Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 637831, 11 pages
http://dx.doi.org/10.1155/2015/637831

Research Article

AuthenticPeer: A Reputation Management System for Peer-to-Peer Wireless Sensor Networks

Heba Kurdi,[1] Sarah Alnasser,[2] and Munira Alhelal[2]

[1] *Computer Science Department, King Saud University, P.O. Box 2454, Riyadh 11451, Saudi Arabia*
[2] *Computer Science Department, Al Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia*

Correspondence should be addressed to Heba Kurdi; heba.kurdi@gmail.com

Received 27 April 2015; Revised 10 July 2015; Accepted 12 July 2015

Academic Editor: Qun Jin

Reputation and trust are dominant challenges in peer-to-peer (P2P) networks in general and peer-to-peer sensor network (P2PWSN) in particular. Numerous efforts have been devoted to developing reputation management systems that can alleviate the difficulties in finding trustworthy communication partners in P2P networks. There are two main approaches in this field: peer-based reputation systems and message-based reputation systems, each of which has its own advantages and limitations. In this paper, we propose a new reputation system for P2PWSN, AuthenticPeer, which utilizes techniques from both approaches in a way that maximizes their advantages and overcomes their limitations. The proposed system has been tested thoroughly in various simulated P2PWSN environments under various number of files and common threat models. Experimental results illustrate enhanced performance of AuthenticPeer reputation system when compared to two P2P reputation systems, EigenTrust and Incremental EigenTrust, in terms of success rate of good users and fraction of inauthentic downloads, specifically with the threat models: individual malicious, malicious collective, camouflaged collective, and malicious spies, regardless of the number of files in the networks.

1. Introduction

Peer-to-peer sensor network (P2PWSN) is an emerging type of both wireless sensor networks and peer-to-peer (P2P) networks. It takes a service-oriented, data-centric view of the deployed wireless sensor networks (WSN) [1]. On one side, P2P networks are appropriate for high-end communication nodes; they are well suited for flexible file sharing, content delivery, and distributed computing [2]. For instance, Gnutella [3] was among astonishingly successful P2P file sharing paradigms over the Internet. Additionally, AN.P2P [4], which enables peers to deliver original content objects with associated workflows to other peers, is considered to be one of the most important content delivery networks (CDN). On the other side, WSN are well aligned with collecting environmental or ambient information in extreme conditions [5].

This integration between P2P networks would offer the important advantages of P2P networks to WSN, which include offering an open system where peers can freely and easily join the systems donating their resources [6]. Thus, as peers arrive and demands on the system increase, the total capacity of the system would also increase. On the other hand, the open nature of P2P networks and the absence of a central control authority have led to abuses of these networks by malicious peers who distribute in authentic messages and viruses. Therefore, finding a trustworthy peer is considered to be the grand challenge facing P2P networks [7]. Many reputation management systems have been introduced to address this challenge by building trust information among the peers, depending on feedback about past transactions between them [8]. Additionally, there is already a plethora of research in the area of reputation systems for WSN [9–12]. However, to the best of our knowledge, no reputation system has been proposed for P2PWSN in particular. Therefore this paper sets out to bridge this gap by proposing the AuthenticPeer reputation system, which evaluates and judges the message authenticity and selects the most reputable trusted peer as a source for the message download.

Basically, there are two main categories of reputation management systems: peer-based reputation management systems and message-based reputation management systems. In the former approach, peers are judged based on their

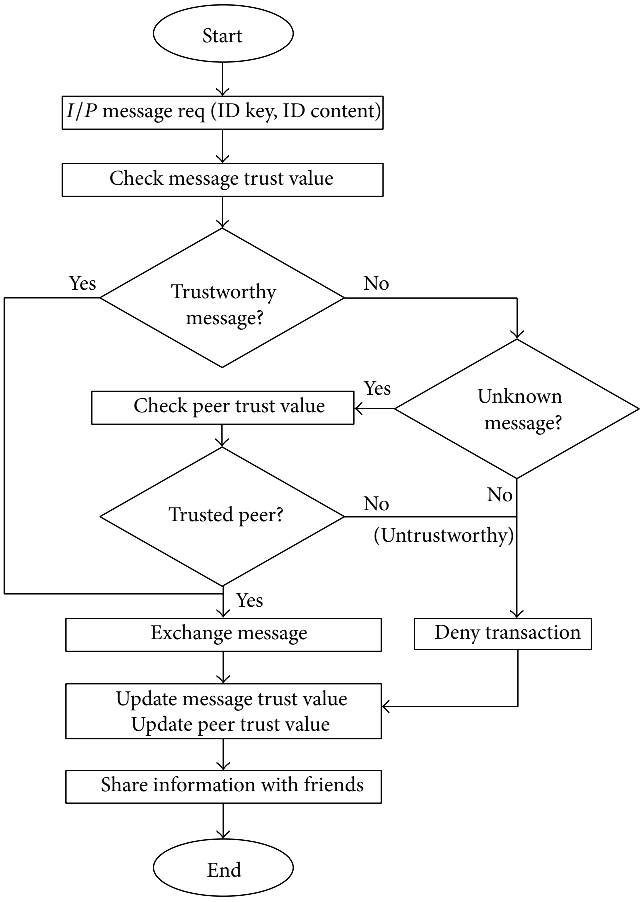


Figure 1: AuthenticPeer algorithm.

behavior with other peers; they gain a good or bad global trust value according to the quality of transactions in which they participated. The selection of a download source is based on the global trust value. In this way, malicious peers, who most likely have low global trust values due to their evil acts, can be isolated. Nevertheless, this approach suffers from some limitations: (1) malicious peers can change their identity if the system catches them, so their global trust values are erased and they are considered as newcomers rather than malicious peers. (2) If a malicious peer is mistakenly chosen as a download source by a good peer, the former can spread inauthentic messages over the system through the latter, who does not know the harms associated with these files. In AuthenticPeer, we overcome these limitations by checking the message trustworthiness before checking the peer trustworthiness. It is only for unknown messages when the peer is checked, as illustrated in Figure 1. This strategy restricts malicious peers from distributing bad messages, even when they change their identity or when mistakenly selected as download sources by good peers.

On the other hand, in message-based reputation systems, the selection of a download source is based on the quality of the file instead of the quality of the peer. In this approach, certain parameters are calculated for each file through several transactions before it can be identified as a trustworthy or untrustworthy file. The main drawbacks of this approach are as follows: (1) It takes time until the quality of a file is recognized. (2) Calculations of required parameters consumes system resources. In AuthenticPeer approach, we overcome the first limitation by calculating the trust and reputation values based on peer quality rather than the message quality until the quality of the file is revealed, as illustrated in Figure 1. To overcome the limitation related to the computation resources, message-based approach is only used when the message trust value is known, based on previous transactions. Additionally, this value is accumulated after each transaction in which the file is involved, rather than computing it all at once when needed.

The remaining parts of this paper are organized as follows: Section 2 reviews related works on reputation management systems. The hybrid approach to reputation management systems, AuthenticPeer, is then introduced illustrating its architecture in Section 3 and algorithm in Section 4. Section 5 describes the comparative study followed to evaluate the proposed system. Experimental results are presented and discussed in Section 6. Finally, Section 7 concludes the paper with a brief summary and directions for future works.

2. Related Work

Since the early days of P2P networks, many reputation management systems have emerged following two main approaches: peer-based reputation systems, where the source of download is decided based on peers' honesty, and message-based reputation systems, where the quality of provided files is the decision factor for download. In this section, we provide a brief review of eminent related works on each approach.

2.1. Peer-Based Reputation Systems. In [13], the PowerTrust reputation system is developed. It dynamically selects a small number of nodes that are most reputable, using a distributed ranking mechanism. These nodes are called power nodes and play an important role in calculating the global and local reputation values of other peers. By using a look-ahead random walk strategy and leveraging the power nodes, PowerTrust is known to improve the global reputation accuracy and the trust values aggregation speed.

The PeerTrust reputation management system, proposed in [14], computes the peer reputation as the average trust values weighted by the score of peers providing these values, the number of transactions in which they were involved, and the credibility of their feedback, among other factors. The main drawback of this approach is the heavy overhead associated with retrieving weighting factors.

A trust management protocol is proposed in [15], specifically for Delay Tolerant Networks (DTNs), which are characterized by high end-to-end latency, frequent disconnection, and opportunistic communication. This protocol applies a secure routing policy that combines Quality of Service (QoS) trust with social trust to obtain a composite trust metric. This allows the best trust setting for trust aggregation to be identified so that bias is minimized.

A peer-based reputation algorithm, EigenTrust, is proposed in [16]. In this algorithm, each peer has a global trust value that is calculated by normalizing an aggregated value of local trust assessments, considering the transitive trust for peers with limited past experience. The algorithm introduces the idea of pretrusted peers who are trusted by everyone. This idea is useful mainly for newcomers who have no previous interactions with any peer. Since normalization scales all trust values to the range from zero to one, the EigenTrust Algorithm can be considered as a fast converging algorithm with low storage overhead for each peer. EigenTrust is known of its ability to decrease the impact of malicious peers in file sharing networks [17, 18]. Despite the vast advantages offered by EigenTrust Algorithm, it has some limitations. Mainly, if a trustworthy peer downloads an inAuthenticPeer from a malicious peer, this would allow the file to be easily accepted by other peers, leading to a chain of inauthentic downloads. A slightly enhanced version of EigenTrust has been proposed in [13] with the title Incremental EigenTrust Algorithm. It suggests using snapshot comparisons to avoid costly recalculation on each trust and reputation calculation cycle.

Another modification to the EigenTrust Algorithm is proposed in [19], under the name of Dual-EigenRep. In this algorithm, two distinct trust values are identified for each peer: recommended reputation value (RDRV) and recommending reputation value (RGRV). The former aggregates the trust ratings and RGRV of other peers who downloaded files from the peer, while the latter aggregates the self-trust ratings and RDRV of other peers from which the peer has downloaded files. The entire network can form different trust communities based on the correlation between RDRV and RGRV. This system shows improved results over EigenTrust when dealing with certain malicious behaviors, namely, exaggeration, collusion, disguise, and single dimension.

In [20], the reputation-based management algorithm is introduced. This algorithm maintains trust between peers of online communities using feedback provided by each peer, depending on responses to a trust query. A threshold is used to determine the number of responses to be evaluated for each trust query. The global trust value is calculated by taking the average of the evaluated trust ratings, weighted by the reputation of their senders. The main advantages of this algorithm include addressing the denial-of-service attack and having a robust security plan. Yet, with this protocol, an honest peer may share a harmful resource that it downloads without being aware of its malicious content. This problem is common problem encountered in [13, 16, 19] as well.

2.2. Message-Based Reputation Systems. The amount of work available on message-based reputation systems is less than that on peer-based reputation systems. Reviewing has revealed only the two systems proposed in [21, 22]. In [21], the reputation management system in a structured P2P network is proposed to prevent inauthentic message distribution and download. Seeing that malicious peers may change their identities if the system catches them, this paper suggests depending on message-based reputation information instead. This is because such information is more difficult to change. The proposed approach relies on a Distributed Hash Table (DHT) P2P network, where each file has an ID key based on its name and content. Each peer has a file repository and a peer repository to store reputation values. Evaluating files and giving reputation values can only be done by trustworthy peers, known as file reputation managers. If a peer needs a file, it sends a query with the file ID to the corresponding file reputation manager, which returns all files with the same ID. The file can be considered a trustworthy, untrustworthy, or unknown file, based on certain system parameters and formulas that are calculated for each file after each transaction.

In [22], a message-based reputation approach, called Fighting P2P Spam and Decoys with Object Reputation, is presented. It describes a simple weighted voting protocol in which any peer can evaluate a file positively or negatively. The votes are then collected and aggregated for final evaluation. To prevent malicious peers from having equal chances to evaluate files as good peers do, the system introduces a new technique based on a local correlation graph. However, a common drawback among message-based reputation systems is the extensive computation required for calculating reputation values, which exhausts system resources.

Since this paper intended to use the EigenTrust Algorithm as a baseline of its implementation, a comparison between peer-based reputation management systems built on EigenTrust and message-based reputation systems is summarized in Table 1, for quick review.

3. System Architecture

AuthenticPeer utilizes techniques from both peer-based reputation approach and message-based reputation approach with an aim of maximizing their advantages and overcoming their limitations. The architecture of AuthenticPeer is composed of three components: user interface, file reputation manager, and peer reputation manager.

3.1. User Interface. This is where the user enters their file request. Each file in the system has two identifications:

(i) ID key to identify the file name.

(ii) ID content to identify the file content.

It is important to note that any two files with the same ID key and different ID content will be treated differently.

3.2. Message Reputation Manager. This component is responsible for the following:

(i) Evaluating the message reputation before any transaction.

(ii) Updating the message trust value after each transaction.

Following the same approach of [21], the message reputation is judged based on the following condition:

$$\left|\text{positive}\right| + \left|\text{negative}\right| > T, \quad (1)$$

Table 1: Comparison between related reputation management systems.

(a) Peer-based reputation systems

System	EigenTrust [16] Incremental EigenTrust [13]	Dual-EigenRep [19]	Reputation-based trust management system [20]
Advantage	(i) Fast convergence (ii) Low storage overhead	Handles exaggeration, collusion, disguise, and single behavior	(i) Addresses denial-of-service attack (ii) Provides robust security
Disadvantage	(i) Needs pretrusted peers (ii) Trustworthy peer may participate in distributing harmful files	(i) Heavy computation overheads (ii) Trustworthy peer may participate in distributing harmful files	(i) Effective only against limited attack models (ii) Trustworthy peer may participate in distributing harmful files

(b) Message-based reputation systems

System	Reputation management system [21]	Fighting P2P Spam and Decoys [22]
Advantage	Only trustworthy peers can evaluate files and give reputation values	Minimize the feedback provided by malicious peers using the correlation graph
Disadvantage	(i) File must go through several transactions before they are identified as trustworthy or untrustworthy (ii) Heavy computation overheads	(i) Overheads associated with the correlation graph (ii) Heavy computation overheads

where

(i) positive is the number of evaluations that consider the message trustworthy,

(ii) negative is the number of evaluations that consider the message untrustworthy,

(iii) T is the threshold of the minimum number of message evaluations.

The message trust value is judged based on the following condition:

$$\frac{\text{positive}}{(|\text{positive}| + |\text{negative}|)} > P, \tag{2}$$

where P is the threshold of the minimum file trust ratio.

Based on these conditions, the massage is classified as follows:

(i) Unknown if it does not satisfy (1).

(ii) Trustworthy if it satisfies (1) and (2).

(iii) Untrustworthy if it satisfies (1) and does not satisfy (2).

To find suitable system wide values for T and P, we ran several experiments with different number of transactions and messages. It has been observed that the following hold:

(i) Value of T:

(a) T is very small: This is not enough to judge a message as trustworthy or untrustworthy because some malicious peers, in some periods, may gain good global trust values and can contribute to evaluating the file.

(b) T is very large: The message takes more time than required to be judged as trustworthy or untrustworthy.

Therefore, after several experiments, it has been concluded that the best value of T is

$$T = \frac{\text{average number of transcation}}{\text{number of files in the network}} * 5. \tag{3}$$

(ii) Value of P:

The best experimental value of P was 0.8.

(a) When P is smaller than that, a lot of untrustworthy messages were considered as trustworthy.

(b) When it is larger than that, a lot of the trustworthy messages were considered as untrustworthy due to malicious evaluators with high global trust values, in some periods.

3.3. Peer Reputation Manager. This component is responsible for the following:

(i) Evaluating the peer global trust value before allowing it to update the message trust value or choosing it as a download source, which is the case when not-enough information is available on the requested file.

(ii) Updating the peer global trust value after each transaction.

Following the same approach of [23], the current global trust value $t_i^{(k+1)}$ of peer i among the set of peers A_i which have downloaded files from peer i is calculated as

$$t_i^{(k+1)} = (1-a)\left(c_{1i}t_1^{(k)} + \cdots + c_{ni}t_n^{k}\right) + ap_i, \tag{4}$$

where

a is constant ≤ 1,

$$p_i = \begin{cases} \dfrac{1}{|P|}, & \text{if } i \in P \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

$|P|$ is the number of pretrusted peers.

The value of c_{ij} represents the local trust value, that is, how much peer i trusts peer j based on experiences with the set of peers B_i from which peer i has exchanged messages. It is calculated as follows:

$$c_{ij} = \begin{cases} \dfrac{\max\left(s_{ij}, 0\right)}{\sum_j \max\left(s_{ij}, 0\right)} & \text{if } \max\left(s_{ij}, 0\right) \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The value s_{ij} is defined as the difference between satisfactory sat transactions and unsatisfactory transactions, unsat:

$$s_{ij} = \text{sat}\left(i, j\right) - \text{unsat}\left(i, j\right). \tag{7}$$

In this way, each time peer i downloads a file from peer j, it rates the transaction as sat(i, j) if positive and unsat(i, j) if negative and keeps a record of how many times each was chosen.

4. Proposed Algorithm

Figure 1 shows the main steps in the AuthenticPeer algorithm starting from accepting a user request for a certain message through the user interface. The request would be handled by the message reputation manager, where the message trust value is calculated based on formulas (1) and (2) and the message is flagged accordingly as unknown, trustworthy, or untrustworthy. Untrustworthy message s would cause the peer to deny the transaction. Trustworthy messages will be accepted directly while unknown message s would require checking the peer trust value, based on formula (4). If the peer is untrusted, the transaction is denied. Otherwise, the message will be accepted. In all cases, the local trust values related to the source message and peer will be updated and the new values are shared with friends.

5. Comparative Study

The aim of this paper is to introduce a hybrid algorithm that takes advantages of both message-based and peer-based reputation systems to choose a suitable download source. The hypothesis was that such a hybrid approach can give better results when compared to trust management systems that use a single approach.

To test this hypothesis, a strictly designed evaluation framework was implemented. Software tools included Microsoft Visual C++ 2010 Express, jGRASP version 1.8.8_23, and NetBeans IDE NetBeans IDE. All simulation scenarios were implemented using an open-source simulation framework for P2P networks, the reputation management (RM) system simulator proposed in [13].

The performance of the AuthenticPeer algorithm was benchmarked using two reputation management algorithms:

(i) EigenTrust Algorithm [16].

(ii) Incremental EigenTrust Algorithm [13].

These systems were selected due to their fast convergence, low execution overhead, and available implementation, as well as their support of pretrusted peers [23], which is an essential component of the AuthenticPeer system. On the other hand, the reason for not including a message-based reputation system was that such systems are not scalable and require extensive computations, making them difficult to run in personal computers when experimented with.

Considered performance measures included the following:

(i) Fraction of inauthentic downloads calculated as

$$\text{Fraction of inauthentic downloads} = \frac{\#\text{ inauthentic downloads}}{\#\text{ completed transactions}}. \tag{8}$$

(ii) Success rate for good peers calculated as

$$\text{Success rate for good peers} = \frac{\#\text{ good peers authentic downloads}}{\#\text{ transactions by good peers}}. \tag{9}$$

Two sets of experiments were designed to evaluate scalability and sustainability [24] of the system following an approach similar to [16]:

(i) Variable number of malicious peers: the experimental parameter was the percentage of malicious peers. Its values were selected from the range [0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%]. The total number of peers and files was 88 and 2,000, respectively.

(ii) Variable number of messages: the experimental parameter was the number of messages. Its values were selected from the range [500, 1000, 1500, 2000, 2500, 3000]. The total number of malicious peers considered was 8 peers.

The rest of the parameters were considered to be constant at the following values:

(i) Number of transactions = 10,000 transaction.

(ii) Number of pretrusted peers = 3 peers.

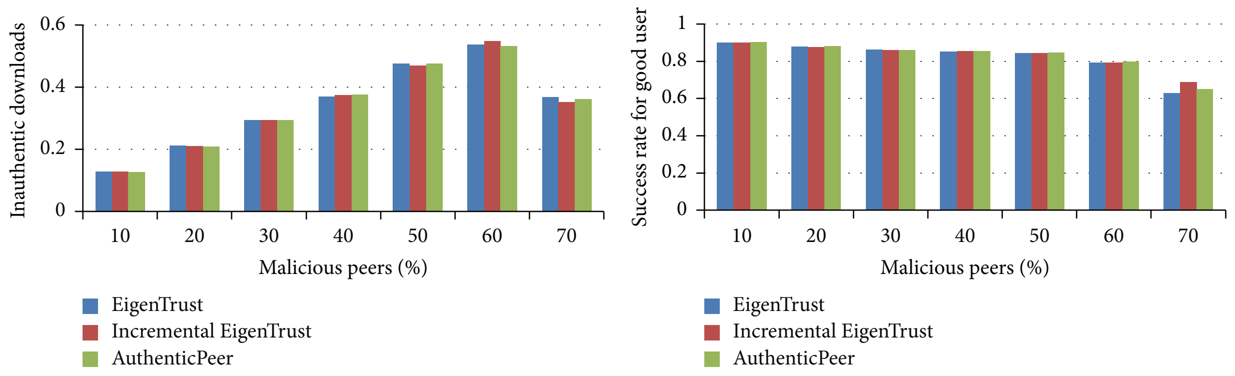
(iii) Number of good peers = 26 peers.

Four threat models were considered:

(i) Malicious individuals: when isolated malicious peers always provide inauthentic message when they are selected as download sources and lie in their feedbacks.

(ii) Malicious collective: when a collection of malicious peers always provide inauthentic message if they are selected as download sources. They know each other and support each other with high local trust values.

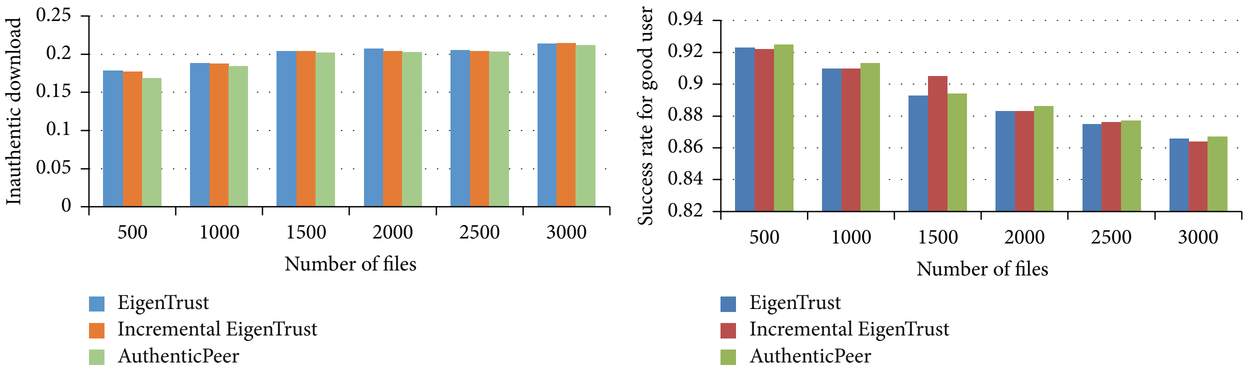
(iii) Camouflaged collective: when a collection of malicious peers try to get high trust values from good peers by, for example, providing authentic files when they are selected as download sources.



(a) Fractions of inauthentic downloads

(b) Success rate of good users

FIGURE 2: Effect of varying the percentage of malicious individuals.



(a) Fractions of inauthentic downloads

(b) Success rate of good users

FIGURE 3: Effect of varying the number of files in presence of malicious individuals.

(iv) Malicious spies: when collective malicious peers always provide authentic files if they are selected as a download source and use the reputation they gain to boost the trust values of another group of malicious peers, which only provide inauthentic message.

Two models of trustworthy peers were considered:

(i) Good peers, which provide honest feedback and have high probability to clean up invalid messages.

(ii) Pretrusted peers, which are identified at the very early stages of the system as trustworthy for all other peers.

The application model considered by this study was P2P file sharing application [25, 26]. This model was implemented by the RM simulator using an intelligent query model, where requested files should not have been requested in any previous transaction. Each peer had an equal opportunity of being a file requester and file selection was dedicated by the Zipf distribution [16]. All users have initial libraries of the same size and have an equal opportunity of requesting a file. It was assumed that the network is a static network, with a fixed number of users; that is, no new users can join or leave the network and it has an infinite bandwidth.

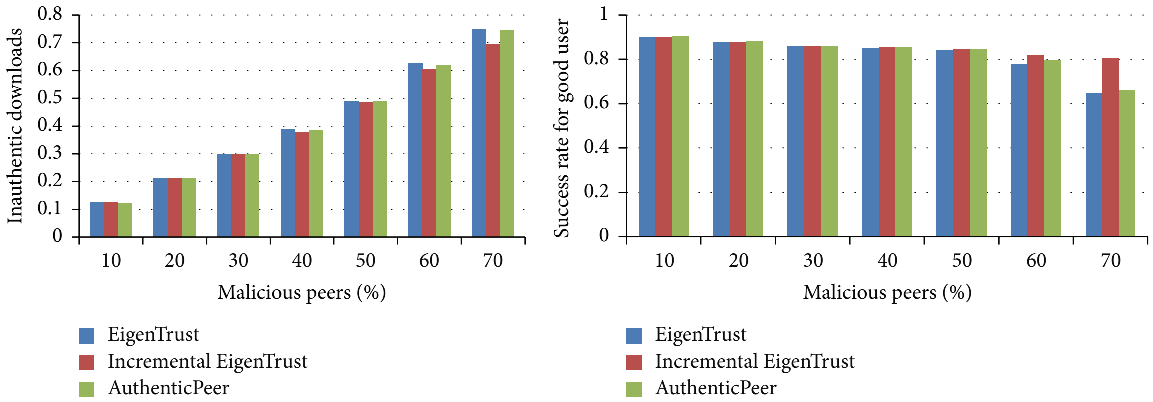
Hence, the total number of experiments was $(8 + 6) * 3 * 4 * 10 = 1{,}680$ experiments: two experiment sets, eight experiments in the first set and six experiments in the second. Each was carried out using the three systems: AuthenticPeer, EigenTrust, and Incremental EigenTrust, under the four threat models. All experiments were repeated at least ten times to increase the reliability of this experimental study; then the means of the results were calculated and analyzed.

6. Experimental Results and Discussion

This section compares the performance of the AuthenticPeer reputation system with the two benchmark algorithms: EigenTrust Algorithm and Incremental EigenTrust Algorithm. The experimental results are presented based on the four threat models: malicious individuals, malicious collective, camouflaged collective, and malicious spies.

Malicious Individuals. Figures 2 and 3 demonstrate how each algorithm performs under varying percentages of malicious individuals and number of files, respectively. In this model, a group of isolated malicious peers always provide inauthentic message when they are selected as download sources and lie in their feedbacks.

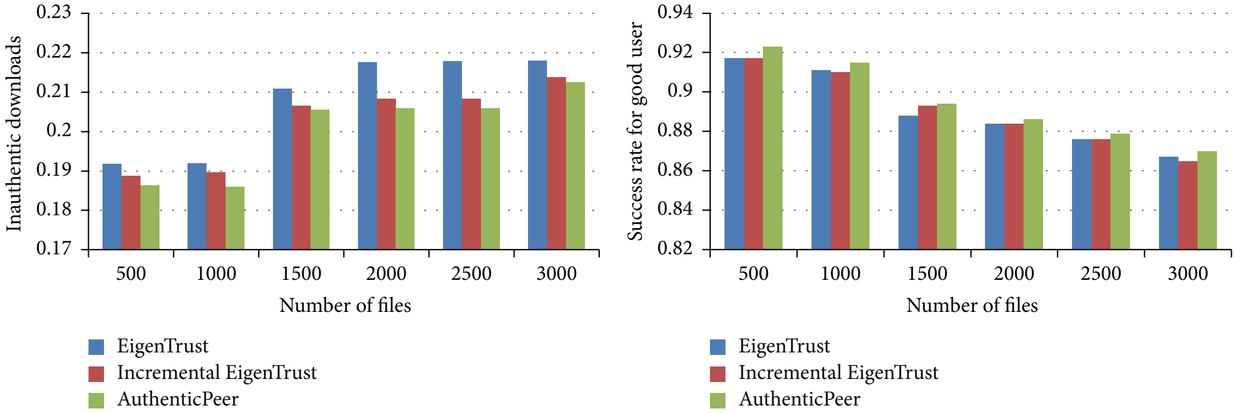
Based on Figure 2, it is clear that under varying percentages of malicious individuals, the three algorithms have



(a) Fractions of inauthentic downloads

(b) Success rate of good users

FIGURE 4: Effect of varying the percentage of malicious collective.



(a) Fractions of inauthentic downloads

(b) Success rate of good users

FIGURE 5: Effect of varying the number of files in presence of malicious collective.

almost identical performance in terms of the fraction of inauthentic message, illustrated in Figure 2(a), and the success rate of good users illustrated in Figure 2(b).

Figure 2(a) shows that, as expected, when the percentage of malicious peers increased, the fraction of inauthentic downloads increased too, as the malicious peers will spread more inauthentic message. However, unexpectedly, when the fraction of malicious peers reached 70%, the fraction of inauthentic downloads remarkably dropped. This suggests that when the number of malicious individuals reaches a certain level, they can be clearly identified based on their consistent behavior and they will not be considered as candidate sources for downloading files.

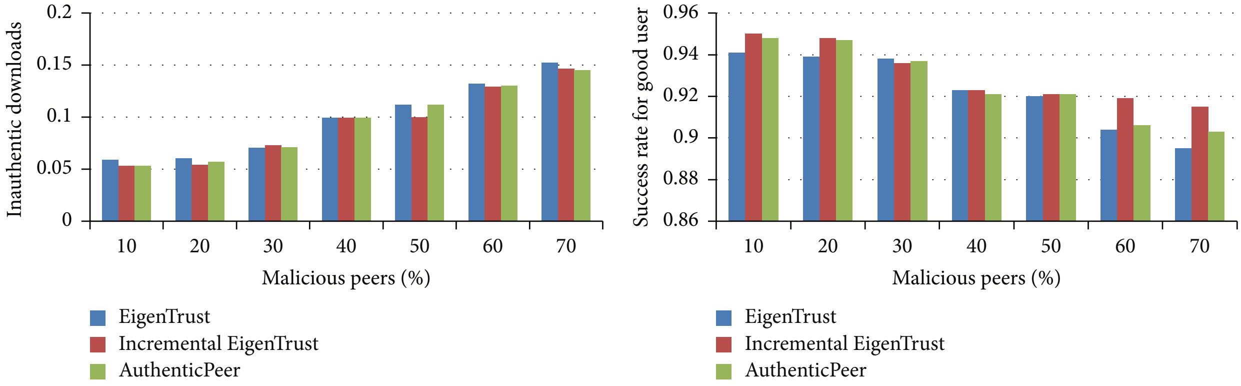
Figure 2(b) illustrates that, as anticipated, the success rate of the good users decreased equally gradually, under all algorithms, as the percentage of malicious peers increased. This is because increasing the number of malicious peers would decrease sources of authentic files, which was negatively reflected on the number of successful transactions completed by good peers.

Figure 3(a) shows that, in general, the fraction of inauthentic downloads, based on the three algorithms, was marginally increased when the number of files in the system increased which suggests that the number of files does not have a direct impact on the number of inauthentic downloads in the case of malicious individual's threat model.

Figure 3(b) shows gradual reduction, based on the three algorithms, in the success rates of good users as the number of files in the system increased. This might be attributed to the small amount of information that will be available about the authenticity of each file when the number of transactions is constant but the number of files is increasing.

Malicious Collective. Figures 4 and 5 illustrate how each of the three algorithms, AuthenticPeer, EigenTrust, and Incremental EigenTrust, performs under varying percentages of malicious collective and number of files, respectively. In this model, malicious peers form a collective and assign each other high trust values. However, they are rarely chosen as download sources. As a result, they cannot gain high global trust values because of the presence of pretrusted peers which break up malicious collectives.

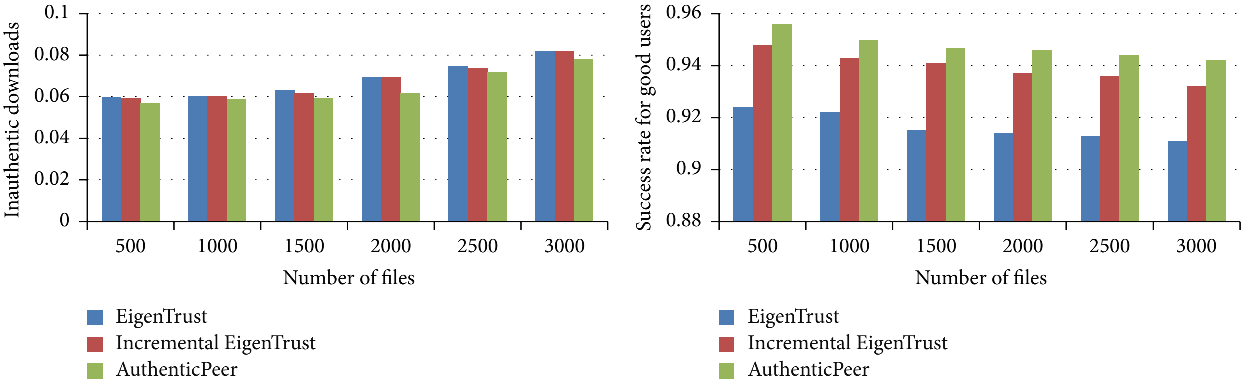
Based on Figure 4, it is clear that the three algorithms have almost identical performances, with slightly better performance by the Incremental EigenTrust and least performance by the EigenTrust, in terms of the fraction of inauthentic



(a) Fractions of inauthentic downloads

(b) Success rate of good users

Figure 6: Effect of varying the percentage of camouflaged collective.



(a) Fractions of inauthentic downloads

(b) Success rate of good users

Figure 7: Effect of varying the number of files in presence of camouflaged collective.

message, as illustrated in Figure 4(a), and the success rate of good users, as illustrated in Figure 4(b).

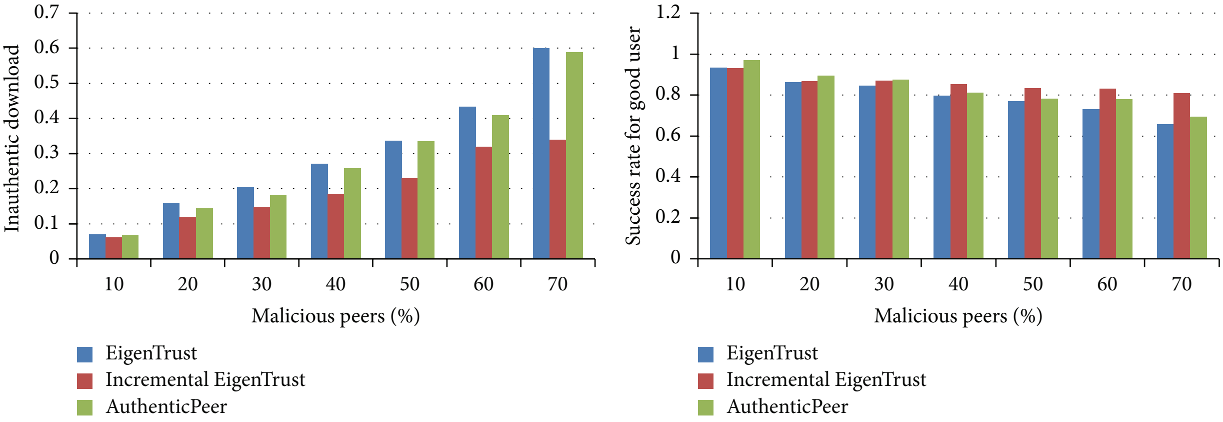
Figure 4(a) shows that, as expected, when the percentage of malicious collectives increased, the fraction of inauthentic downloads also increased. Figure 4(b) shows that when the percentage of malicious collectives increased, the success rates of good peers decreased. These observations indicate similar effects of malicious collectives to malicious individuals. This is because malicious collectives help in spreading inauthentic message by giving other malicious peers high trust values.

Figure 5(a) demonstrates superior performance of AuthenticPeer when compared to the benchmark algorithms, specially the EigenTrust. Although the AuthenticPeer algorithm chooses the source for download in the same way as the EigenTrust Algorithm, after T number of transactions, the untrustworthy files will be identified so they will not be downloaded by good peers. This also explains the higher success rates of good peers under AuthenticPeer as illustrated in Figure 5(b).

Camouflaged Collective. Figures 6 and 7 demonstrate how each algorithm performs under varying percentages of camouflaged collectives and number of files, respectively. In this model malicious peers work as a group to get high trust values from good peers by, for example, providing authentic files when they are selected as download sources.

Figure 6(a) shows that, based on the three algorithms, the fraction of inauthentic downloads increased gradually when increasing the percentage of malicious peers with minor difference between the three algorithms. Figure 6(b) shows how the success rate, based on the three algorithms, was slightly decreased as the percentage of malicious peers increased. However, the Incremental EigenTrust showed slightly better performance, especially with large percentages (>50%) of malicious peers. This might be attributed to the snapshot strategy adopted by this algorithm. The snapshot included an increased number of malicious peers, which helped identify them more easily and bias good peers to avoid downloading from them.

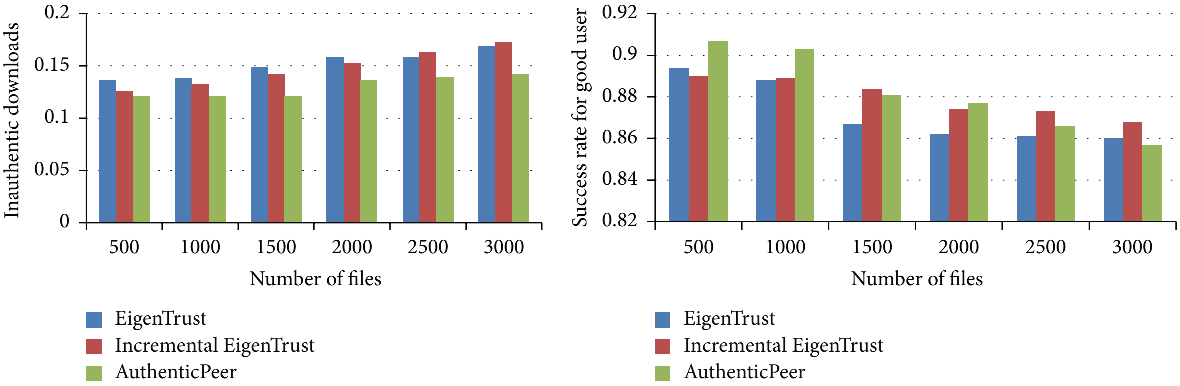
Figure 7(a) shows a marginal increase in the fraction of inauthentic downloads, by the three algorithms, as the number of files in the system increased, with slightly enhanced performance in the case of AuthenticPeer. This observation is similar to the case in malicious individuals of Figure 3(a). In Figure 7(b), the variations in performance between the three algorithms are clearer, with superior success rate when AuthenticPeer was used. This clearly indicates that having



(a) Fractions of inauthentic downloads

(b) Success rate of good users

FIGURE 8: Effect of varying the percentage of malicious spies.



(a) Fractions of inauthentic downloads

(b) Success rate of good users

FIGURE 9: Effect of varying the number of files in presence of malicious spies.

both peer and file trust values to decide the download source would effectively marginalize the role of the unreliable camouflaged peers.

Malicious Spies. Figures 8 and 9 illustrate how each algorithm performs under varying percentages of malicious spies and number of files, respectively. In this model malicious peers form a collective malicious and always provide authentic files if they are selected as a download source. They use the reputation they gain to boost the trust values of another group of malicious peers, which only provide inauthentic message.

Based on Figure 8(a), when the percentage of camouflaged malicious peers increased, the fraction of inauthentic downloads, by the three algorithms, increased too, as expected. However, it is clear that the Incremental EigenTrust Algorithm has successfully marginalized the effects of such a model of malicious peers when compared to AuthenticPeer and EigenTrust. Yet, this is not the case when the success rate of good peers is the main concern, as illustrated in Figure 8(b). In this case, the three algorithms showed almost similar performances, with slightly better performance by AuthenticPeer for low percentages, although the Incremental EigenTrust performed better as the percentage increased.

The same observation is also clear in Figure 9(b), as the number of files increased with more variations in performance among the three algorithms. Based on Figure 9(a), AuthenticPeer maintained the lowest fraction of inauthentic downloads, when compared to the benchmark algorithms, regardless of the number of files.

The poor performance showed by the EigenTrust Algorithm, when the camouflaged collective peers are considered, highlights the problem associated with it, when mistakenly choosing a malicious peer as a download source, which would help in spreading inauthentic message in the network.

The main findings regarding the performance of the three algorithms, under each of the four considered threat models and different conditions of a number of malicious peers and files in the system, are summarized in Table 2. The term "identical" means the three algorithms showed very similar performances, while "varied" indicates that the performances differed among the three. Based on the table, it is clear that AuthenticPeer system is recommended for P2P networks,

Table 2: Summary of the best performance for each threat model.

Threat model	Best performance			
	Varying percentage of malicious peers		Varying number of files	
	Inauthentic downloads	Success rate	Inauthentic downloads	Success rate
Individual malicious	Identical	Identical	Identical	Identical
Malicious collective	Identical	Identical	AuthenticPeer	AuthenticPeer
Camouflaged collective	Identical	Incremental EigenTrust	AuthenticPeer	AuthenticPeer
Malicious spies	Incremental EigenTrust	Varied	AuthenticPeer	Varied

especially with the threat models: malicious collective, camouflaged collective, and malicious spies, regardless of the number of files in the network.

7. Conclusion

Reputation management systems are employed to build trust in P2P systems. They usually use the quality of either peers or messages to identify safe sources for downloads. In this paper, the AuthenticPeer reputation system is proposed for P2PWSN. It employs a unique approach that considers the quality of both messages and peers.

Experimental results suggest that AuthenticPeer is recommended for P2PWSN networks, especially with the threat models: malicious collective, camouflaged collective, and malicious spies, regardless of the number of files in the network.

The main contributions of this paper can be summarized as follows:

(i) Design and implementation of a new reputation system based on hybrid technique, an approach that has not been considered before in similar systems.

(ii) Comprehensively evaluating the proposed system based on a strictly designed evaluation framework.

(iii) Review and classification of considerable literature in reputation management systems.

For future work, we intend to extend and improve the AuthenticPeer system in the following directions:

(i) Considering more threat models such as selfish peers and denial-of-service attack.

(ii) Reordering the main steps of the algorithm to start by checking the quality of the peer and then quality of the message. This is expected to remarkably improve the performance.

(iii) Enhancing the message reputation manager by considering an algorithm other than [21] as a baseline.

(iv) Enhancing the peer reputation manager by considering an algorithm other than [16] as a baseline, specifically the Incremental EigenTrust as it has shown better performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was funded by the Long-Term Comprehensive National Plan for Science, Technology and Innovation of the Kingdom of Saudi Arabia, Grant no. 11-INF1895-08.

References

[1] C. Sastry, C. Ma, M. Loiacono, N. Tas, and V. Zahorcak, "Peer-to-peer wireless sensor network data acquisition system with pipelined time division scheduling," in *Proceedings of the IEEE Sarnoff Symposium*, pp. 1–4, IEEE, Princeton, NJ, USA, March 2006.

[2] M. Su and C.-H. Chi, "Application networking on peer-to-peer networks," in *Proceedings of the 14th International World Wide Web Conference (WWW '05)*, pp. 1134–1135, ACM, Chiba, Japan, May 2005.

[3] Gnutella, The Gnutella Protocol Specifications v0.4. Document Revision 1.2, April 2013, http://cryptnet.net/fsp/cpcd/gnutella_protocol_0.4.pdf.

[4] W. Dong, Y. Yang, and W. Zhang, "On the selection of information sources for gossip spreading," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 276014, 10 pages, 2015.

[5] D. Granlund, C. Åhlund, and P. Holmlund, "EAP-swift: an efficient authentication and key generation mechanism for resource constrained WSNs," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 460914, 12 pages, 2015.

[6] G. Yang, L. Zhang, Z. Tan, H. Yu, and S. Li, "A new method of trust inference based on Markov model for peer-to-peer network," in *Proceedings of the 12th IEEE International Conference on Computer and Information Technology (CIT '12)*, pp. 349–354, IEEE, Chengdu, China, October 2012.

[7] Z. Li, H. Shen, and K. Sapra, "Leveraging social networks to combat collusion in reputation systems for peer-to-peer networks," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1745–1759, 2013.

[8] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.

[9] A. Boukerch, L. Xu, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2413–2427, 2007.

[10] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, article 15, 2008.

[11] L. Huang, L. Li, and Q. Tan, "Behavior-based trust in wireless sensor network," in *Advanced Web and Network Technologies,*

and Applications, vol. 3842 of *Lecture Notes in Computer Science*, pp. 214–223, Springer, Berlin, Germany, 2006.

[12] H. Chen, H. Wu, X. Zhou, and C. Gao, "Agent-based trust model in wireless sensor networks," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, pp. 119–124, IEEE, Qingdao, China, August 2007.

[13] A. G. West, S. Kannan, I. Lee, and O. Sokolsky, "An evaluation framework for reputation management system," in *Trust Modeling and Management in Digital Environments: From Social Concept to System Development*, Y. Zheng, Ed., pp. 282–308, IGI Global, Helsinki, Finland, 2009.

[14] L. Xiong and L. Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[15] I.-R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1200–1210, 2013.

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

[17] D. Donato, C. Castillo, M. Paniccia, G. Cortese, M. Selis, and S. Leonardi, "New metrics for reputation management in P2P networks," in *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb '07)*, pp. 65–72, Banff, Canada, May 2007.

[18] N. Chiluka, N. Andrade, D. Gkorou, and J. Pouwelse, "Personalizing EigenTrust in the face of communities and centrality attack," in *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA '12)*, pp. 503–510, IEEE, Fukuoka, Japan, March 2012.

[19] X. Fan, M. Li, Y. Ren, and J. Ma, "Dual-eigenrep: a reputation-based trust model for P2P file-sharing networks," in *Proceedings of the 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC)*, pp. 358–363, Xi'an, China, October 2010.

[20] A. Selcuk, E. Uzun, and M. R. Pariente, "A reputation-based trust management system for P2P networks," in *Proceedings of the 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid '04)*, pp. 251–258, April 2004.

[21] S. Y. Lee, O.-H. Kwon, J. Kim, and S. J. Hong, "A reputation management system in structured peer-to-peer networks," in *Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '05)*, pp. 362–367, June 2005.

[22] K. Walsh and E. G. Sirer, "Fighting peer-to-peer SPAM and decoys with object reputation," in *Proceedings of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON '05)*, pp. 138–143, ACM, Philadelphia, Pa, USA, August 2005.

[23] R. Zhou and K. Hwang, "PowerTrust: a robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, April 2007.

[24] S. K. Dhurandher, S. Misra, M. S. Obaidat, I. Singh, R. Agarwal, and B. Bhambhani, "Simulating peer-to-peer networks," in *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '09)*, pp. 336–341, IEEE, Rabat, Morocco, May 2009.

[25] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," in *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P '03)*, pp. 150–157, Linkoping, Sweden, September 2003.

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