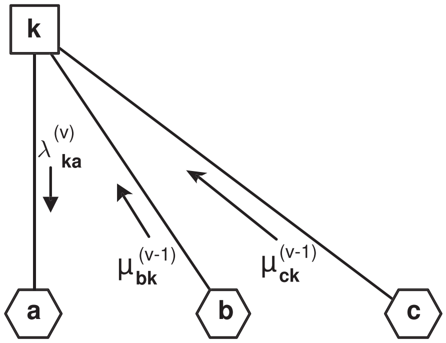

Fig. 3. Message from the factor node k to the variable node a at the νth iteration.

The message from the factor node k to the variable node a at the νth iteration is formed using the principles of the belief propagation as

$$\begin{aligned} &\lambda_{k\to a}^{(\nu)}(G_a^{(\nu)}) \\ &= \sum_{\mathbb{G}^{(\nu-1)}\setminus\{G_a^{(\nu-1)}\}} f_k(\mathcal{G}_k, \mathbb{T}_k, R_k^{(\nu-1)}) \prod_{x\in\Delta} \mu_{x\to k}^{(\nu-1)}(G_x^{(\nu-1)}), \end{aligned} \qquad (4)$$

where $\mathcal{G}_k$ is the set of variable nodes which are the arguments of the local function f_k at the factor node k. This message transfer is illustrated in Fig. 3. Further, $R_k^{(\nu-1)}$ (the trustworthiness of rater k calculated at the end of $(\nu-1)$th iteration) is a value between zero and one and can be calculated as follows:

$$R_k^{(\nu-1)} = 1 - \frac{1}{|N_k|}\sum_{i\in N_k}\sum_{x\in\{0,1\}} |T_{ki} - x|\mu_{i\to k}^{(\nu-1)}(x). \qquad (5)$$

The above equation can be interpreted as one minus the average inconsistency of rater k calculated by using the messages it received from all its neighbors. Using (4) and the fact that the reputation values in set $\mathbb{G}$ are independent from each other, it can be shown that $\lambda_{k\to a}^{(\nu)}(G_a^{(\nu)}) \propto p(G_a^{(\nu)}|T_{ka}, R_k^{(\nu-1)})$, where

$$\begin{aligned} &p\big(G_a^{(\nu)}|T_{ka}, R_k^{(\nu-1)}\big) = \\ &\Bigg[\Bigg(R_k^{(\nu-1)} + \frac{1-R_k^{(\nu-1)}}{2}\Bigg)T_{ka} + \frac{1-R_k^{(\nu-1)}}{2}(1-T_{ka})\Bigg]G_a^{(\nu)} + \\ &\Bigg[\frac{1-R_k^{(\nu-1)}}{2}T_{ka} + \Bigg(R_k^{(\nu-1)} + \frac{1-R_k^{(\nu-1)}}{2}\Bigg)(1-T_{ka})\Bigg] \\ &\big(1-G_a^{(\nu)}\big). \end{aligned} \qquad (6)$$

This resembles the belief/plausibility concept of the Dempster-Shafer Theory [39], [40]. Given $T_{ka} = 1$, $R_k^{(\nu-1)}$ can be viewed as the belief of the kth rater that $G_a^{(\nu)}$ is one (at the νth iteration). In other words, in the eyes of rater k, $G_a^{(\nu)}$ is equal to one with probability $R_k^{(\nu-1)}$. Thus, $(1 - R_k^{(\nu-1)})$ corresponds to the uncertainty in the belief of rater k. In order to remove this uncertainty and express $p(G_a^{(\nu)}|T_{ka}, R_k^{(\nu-1)})$ as the probabilities that $G_a^{(\nu)}$ is zero and one, we distribute the uncertainty uniformly between two outcomes (one and zero). Hence, in the eyes of the kth rater, $G_a^{(\nu)}$ is equal to one with probability $(R_k^{(\nu-1)} + (1 - R_k^{(\nu-1)})/2)$, and zero

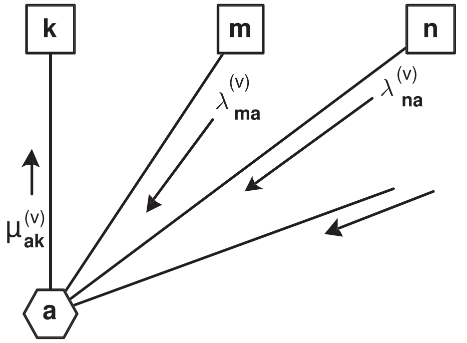

Fig. 4. Message from the variable node a to the factor node k at the νth iteration.

with probability $((1 - R_k^{(\nu-1)})/2)$. We note that a similar statement holds for the case when $T_{ka} = 0$. It is worth noting that, as opposed to the Dempster-Shafer Theory, we do not combine the beliefs of the raters. Instead, we consider the belief of each rater individually and calculate probabilities that $G_a^{(\nu)}$ being one and zero in the eyes of each rater as in (6). The above computation must be performed for every neighbors of each factor nodes. This finishes the first half of the νth iteration.

During the second half, the variable nodes generate their messages (μ) and send it to their neighbors. Variable node a forms $\mu_{a\to k}^{(\nu)}(G_a^{(\nu)})$ by multiplying all information it receives from its neighbors excluding the factor node k, as shown in Fig. 4. Hence, the message from variable node a to the factor node k at the νth iteration is given by

$$\mu_{a\to k}^{(\nu)}(G_a^{(\nu)}) = \frac{1}{\sum_{h\in\{0,1\}}\prod_{i\in\Xi}\lambda_{i\to a}^{(\nu)}(h)} \times \prod_{i\in\Xi}\lambda_{i\to a}^{(\nu)}(G_a^{(\nu)}). \qquad (7)$$

This computation is repeated for every neighbors of each variable node. The algorithm proceeds to the next iteration in the same way as the νth iteration. We note that the iterative algorithm starts its first iteration by computing $\lambda_{k\to a}^{(1)}(G_a^{(1)})$ in (4). However, instead of calculating in (5), the trustworthiness value R_k from the previous execution of BP-ITRM is used as initial values in (6).

The iterations stop when all variables in $\mathbb{G}$ converge. Therefore, at the end of each iteration, the reputations are calculated for each SP. To calculate the reputation value $G_a^{(\nu)}$, we first compute $\mu_a^{(\nu)}(G_a^{(\nu)})$ using (7) but replacing Ξ with $\mathbf{N_a}$, and then we set $G_a^{(\nu)} = \sum_{i=0}^{1} i\mu_a^{(\nu)}(i)$.

3 Security Evaluation of BP-ITRM

In this section, we mathematically model and analyze BP-ITRM. Moreover, we support the analysis via computer simulations and compare BP-ITRM with the existing and commonly used trust management schemes. In order to facilitate future references, frequently used notations are listed in Table 1.

3.1 Attack Models

We consider two major attacks that are common for any trust and reputation management mechanisms. Further, we assume that the attackers may collude and collaborate with each other:

TABLE 1
Notations and Definitions

$\mathbb{S}$	The set of service providers (SPs)
$\mathbb{U}_M$	The set of malicious raters
$\mathbb{U}_R$	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

- **Bad mouthing.** Malicious raters collude and attack the service providers with the highest reputation by giving low ratings in order to undermine them. It is also noted that in addition to the malicious peers, 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.

3.2 Analytic Evaluation

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 service provider remains unchanged during time slots. Moreover, the rating values are either 0 or 1 where 1 represents a good service quality. Ratings generated by the nonmalicious raters are distributed uniformly among the SPs (i.e., their ratings/edges in the graph representation are distributed uniformly among SPs). We further assumed that the rating value r_h (provided by the nonmalicious raters) is a random variable with Bernoulli distribution, where $Pr(r_h = \hat{G}_j) = p_c$ and $Pr(r_h \neq \hat{G}_j) = (1 - p_c)$, and $\hat{G}_j$ is the actual value of the global reputation of SP j. Even though we assumed binary values (0 or 1) for the actual reputation values of SPs, BP-ITRM also performs well and gives accurate results when the actual reputation values of the SPs are between 0 and 1. Indeed in Section 3.3, we implemented BP-ITRM when the rating values are from the set $\{1, \ldots, 5\}$ instead of binary values and illustrated the performance of the proposed scheme.[1] 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 before mounting their attacks at the $(T + 1)$th time-slot. Finally, we assumed that d is a random variable with Yule-Simon distribution, which resembles the power-law distribution used in modeling online systems [41], with the

[1] The performance of BP-ITRM in this nonbinary rating system (in which the rating values are from the set $\{1, \ldots, 5\}$) also illustrates its performance when the actual reputation values of the SPs are between 0 and 1 in the binary rating system. For example, a reputation value of 4 in the nonbinary rating system stands for a reputation value of 0.8 in the binary rating system.

probability mass function $f_d(d;\rho) = \rho B(d, \rho+1)$, where B is the Beta function. For modeling the adversary, we made the following assumptions. We assumed that the malicious raters initiate bad mouthing and collude while attacking the SPs (they attack the SPs who have the highest reputation values by rating them as $r_m = 0$). Further, the malicious raters attack the same set Γ of SPs at each time-slot. In other words, we denote by Γ the set of size b in which every victim SP has one edge from each of the malicious raters. We note that the results we provide in this section are based on the threat model described above. We wish to evaluate the performance for the time-slot $(T+1)$. It is worth noting that even though we discuss the details for bad-mouthing attack, similar counterpart results hold for ballot stuffing and combinations of bad mouthing and ballot stuffing as well.

ϵ-Optimal scheme. The performance of a reputation scheme is determined by its accuracy of estimating the global reputations of the SPs. We declare a reputation scheme to be ϵ-optimal if the mean absolute error (MAE) ($|G_j - \hat{G}_j|$) is less than or equal to ϵ for every SP. This introduces a class of optimal schemes.

Naturally, we need to answer the following question: for a fixed ϵ, what are the conditions to have an ϵ-optimal scheme? In order to answer this question we require two conditions to be satisfied: 1) the scheme should iteratively reduce the impact of malicious raters and decrease the error in the reputation values of the SPs until it converges, and 2) the error on the G_j value of each SP j should be less than or equal to ϵ once the scheme converges. In the following, we obtained the condition to arrive at the ϵ-optimal scheme. Although the discussions of the analysis are based on bad-mouthing attack, the system designed using these criteria will be robust against ballot stuffing and combinations of bad mouthing and ballot stuffing as well.

The bad-mouthing attack is aimed to reduce the global reputation values of the victim SPs. Hence, G_j value of a victim SP j should be a nondecreasing function of iterations. This leads to the first condition on the ϵ-optimal scheme.

Lemma 1 (Condition 1). *The error in the reputation values of the SPs decreases with each successive iterations (until convergence) if $G_a^{(2)} > G_a^{(1)}$ is satisfied with high probability for every SP a ($a \in \mathbb{S}$) with $\hat{G}_a = 1$.*[2]

Proof. Let $G_a^{(\omega)}$ and $G_a^{(\omega+1)}$ be the reputation value of an arbitrary SP a with $\hat{G}_a = 1$ calculated at the (ω)th and $(\omega+1)$th iterations, respectively. $G_a^{(\omega+1)} > G_a^{(\omega)}$ if the following is satisfied at the $(\omega+1)$th iteration:

$$\prod_{j \in \mathbb{W}_R \cap \mathbf{N_a}} \frac{2p_c R_j^{(w+1)} + 1 - R_j^{(w+1)}}{-2p_c R_j^{(w+1)} + 1 + R_j^{(w+1)}} \prod_{j \in \mathbb{W}_M \cap \mathbf{N_a}} \frac{1 - \hat{R}_j^{(w+1)}}{1 + \hat{R}_j^{(w+1)}} > \prod_{j \in \mathbb{W}_R \cap \mathbf{N_a}} \frac{2p_c R_j^{(w)} + 1 - R_j^{(w)}}{-2p_c R_j^{(w)} + 1 + R_j^{(w)}} \prod_{j \in \mathbb{W}_M \cap \mathbf{N_a}} \frac{1 - \hat{R}_j^{(w)}}{1 + \hat{R}_j^{(w)}}, \tag{8}$$

where $R_j^{(w)}$ and $\hat{R}_j^{(w)}$ are the trustworthiness values of a reliable and malicious rater calculated as in (5) at the wth iteration, respectively.

Given $G_a^{(\omega)} > G_a^{(\omega-1)}$ holds at the ωth iteration, we would get $\hat{R}_j^{(w)} > \hat{R}_j^{(w+1)}$ for $j \in \mathbb{W}_M \cap \mathbf{N_a}$ and $R_j^{(w+1)} \geq R_j^{(w)}$ for $j \in \mathbb{W}_R \cap \mathbf{N_a}$. Thus, (8) would hold for the $(w+1)$th iteration. On the other hand, if $G_a^{(\omega)} < G_a^{(\omega-1)}$, we get $\hat{R}_j^{(w)} < \hat{R}_j^{(w+1)}$ for $j \in \mathbb{W}_M \cap \mathbf{N_a}$ and $R_j^{(w+1)} < R_j^{(w)}$ for $j \in \mathbb{W}_R \cap \mathbf{N_a}$. Hence, (8) is not satisfied at the $(w+1)$th iteration. Therefore, if $G_a^{(\omega)} > G_a^{(\omega-1)}$ holds for some iteration ω, then the BP-ITRM algorithm reduces the error on the global reputation value (G_a) until the iterations stop, and hence, it is sufficient to satisfy $G_j^{(2)} > G_j^{(1)}$ with high probability for every SP j with $\hat{G}_j = 1$ (the set of SPs from which the victims are taken) to guarantee that BP-ITRM iteratively reduces the impact of malicious raters until it stops. □

As we described in Section 2, iterations of BP-ITRM stop when the G_j values converge for every SP j (i.e., do not change anymore). The following lemma shows that BP-ITRM converges to a unique solution given *Condition 1* is satisfied.

Lemma 2. *Given* Condition 1 *holds, G_j value of SP j converges to a unique solution ($\overline{G_j}$).*

Proof. From Lemma 1, BP-ITRM iteratively reduces the error in the reputation values of the SPs provided that *Condition* 1 is satisfied. Further, given *Condition* 1 is satisfied, the error in the reputation value of an arbitrary SP j stops decreasing at the νth iteration when $G_j^{(\nu)} = G_j^{(\nu+1)}$, where the value of ν depends on the fraction of malicious raters. Thus, given that BP-ITRM satisfies *Condition 1*, the reputation value of every SP converges to a unique value. □

Although because of the *Condition* 1, the error in the reputation values of the SPs decrease with successive iterations, it is unclear what would be the eventual impact of malicious raters. Hence, in the following, we derive the probability P for ϵ-optimality.

Lemma 3 (Condition 2). *Suppose that the* Condition 1 *is met. Let ν be the iteration at which the algorithm has converged. Then, BP-ITRM would be an ϵ-optimal scheme with probability P, where P is given in (9) as follows:*

$$P = \prod_{a \in \mathbb{S}} Pr\Bigg\{ \epsilon \geq 1 - \Bigg\{ \prod_{j \in \mathbb{W}_R \cap \mathbf{N_a}} \big(2p_c R_j^{(\nu+1)} + 1 - R_j^{(\nu+1)}\big) \prod_{j \in \mathbb{W}_M \cap \mathbf{N_a}} \big(1 - \hat{R}_j^{(\nu+1)}\big) \Bigg\} \Bigg/ \Bigg\{ \prod_{j \in \mathbb{W}_R \cap \mathbf{N_a}} \big(2p_c R_j^{(\nu+1)} + 1 - R_j^{(\nu+1)}\big) \prod_{j \in \mathbb{W}_M \cap \mathbf{N_a}} \big(1 - \hat{R}_j^{(\nu+1)}\big) + \prod_{j \in \mathbb{W}_R \cap \mathbf{N_a}} \big(-2p_c R_j^{(\nu+1)} + 1 + R_j^{(\nu+1)}\big) \prod_{j \in \mathbb{W}_M \cap \mathbf{N_a}} \big(1 + \hat{R}_j^{(\nu+1)}\big) \Bigg\} \Bigg\}. \tag{9}$$

Proof. Given *Condition* 1 is satisfied, G_a value of an arbitrary SP a (with $\hat{G}_a = 1$) increases with iterations. Let BP-ITRM converges at the νth iteration. Then, to have an ϵ-optimal scheme, G_a value calculated at the last iteration of

[2] The opposite must hold for any SP with $\hat{G}_a = 0$.

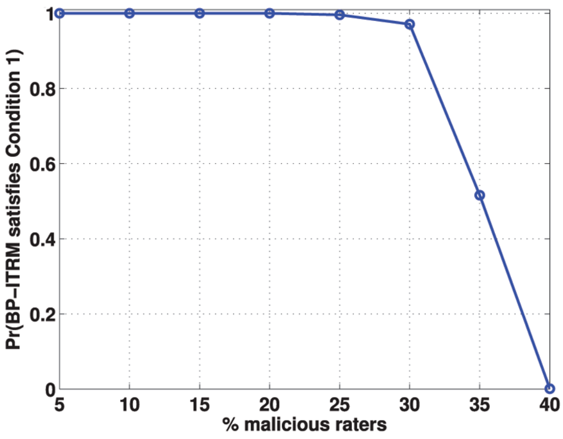

Fig. 5. Probability of BP-ITRM to satisfy *Condition 1* versus fraction of malicious raters.

BP-ITRM ($G_a^{(\nu)}$) should result in an error less than or equal to ϵ for every SP. That is, the following should hold for every SP

$$1 - G_a^{(\nu)} \leq \epsilon. \quad (10)$$

Further, if the scheme continues one more iteration after convergence, it can be shown that

$$G_a^{(\nu+1)} = G_a^{(\nu)}. \quad (11)$$

Thus, combining (10) and (11) leads to (9). □

We note that *Conditions* 1 and 2 in Lemmas 1 and 3 are to give an insight about the performance of the algorithm prior to the implementation. Hence, these conditions do not need to be checked at each execution of BP-ITRM in the real-life implementation of the algorithm.

Finally, the variation of the probability of BP-ITRM being an ϵ-optimal scheme over time is an important factor affecting the performance of the scheme. We observed that given BP-ITRM satisfies *Condition* 1 (that the error in the reputation values of the SPs monotonically decreases with iterations), the probability of BP-ITRM being an ϵ-optimal scheme increases with time. This criteria is given by the following lemma:

Lemma 4. *Let P_{T+1} and P_{T+2} be the probabilities that BP-ITRM is ϵ-optimal at the $(T+1)$th and $(T+2)$th time-slots, respectively. Then, given* Condition 1 *holds at the $(T+1)$th time-slot, we have $P_{T+2} > P_{T+1}$.*

Proof. Due to the fading factor, the contributions of the past reliable ratings of the malicious raters to their R_i values become less dominant with increasing time. Let $R_i(T)$ and $\hat{R}_i(T)$ be the trustworthiness of a reliable and malicious rater at the Tth time-slot, respectively. Then, given that *Condition* 1 is satisfied at the $(T+1)$th time-slot, it can be shown that $R_i(T+1) \geq R_i(T)$ and $\hat{R}_i(T+1) < \hat{R}_i(T)$. Thus, the probability that BP-ITRM satisfies *Condition* 1 increases at the $(T+2)$th time-slot. □

In the following example, we illustrate the results of our analytical evaluation. The parameters we used are $|\mathbb{U}_M| + |\mathbb{U}_R| = 100$, $|\mathbb{S}| = 100$, $\rho = 1$, $\vartheta = 0.9$, $T = 50$, $b = 5$ and $p_c = 0.8$. We note that there is no motive to select these parameters. We evaluated BP-ITRM with different parameters and obtained similar results. BP-ITRM works properly when the error in the reputation values of the SPs decreases monotonically with iterations until convergence. In other words, *Condition* 1 (in Lemma 1) is a fundamental requirement. In Fig. 5, we illustrated the probability of BP-ITRM to satisfy *Condition 1* versus fraction of malicious raters. We observed that BP-ITRM satisfies *Condition* 1 with a high probability for up to 30 percent malicious raters. Further, we observed a threshold phenomenon. That is, the probability of BP-ITRM to satisfy *Condition* 1 suddenly drops after exceeding a particular fraction of malicious raters. Next, in Fig. 6, we illustrated the probability of BP-ITRM being an ϵ-optimal scheme versus fraction of malicious raters for three different ϵ values. Again, we observed a threshold phenomenon. As the fraction of adversary exceeds a certain value, the probability of BP-ITRM being an ϵ-optimal scheme drops sharply. Moreover, Fig. 7 illustrates the average ϵ values (ϵ_{av}) for which BP-ITRM is an ϵ-optimal scheme with high probability for different fractions of malicious raters. We observed that BP-ITRM provides significantly small error values for up to 30 percent malicious raters. We note that these analytical results are also consistent with our simulation results that are illustrated in the next section.

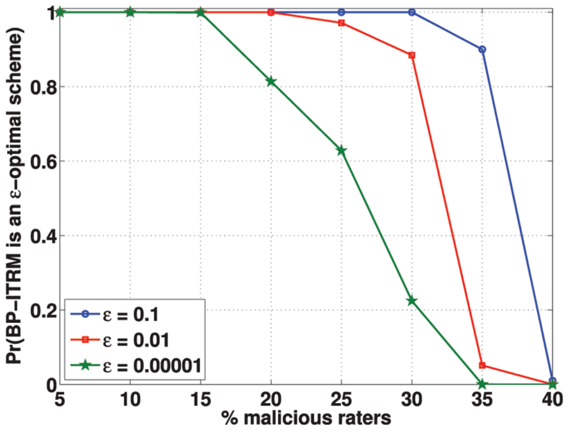

Fig. 6. Probability that BP-ITRM is an ϵ-optimal scheme versus fraction of malicious raters for different ϵ values.

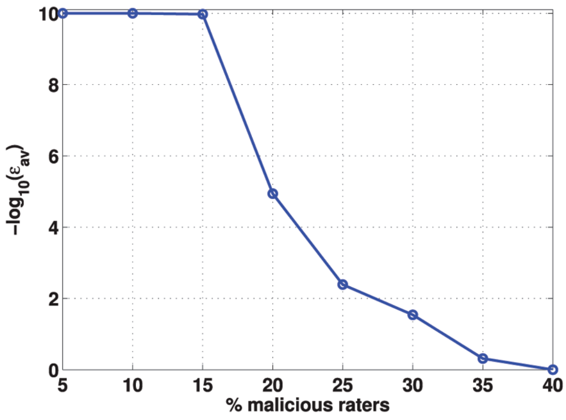

Fig. 7. The average ϵ values for which BP-ITRM is an ϵ-optimal scheme with high probability versus fraction of malicious raters.

3.3 Simulations

We evaluated the performance of BP-ITRM in the presence of bad mouthing, ballot stuffing, and combinations of bad mouthing and ballot stuffing. Here, we provide an evaluation of the bad-mouthing attack only, as similar results hold for ballot stuffing and combinations of bad mouthing and ballot stuffing. We compared the performance of BP-ITRM with three well known and commonly used reputation management schemes: 1) *The Averaging Scheme*, 2) *Bayesian Approach*, and 3) *Cluster Filtering*. The Averaging Scheme is widely used as in eBay or Amazon. The Bayesian Approach [27], [33] updates G_j using a Beta distribution. We implemented the Buchegger's Bayesian approach in [27] for the comparison with the deviation threshold $d = 0.5$ and trustworthiness threshold $t = 0.75$[3] (for details refer to [27]). It is worth noting that since we present evaluate BP-ITRM in a centralized setting, Buchegger's work in [27] and Whitby's work in [33] can be considered as similar. In [27], if a rater's rating deviates beyond the deviation threshold d from the calculated reputation value, its trustworthiness value is modified accordingly. Further, if a rater's trustworthiness exceeds a definite threshold t, it is detected as malicious. Similarly, in [33], instead of using the deviation threshold, the authors check if the calculated reputation value for the SP falls between a definite interval for each rater's rating distribution. As we will discuss later, both [27] and [33] have the same problem against colluding malicious raters. Cluster Filtering [29], [34], on the other hand, performs a dissimilarity test among the raters and then updates G_j using only the reliable raters. Finally, we compared BP-ITRM with our previous work on iterative trust and reputation management [8] (referred to as ITRM) to show the benefit of using belief propagation.

We assumed that d is a random variable with Yule-Simon distribution (with $\rho = 1$ throughout the simulations) as discussed in Section 3.2. Further, the fading parameter is set as $\vartheta = 0.9$[4] and number of ratings, per time-slot, by a malicious rater as $b = 5$. Let $\hat{G}_j$ be the actual value of the global reputation of SP j. Then, we obtained the performance of BP-ITRM, for each time-slot, as the mean absolute error (MAE) $|G_j - \hat{G}_j|$, averaged over all the SPs that are under attack.

We assumed that the malicious raters collude and attack the SPs who have the highest reputation values (assuming that the attackers knows the reputation values) and received the lowest number of ratings from the reliable raters (assuming that the attackers have this information). We note that this assumption may not hold in practice since the actual values of the global reputations and number of ratings received by each SP may not be available to malicious raters. However, we assumed that this information is available to the malicious raters to consider the worst case scenario. Further, the malicious raters collude and attack the same set Γ of SPs in each time-slot (which represents the strongest attack by the malicious raters). We further assumed that there are $|\mathbb{W}| = 100$ rater peers and $|\mathbb{S}| = 100$ SPs. Moreover, a total of $T = 50$ time-slots had passed since the lunch of the system, and reliable reports generated during those time-slots were distributed among the SPs uniformly. We note that we start our observations at time slot 1 after the initialization period.

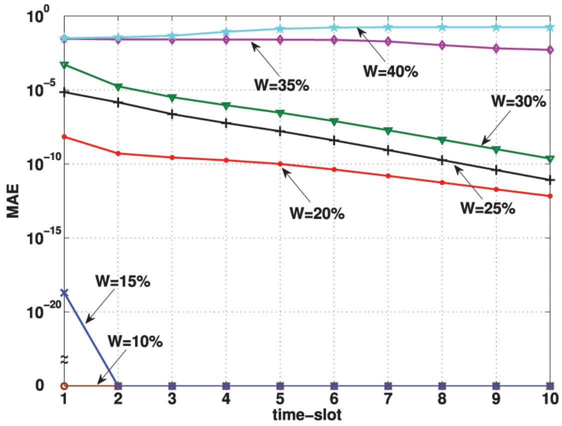

Fig. 8. MAE performance of BP-ITRM versus time when W of the existing raters become malicious in RepTrap [42].

Initially, we assumed that a fraction of the existing raters change behavior and become malicious after the start of the system (at time-slot one). The rating values are either 0 or 1. Using all their edges, the malicious raters collude and attack the SPs who have the highest reputation values and received the lowest number of ratings from the reliable raters, by rating them as $r_m = 0$. We note that this attack scenario also represents the RepTrap attack in [42] which is shown to be a strong attack. Since the ratings of the nonmalicious raters deviate from the actual reputation values via Bernoulli distribution, our attack scenario becomes even more severe than the RepTrap [42]. Further, we assumed that the rating r_h (provided by the nonmalicious raters) is a random variable with Bernoulli distribution, where $Pr(r_h = \hat{G}_j) = 0.8$ and $Pr(r_h \neq \hat{G}_j) = 0.2$. First, we evaluated the MAE performance of BP-ITRM for different fractions of malicious raters ($W = \frac{|\mathbb{W}_M|}{|\mathbb{W}_M|+|\mathbb{W}_R|}$), at different time-slots (measured since the attack is applied) in Fig. 8.[5] We observed that the proposed BP-ITRM provides significantly low errors for up to $W = 30\%$ malicious raters. Moreover, MAE at the first time slot is consistent with our analytical evaluation which was illustrated in Fig. 7. Next, we observed the change in the average trustworthiness (R_i values) of malicious raters with time. Figure 9 illustrates the drop in the trustworthiness of the malicious raters with time. We conclude that the R_i values of the malicious raters decrease over time, and hence, the impact of their malicious ratings is totally neutralized over time. We further observed the average number of required iterations of BP-ITRM at each time-slot in Fig. 10. We conclude that the average number of iterations for BP-ITRM decreases with time and decreasing fraction of malicious raters. Finally, we compared the MAE performance of BP-ITRM with the other schemes. Figure 11 illustrates the comparison of BP-ITRM with the other schemes for bad mouthing when the fraction

[3] We note that these are the same parameters used in the original paper [27].

[4] We note that for the Averaging Scheme, Bayesian Approach, and Cluster Filtering we used the same fading mechanism as BP-ITRM (discussed in Section 2) and set the fading parameter as $\vartheta = 0.9$.

[5] The plots in Figs. 8, 9, 10, 11, 12, and 13 are shown from the time-slot the adversary introduced its attack.

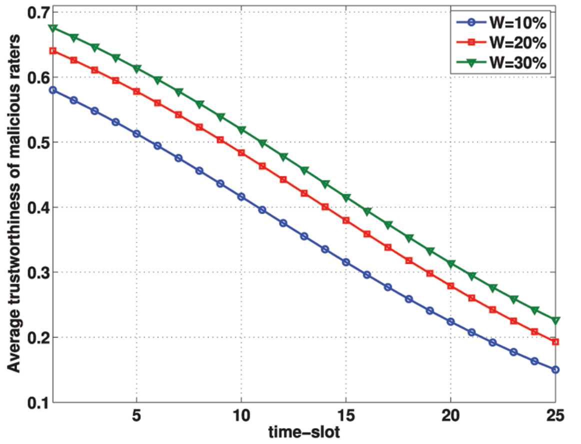

Fig. 9. Change in average trustworthiness of malicious raters versus time for BP-ITRM when W of the existing raters become malicious in RepTrap [42].

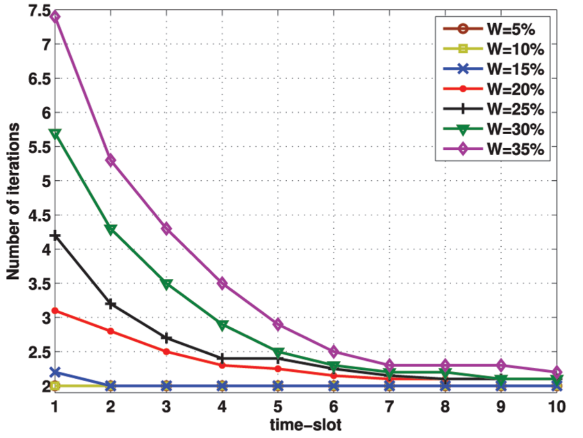

Fig. 10. The average number of iterations versus time for BP-ITRM when W of the existing raters become malicious in RepTrap [42].

of malicious raters (W) is 30 percent. It is clear that BP-ITRM outperforms all the other techniques significantly.

Next, we simulated the same attack scenario when ratings are integers from the set $\{1, \ldots, 5\}$ instead of binary values. We assumed that the rating r_h is a random variable with folded normal distribution (mean $\hat{G}_j$ and variance 0.5), however, it takes only discrete values from 1 to 5. Malicious raters choose SPs from Γ and rate them as $r_m = 4$. The malicious raters do not deviate very much from the actual $\hat{G}_j = 5$ values to remain undercover (while still attacking) as many time-slots as possible. We also tried higher deviations from the $\hat{G}_j$ value and observed that the malicious raters were easily detected by BP-ITRM. Figure 12 illustrates that BP-ITRM provides significantly low MAE for up to $W = 40\%$ malicious raters. We then compared the MAE performance of BP-ITRM with the other schemes in Fig. 13 and observed that BP-ITRM outperforms all the other techniques significantly.

In most trust and reputation management systems, the adversary causes the most serious damage by introducing newcomer raters to the system. Since it is not possible for the system to know the trustworthiness of the newcomer raters, the adversary may introduce newcomer raters to the systems and attack the SPs using those raters. To study the effect of newcomer malicious raters to the reputation management scheme, we introduced 100 more raters as newcomers. Hence, we had $|\mathbb{U}_M| + |\mathbb{U}_R| = 200$ raters and $|\mathbb{S}| = 100$ SPs in total. We assumed that the rating values are either 0 or 1, r_h is a random variable with Bernoulli distribution as before, and malicious raters choose SPs from Γ and rate them as $r_m = 0$ (this particular attack scenario

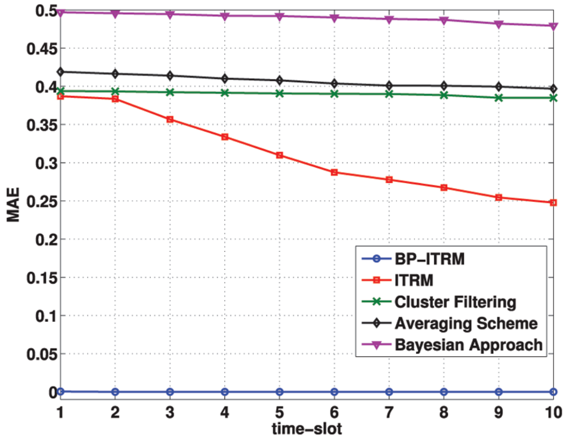

Fig. 11. MAE performance of various schemes when 30 percent of the existing raters become malicious in RepTrap [42].

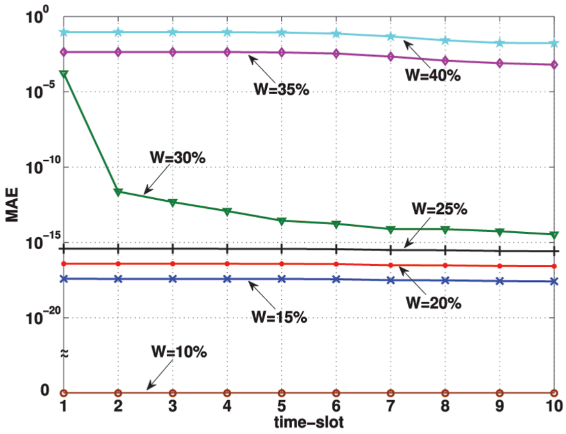

Fig. 12. MAE performance of BP-ITRM versus time when W of the existing raters become malicious and rating values are integers from $\{1, \ldots, 5\}$ in RepTrap [42].

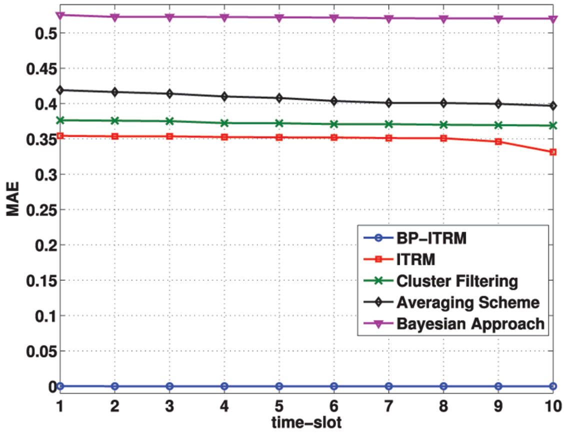

Fig. 13. MAE performance of various schemes when 30 percent of the existing raters become malicious and rating values are from $\{1, \ldots, 5\}$ in RepTrap [42].

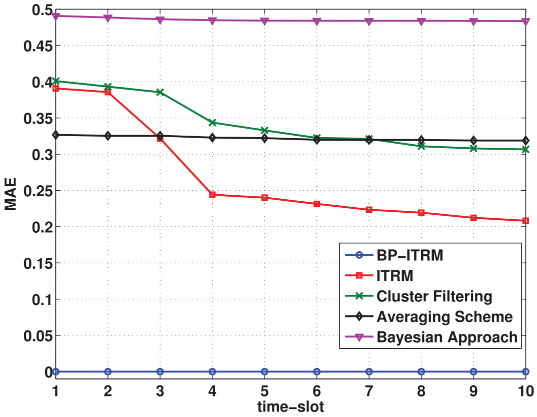

Fig. 14. MAE performance of various schemes when 30 percent of the newcomer raters are malicious.

does not represent the RepTrap attack). We compared the MAE performance of BP-ITRM with the other schemes for this scenario in Fig. 14.[6]

From these simulation results, we conclude that BP-ITRM significantly outperforms the Averaging Scheme, Bayesian Approach, and Cluster Filtering in the presence of attackers. We identify that the Bayesian Approach performs the worst against the RepTrap attack and colluding attacks from malicious raters. Indeed, both [27] and [33] have the same shortcoming against colluding malicious raters. Both [27] and [33] first calculate the reputation value of a particular SP, and then based on the calculated value, they adjust each rater's trustworthiness value. On the other hand, when the malicious raters collude (as in our attack scenario), it is likely that the majority of the ratings to the victim SPs will be from malicious raters. In this scenario, the Bayesian approach not only fails to filter the malicious ratings but it also punishes the reliable raters which rates the victim SPs. We also identify that ITRM (i.e., our algebraic iterative scheme) is the closest in accuracy to BP-ITRM. This emphasizes the robustness of using iterative message passing algorithms for reputation management.

3.4 Computational Complexity

In this section, we provide some discussion on the computational complexity. It can be argued that the computational complexity of BP-ITRM is quadratic with the number of raters (or SPs) due to the use of the probability-domain message passing algorithm. This is because of multiplications of probabilities in (7) and (4). However, this quadratic computational complexity can be further reduced by using similar techniques developed for message passing decoding of LDPC codes (using belief propagation) for lower complexity. We used a log-domain algorithm in our implementation, which is often used for LDPC codes [43] to reduce the complexity. Specifically, assuming $|\mathbb{U}| = u$ raters and $|\mathbb{S}| = s$ SPs in the system, we obtained the computational complexity of BP-ITRM as $\mathbf{max}(\mathrm{O}(cu), \mathrm{O}(cs))$ in the number of multiplications, where c is a small constant number representing the average number of ratings (reports) per rater. On the other hand, Cluster Filtering suffers quadratic complexity versus the number of raters (or SPs).

[6] The plot is shown from the time-slot the newcomers are introduced.

4 Conclusion

In this paper, we introduced the Belief Propagation-based Iterative Trust and Reputation Management Scheme (BP-ITRM). Our work is an iterative probabilistic algorithm motivated by the prior success of message passing techniques and belief propagation algorithms on decoding of turbo codes and low-density parity-check codes. BP-ITRM relies on a graph-based representation of an appropriately chosen factor graph for reputation systems. In this representation, service providers and raters are arranged as two sets of variable and factor nodes that are connected via some edges. The reputation values of SPs are computed by message passing between nodes in the graph until the convergence. The proposed BP-ITRM is a robust mechanism to evaluate the quality of the service of the SPs from the ratings received from the recipients of the service (raters). Moreover, it effectively evaluates the trustworthiness of the raters. We studied BP-ITRM by a detailed analysis and showed the robustness using computer simulations. We proved that BP-ITRM iteratively reduces the error in the reputation values of SPs due to the malicious raters with a high probability. Further, we observed that this probability demonstrates a threshold property. That is, exceeding a particular fraction of malicious raters reduces the probability sharply. We also compared BP-ITRM with some well-known reputation management schemes and showed the superiority of our scheme both in terms of robustness and efficiency.

Acknowledgments

This material is based upon work supported partially by the US National Science Foundation under Grant No. IIS-1115199, and a gift from the Cisco University Research Program Fund, an advised fund of Silicon Valley Community Foundation.

References

[1] S. Buchegger and J. Boudec, "Performance Analysis of Confidant Protocol (Coorperation of Nodes: Fairness in Dynamic Ad-Hoc Networks)," *Proc. IEEE/ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHOC),* June 2002.

[2] S. Buchegger and J. Boudec, "A Robust Reputation System for P2P and Mobile Ad-Hoc Networks," *Proc. Second Workshop the Economics of Peer-to-Peer Systems,* 2004.

[3] S. Ganeriwal and M. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *Proc. Second ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '04),* pp. 66-77, 2004.

[4] Y. Sun, W. Yu, Z. Han, and K. Liu, "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 2, pp. 305-317, Feb. 2006.

[5] H. Pishro-Nik and F. Fekri, "On Decoding of Low-Density Parity-Check Codes on the Binary Erasure Channel," *IEEE Trans. Information Theory,* vol. 50, no. 3, pp. 439-454, Mar. 2004.

[6] H. Pishro-Nik and F. Fekri, "Results on Punctured Low-density Parity-Check Codes and Improved Iterative Decoding Techniques," *IEEE Trans. Information Theory,* vol. 53, no. 2, pp. 599-614, Feb. 2007.

[7] B.N. Vellambi and F. Fekri, "Results on the Improved Decoding Algorithm for Low-Density Parity-Check Codes over the Binary Erasure Channel," *IEEE Trans. Information Theory,* vol. 53, no. 4, pp. 1510-1520, Apr. 2007.

[8] E. Ayday, H. Lee, and F. Fekri, "An Iterative Algorithm for Trust and Reputation Management," *Proc. IEEE Int'l Symp. Information Theory (ISIT '09),* 2009.

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

[10] F. Kschischang, B. Frey, and H.A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Information Theory,* vol. 47, no. 2, pp. 498-519, Feb. 2001.

[11] J. Zhang and M. Fossorier, "Shuffled Belief Propagation Decoding," *Proc. 36th Asilomar Conf. Signals, Systems and Computers,* Nov. 2002.

[12] E. Ayday, H. Lee, and F. Fekri, "Trust Management and Adversary Detection in Delay Tolerant Networks," *Proc. IEEE Military Comm. Conf. (MILCOM '10),* 2010.

[13] Y. Liu, A.H. Ngu, and L.Z. Zeng, "Qos Computation and Policing in Dynamic Web Service Selection," *Proc. 13th Int'l World Wide Web Conf. Alternate Track Papers & Posters (WWW Alt. '04),* pp. 66-73, 2004.

[14] U.S. Manikrao and T.V. Prabhakar, "Dynamic Selection of Web Services with Recommendation System," *Proc. Int'l Conf. Next Generation Web Services Practices (NWESP '05),* p. 117, 2005.

[15] E.M. Maximilien and M.P. Singh, "Conceptual Model of Web Service Reputation," *SIGMOD Record,* vol. 31, no. 4, pp. 36-41, 2002.

[16] E.M. Maximilien and M.P. Singh, "Toward Autonomic Web Services Trust and Selection," *Proc. Second Int'l Conf. Service Oriented Computing (ICSOC '04),* pp. 212-221, 2004.

[17] E.M. Maximilien and M.P. Singh, "Multiagent System for Dynamic Web Services Selection," *Proc. First Workshop Service-Oriented Computing and Agent-Based Eng.,* 2005.

[18] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," *Proc. Tenth Int'l Conf. Information and Knowledge Management (CIKM '01),* pp. 310-317, 2001.

[19] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," *Proc. 11th Int'l Conf. World Wide Web (WWW '02),* pp. 376-386, 2002.

[20] 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," *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS '02),* pp. 207-216, 2002.

[21] D. Fahrenholtz and W. Lamersdorf, "Transactional Security for a Distributed Reputation Management System," *Proc. Third Int'l Conf. E-Commerce and Web Technologies (EC-WEB '02),* pp. 214-223, 2002.

[22] M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," *Proc. 13th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '03),* pp. 144-152, 2003.

[23] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks," *Proc. 12th Int'l Conf. World Wide Web (WWW '03),* pp. 640-651, 2003.

[24] C.-W. Hang, Y. Wang, and M.P. Singh, "An Adaptive Probabilistic Trust Model and Its Evaluation," *Proc. Seventh Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '08),* vol. 3, pp. 1485-1488, 2008.

[25] Y. Wang and M.P. Singh, "Evidence-Based Trust: A Mathematical Model Geared for Multiagent Systems," *ACM Trans. Autonomous and Adaptive Systems,* vol. 5, pp. 14:1-14:28, Nov. 2010.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank Citation Ranking: Bringing Order to the Web," technical report, Stanford Digital Library Technologies Project, 1998.

[27] S. Buchegger and J. Boudec, "Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad Hoc Networks," Technical Report IC/2003/31, EPFL-DI-ICA, 2003.

[28] G. Zacharia, A. Moukas, and P. Maes, "Collaborative Reputation Mechanisms in Electronic Marketplaces," *Proc. 32nd Ann. Hawaii Int'l Conf. System Sciences (HICSS '99),* 1999.

[29] C. Dellarocas, "Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior," *Proc. Second ACM Conf. Electronic Commerce (EC '00),* pp. 150-157, 2000.

[30] P. Resnick and R. Zeckhauser, "Trust among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System," *Proc. Workshop Empirical Studies of Electronic Commerce,* 2002.

[31] J.M. Pujol, R. Sangüesa, and J. Delgado, "Extracting Reputation in Multi Agent Systems by Means of Social Network Topology," *Proc. First Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '02),* pp. 467-474, 2002.

[32] P. Yolum and P. Singh, "Self-Organizing Referral Networks: A Process View of Trust and Authority," *First Int'l Workshop Eng. Self-Organising Applications (ESOA '03),* July 2003.

[33] A. Whitby, A. Josang, and J. Indulska, "Filtering Out Unfair Ratings in Bayesian Reputation Systems," *Proc. Seventh Int'l Workshop Trust in Agent Societies (AAMAS '04),* 2004.

[34] P. Macnaughton-Smith, W.T. Williams, M.B. Dale, and L.G. Mockett, "Dissimilarity Analysis: A New Technique of Hierarchical Sub-Division," *Nature,* vol. 202, pp. 1034-1035, 1964.

[35] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Comm. ACM,* vol. 35, pp. 61-70, Dec. 1992.

[36] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW '94),* pp. 175-186, 1994.

[37] J. Herlocker, J.A. Konstan, and J. Riedl, "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms," *Information Retrieval,* vol. 5, no. 4, pp. 287-310, 2002.

[38] B.M. Sarwar, G. Karypis, J.A. Konstan, and J.T. Riedl, "Application of Dimensionality Reduction in Recommender System—A Case Study," *Proc. ACM WebKDD Web Mining ECommerce Workshop,* 2000.

[39] G. Shafer, *A Mathematical Theory of Evidence.* Princeton Univ. Press, 1976.

[40] G. Shafer, "The Dempster-Shafer Theory," *Encyclopedia of Artificial Intelligence,* 1992.

[41] F. Slanina and Y.C. Zhang, "Referee Networks and Their Spectral Properties," *Acta Physica Polonica B,* vol. 36, p. 2797, Sep. 2005.

[42] Y. Yang, Q. Feng, Y.L. Sun, and Y. Dai, "RepTrap: a Novel Attack on Feedback-Based Reputation Systems," *Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (Secure Comm '08),* pp. 1-11, 2008.

[43] J. Chen, A. Dholakia, E. Eleflhetiou, M. Fossotier, and X.-Y. Hu, "Near Optimum Reduced-Complexity Decoding Algonhm for LDPC Codes," *Proc. IEEE Int'l Symp. Information Theory,* July 2002.

Erman Ayday received the BS degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2005. He received the MS and PhD degrees from the School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta, Georgia, in 2007 and 2011, respectively. His current research interests include wireless network security, game theory for wireless networks, trust and reputation management, and recommender systems. He is the recipient of 2010 Outstanding Research Award from the Center of Signal and Image Processing (CSIP) at Georgia Tech and 2011 ECE Graduate Research Assistant (GRA) Excellence Award from Georgia Tech. He is a student member of the IEEE.

Faramarz Fekri received the PhD degree from the Georgia Institute of Technology in 2000. Since 2000, he has been with the faculty of the School of Electrical and Computer Engineering at the Georgia Institute of Technology where he currently holds a full professor position. He serves on the editorial board of the *IEEE Transactions on Communications*, and on the Technical Program Committees of several IEEE conferences. His current research interests include the area of communications and signal processing, in particular coding and information theory, information processing for wireless and sensor networks, and communication security. He received the US National Science Foundation CAREER Award (2001), and Southern Center for Electrical Engineering Education (SCEEE) Research Initiation Award (2003), Outstanding Young faculty Award of the School of ECE (2006). He is a senior member of the IEEE.