SecTrust: A Trust and Recommendation System for Peer-2-Peer Networks

David Airehrour
Nelson Marlborough Institute of Technology, david.airehrour@nmit.ac.nz

Jairo Gutierrez
Auckland University of Technology, jairo.gutierrez@aut.ac.nz

Sayan Kumar Ray
Manukau Institute of Technology, sayan.ray@manukau.ac.nz

Follow this and additional works at: http://aisel.aisnet.org/confirm2018

Recommended Citation
Airehrour, David; Gutierrez, Jairo; and Ray, Sayan Kumar, "SecTrust: A Trust and Recommendation System for Peer-2-Peer Networks" (2018). CONF-IRM 2018 Proceedings. 41.
http://aisel.aisnet.org/confirm2018/41
**SEC\textsc{Trust}: A Trust and Recommendation System for Peer-2-Peer Networks**

David Airehrour  
Nelson Marlborough Institute of Technology  
New Zealand  
david.airehrour@nmit.ac.nz

Jairo A. Gutiérrez  
Auckland University of Technology  
New Zealand  
jairo.gutierrez@aut.ac.nz

Sayan K. Ray  
Manukau Institute of Technology  
New Zealand  
sayan.ray@manukau.ac.nz

**Abstract**

The concept of Trust has been applied in many information systems such as e-commerce, social networks and smart grid networks. It is currently getting attention on the Internet of Things. The fundamental concept in Trust, is the ability of a user to evaluate its neighbour, or a recommended neighbour with the hope of being able to trust that user. In this paper, we propose SecTrust, which is a trust-based system that computes trust values as a time-based packet forwarding behaviour of a node in a peer-2-peer (P2P) network. It uses a trust rating iteration and rating scheme to select only trusted nodes for communication while isolating malicious nodes in the network. A penalty is also introduced in our system that penalizes the continuous misbehaviour of a node in the network. Simulation evaluation results show the superior performance of SecTrust over other trust systems in isolating malicious peers, improving information interaction success rates, enhancing trust formation, and promoting better reputation between trusted peer nodes.

**Keywords**
P2P, Trust, SecTrust, Recommendation System.

1. **Introduction**

Peer-to-Peer (P2P) networking is a distributed architectural system, which divides tasks among peer nodes in a network. Peers willingly provide a part of their resources like storage, processing capacity or network bandwidth to other peer nodes. This is achieved with no central coordination from any servers or central coordination mechanisms. Typical application areas of P2P include: file sharing (Bit torrent and Gnutella); Peer-to-peer assisted streaming solution in multimedia (P2PTV and PDTP protocols); other areas include e-commerce and social media where P2P technology is used as a recommender system for user verification.

In a P2P system, nodes depend on each other for seamless communication and transactions. However, this becomes a security concern when malicious peers abuse this relationship. To evenly distribute workload in P2P, a peer node exposes some of its internals to its neighbours, but attackers normally leverage on this to compromise the P2P network. Due to this compromise, the activities of the malicious peers lead to network problems including: the inhibition of P2P traffic (distributed denial-of-service), privacy and identity issues, network poisoning and the sharing and propagation of inauthentic files. Furthermore, due to the lack of a central
coordination system, P2P networks are often prone to various malicious attacks like Sybil, Blackhole, Greyhole and Wormhole attacks. To address these attacks and maintain a consistent service delivery in P2P networks, the role of Trust and Recommendation Systems have become important. However, the challenge with most Trust and Recommendation systems in P2P networks, is the absence of effective trust mechanisms that could moderate peer node behaviour to effectively protect the network from trust related attacks like, good-mouthing, bad-mouthing and self-endorsement attacks. This is still an open research problem (Ali, Hong, & Kim, 2017).

This paper proposes SecTrust; a trust-based system that mitigates trust related attacks and enhance trust reputation among nodes in a Sybil scenario. Furthermore, SecTrust scales well from a small P2P to a large P2P network. SecTrust is based on the concept of social trust in humanity and uses the computed trust values of neighbouring nodes to detect and isolate malicious actors in a distributed P2P network. The paper further provides a system of validating the trustworthiness of peer nodes in P2P networks hence, mitigating the influence of malicious attacks in a P2P network. The remainder of this paper is organised as follows. Section 2 introduces the concept of trust in P2P networks, and reviews some of the notable trust and recommendation systems used in peer-to-peer networks. Section 3 intrces SecTrust and describes its operational framework and key design considerations for a 2P platform. Section 4 presents the simulation studies to evaluate the performance of SecTrust and compares the proposed trust model against EigenTrust. Finally, Section 5 concludes the paper.

2. Related Work
Trust is a social concept used among humans and it shows the level of confidence, reliability, assurance or even faith one party has towards another. The trust value of a peer (node) is calculated and evaluated based on the assurance or confidence a node has towards its neighbour(s) while the reputation is a function of the feedback from recommendations from its neighbour after due authentication and evaluation of the recommendation process (J. P. Wang, Bin, Yu, & Niu, 2013). A notable aspect of a trust and recommendation system is the evaluation
of both the service requesting peers and the service provider peers, and this is necessary to ensure the trustworthiness of both parties and by extension the whole system. The formation of trust among peers in a distributed P2P system could help filter out malicious peers in a network. Trust in P2P networks plays a vital role in the development of a stable reputation management, and it facilitates the management and maintenance of such a network, especially when the number of nodes grows on a massive scale. Trust as a mechanism for securing P2P networks is an active area of research, and it remains a very important and challenging area for further studies (Valdez, Zieﬂe, Verbert, Felfernig, & Holzinger, 2016).

2.1 Trust Management and Recommendation Models

2.2.1 EigenTrust

EigenTrust (Kamvar, Schlosser, & Garcia-Molina, 2003) is a renowned Trust and Recommendation system, which is implemented in Gnutella, a ﬁle-sharing site with millions of users (Chang, 2017). It utilizes local normalization together with global convergence by means of a vector multiplication matrix. It is based on past interactions, where past interactions could be viewed as a square matrix or an \( n \times n \) matrix with \( n \) deﬁned as the number of peers in the network. In computing the trust between two parties, the number of positive (pos) and negative (neg) communications is stored in what is referred to as relations. The EigenTrust model assigns more trust weight to some parties in the network, which are presumed pre-trusted nodes. EigenTrust makes available the global recommendation scores to ﬁle requesters when ﬁle owners are chosen. Through its trust computation system, EigenTrust is able to detect and isolate malicious peer nodes.

Studies have shown EigenTrust to be weak in isolating masquerading malicious peers perpetrating themselves as trusted peers (Girdhar, Kumar, & Singh, 2013). This promotes the download of spurious ﬁles and an imprecise peer trustworthy assessment. In EigenTrust, a peer node seeking a speciﬁc ﬁle could continuously download ﬁles until it ﬁnds the precise ﬁle. This process amounts to excessive bandwidth usage, application efﬁciency decline, and creates a platform for the transmission of downloaded spurious ﬁles to other unsuspecting peer nodes in the network. Table 1 provides a summary of various trust management models, metrics employed, and the attacks addressed. From Table 1 and discussions above, it can be noted that research work on trust management and recommendation have mostly focused on successful and unsuccessful node transactions, feedback and historical observation of nodes in P2P networks. Clearly, the modelling of trust in P2P networks from a communication perspective still requires more research, and that is the focus of this work. For further reading, a survey of some notable trust and recommendation systems have been presented in (Agrawal & Mishra, 2012; Rashed, Ullah, & Yasmin, 2013) and it will be beneﬁcial for additional reading.

3. SecTrust

Secure Trust (SecTrust) is a peer-to-peer lightweight trust-based system, which identiﬁes and isolates malicious attacks in P2P networks. It essentially computes and evaluates the trustworthy behaviour of a peer node. A node’s trustworthiness in SecTrust is based upon the evaluation from its neighbouring nodes while providing service (recommendations and indirect trust values) to other nodes. This trustworthiness reﬂects the level of reliability or dependability (trust) that a peer node in the network has on another node. The trustworthiness of a node is evaluated as a time-based successful packet exchange between neighbour nodes, and the positive packet acknowledgements with a continuous trust observation of connected neighbour nodes.
In this system, every node computes the trustworthiness of its direct neighbours based on the computed direct trust value (the packet forwarding behaviour of a node towards its neighbour(s)) and the recommended trust value. Neighbours with higher trust values are chosen for communication while nodes with lower trust values are categorised as either malicious, compromised, or even selfish nodes that seek to preserve their resources, such as battery power. The proposed SecTrust system comprises of five major processes: trust calculation process, trust monitoring process, process to detect and isolate malicious nodes, trust rating process, and trust backup/recuperation process.

The preliminary concept of SecTrust was introduced by us in (Airehrour, Gutierrez, & Ray, 2016), but no comparison was made with other global trust and recommendation systems. Our motivation for providing a P2P trust system for reputation management as proposed in our first paper (Airehrour et al., 2016), is based on the fact that malicious devices could perpetrate “prejudiced” attacks to either destroy the reputation of unsuspecting peer devices like IoT.

Table 1: A summary of trust and recommendation systems with metrics used and attacks addressed
sensors that provide seamless network communication or provide some essential services in the network. In addition, in applying our trust-based system to distributed IoT networks, which comprise of large number of sensors and devices working together to provide a variety of services, there is the risk that some will typically be malicious in their activities. Thus, a reputation management platform that scales across a large sized-network will prove invaluable to mitigating the attacks. Although our initial paper (Airehrour et al., 2016) focused on a secure communication in IoT routing, this proposed study however, is a refinement of that work with a focus on reputation management in P2P networks. This is because this proposed trust-based system, is designed as a conceptual trust system for reputation management in P2P communication thus, finding relevance and application in P2P networks. Furthermore, a comparison is made with a similar P2P trust algorithm (the EigenTrust).

### 3.1 Quantifying Trust in SecTrust

Many trust properties have been employed for trust evaluation in Wireless Sensor Networks (Bao & Chen, 2012; Bao et al., 2012; Gu, Wang, & Sun, 2014; J. P. Wang et al., 2013; Wu et al., 2014). However, for this research paper key metrics have been formulated which are used for the computation, evaluation and trust formation among peer nodes. The key metrics for the formation and computation of trust in SecTrust include:

1. The positive interaction between nodes in order to determine a node’s packet forwarding behaviour towards its neighbours. This is determined by a node forwarding the assigned packet to its destination or to the next trusted hop.
2. Recommended trust of a node to another node as the product of the indexed trust values among the nodes.
3. The evaluation time of a node’s trustworthiness.
4. A penalty weight attached to any misbehaving node in the network.

SecTrust uses the concept of a threshold-based trust broadcast (Dai & Gong, 2010). According to this concept, trustworthy nodes can be broadcast throughout the network by i) maintaining effective communication only among trusted nodes and, ii) ensuring the transmission of only trustworthy information to neighbour nodes in the network based on the trust threshold. SecTrust ensures this by validating the trustworthy nature of every forwarding node in the network and by securing the source of route information transmitted. Furthermore, we provide a method to rank the best trust information to optimize the performance of reputation-based communication among peer nodes. The SecTrust system observes various trust properties while assigning different weights to them. It is particularly important to assign more weight to how a node is currently evaluated rather than its history or past observable behaviour. This is needed to identify and isolate malicious nodes that may seek to build a good reputation over time and then later begin to perpetrate their malicious activities. Thus, the SecTrust system demonstrates that trust converges to the extremes (i.e. highly trusted nodes tend to maintain high trust values while malicious nodes quickly drift to low trust values). In addition, a repository of nodes with high trust values are formed. Finally, based on the penalty introduced for misbehaving nodes, the trust value between any two adjoining nodes that do not optimally trust each other tends to zero. A detailed description of the SecTrust processes and its components are presented next.
3.2 SecTrust Process
SecTrust is a composition of five systemic processes that operate in unison to provide trust and reputation management system for P2P nodes. A description of the inner workings of each process is discussed.

3.2.1 Trust Calculation Process
In the design of SecTrust, a trustor node evaluates a trustee node. It uses the trust value derived to judge if the trustee node is sufficiently reliable to fulfill an assigned mission as assessed by the five-tuple trust level band specified in Table 2. More specifically, the trust computation represents reliability, dependability, competence, and successful positive interactions or positive recommendations on performance of tasks sent by nodes from direct or indirect interactions with other nodes. Direct and recommended trust relationships could exist between two or more nodes. Each node gathers the direct trust values of directly linked neighbour nodes and the recommended trust values of neighbours that are one or more hops away. Here, we describe these two types of trusts. Figure 1 illustrates the relationship between direct and recommended trusts. In direct trust, node $i$ computes the trust value of node $j$ since it has a direct relationship with node $j$. Thus, when there is a positive direct interaction between node $i$ and node $j$, trust is established, and a value is computed (i.e., node $j$ successfully forwards packets sent by node $i$ to either the destination or the next trusted hop node ($m$)). This relationship is expressed in Equation 1 below. In order to ensure nodes, behave as expected, a penalty ($\beta$) is introduced to penalize any misbehaving node. $\beta$ is a constant, which defines the penalty weight applied to any misbehaving node.

\[
DT(N_i, N_j) = \frac{PF_i(t)}{PF_i(t) + \beta[PS_i(t) - PF_i(t)]}
\]  

(1)

In the context of Figure 1, the Recommended trust between nodes $i$ and $m$ is described as which is the product of the direct trust values between nodes $i,j$ and nodes $j,m$ as shown in Equation 2 below. This scheme is also used for nodes 2-hops and beyond.

\[
RT(N_i, N_m) = DT(N_i, N_j) \times DT(N_j, N_m)
\]  

(2)

Figure 1: Relationship between Direct and Recommended trusts
3.2.2 Trust rating Process

SecTrust employs the concept of a threshold-based trust rating system. In this concept, only nodes with high trust values will be considered for communication decisions. After the trust values have been determined, a trust rating system is further adopted to rank the highest to the lowest trust values obtained during the trust calculation process. This helps in not only utilising highly rated (trusted) nodes for communication such as routing, but also in detecting and isolating misbehaving nodes, which seek to adjust their trust values maliciously.

To estimate the trust values, we developed a five-tuple trust level system and deduce its trust degree by using threshold boundary judgment. A five-tuple trust level is assigned to each node, and it is defined as: \( V = \{v_1, v_2, v_3, v_4, v_5\} \). This is expressive of the various trust levels as: [“no trust”, “poor trust”, “fair trust”, “good trust”, “complete trust”], respectively, and is represented in Table 2.

<table>
<thead>
<tr>
<th>Five tuple (( V ))</th>
<th>Trust level</th>
<th>Range of positive relations (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_1 )</td>
<td>No trust</td>
<td>( \geq 0 ) and ( \leq 25 )</td>
</tr>
<tr>
<td>( v_2 )</td>
<td>Poor trust</td>
<td>( \geq 26 ) and ( \leq 50 )</td>
</tr>
<tr>
<td>( v_3 )</td>
<td>Fair trust</td>
<td>( \geq 51 ) and ( \leq 75 )</td>
</tr>
<tr>
<td>( v_4 )</td>
<td>Good trust</td>
<td>( \geq 76 ) and ( \leq 90 )</td>
</tr>
<tr>
<td>( v_5 )</td>
<td>Complete trust</td>
<td>( \geq 91 ) and ( \leq 100 )</td>
</tr>
</tbody>
</table>

Table 2: A Five-Tuple Trust Level Band

SecTrust considers nodes in the \( v_5 \) tuple as perfect nodes for trustworthy communication while nodes in the \( v_4 \) tuple, although good, could possibly have some mixed-element behaviours. SecTrust does not rely on nodes in the \( v_3, v_2, \) and \( v_1 \) tuples, but to maintain stable communications, it uses nodes in the order of \( v_3 \) and \( v_2 \), if nodes in the \( v_5 \) and \( v_4 \) categories are unavailable.

3.2.3 Process to Detect and Isolate Malicious Nodes

The most important task of any secure information system is to guarantee the confidentiality, integrity, availability and authenticity of the information transmitted. This issue becomes challenging in some P2P networks, because (i) the peer nodes are resource-constrained, (ii) the number of connected peer devices are numerous, and (iii) some P2P networks have lossy wireless transmission links. The work done in this research focuses on the detection and isolation of insider attacks that are rather more difficult to detect. This is because the malicious nodes form part of the network and are privy to every detail of the network process. This study uses node overhearing and monitoring methods to determine anomalous traffic routes towards a node as the possible indication of a rank attack or a colluding attack. Furthermore, SecTrust, being a trust-based system, can be prone to trust related attacks like Good-mouthing (endorsing a misbehaving node) and Bad-mouthing attacks (maligning a good node). To address this, SecTrust has an internal mechanism to defend against the above mentioned attacks in the following three ways:

- By allocating a higher weightage (80%) to the current trust value of the recommender node rather than the recommended node;
- If the current trust value of the recommender node falls short of the trust threshold value required, then both the recommending and the recommended nodes are discarded, and alternative nodes are sought.
• However, if the recommender node’s trust value is above the trust threshold then the computed trust value of the recommending node is multiplied with the computed direct trust value of the recommended node to get the final weighted computed trust value of the recommended node.

3.2.4 Trust Monitoring Update Process
This process is an observation and update phase where every node gathers the trust information of their immediate and distant neighbours based on (a) direct relationships and (b) the recommendations between the nodes. SecTrust also updates the trust values of nodes and in doing that it employs any of the two methods namely; the periodic trust update and the reactive trust update.
• The periodic trust update is based on a given set time when trust values are re-computed for updates.
• The reactive trust update is intended for the spatially connected and distributed nodes that trigger network communication updates (e.g., route updates) and are hence, reactive based on their sparse communication links.

4. Trust Framework and Performance Evaluation
In a peer-to-peer network, an evaluating peer (trustor) evaluates another peer’s (trustee) trustworthy level to decide on its reliability to perform certain expected tasks. Nodes in trust models aim to locate a reliable party that will cooperate with them to perform specific tasks. In this Section, we assess the general behaviour and performance efficiency of SecTrust against EigenTrust (Kamvar et al., 2003). We have used the Quantitative Trust Management (QTM) P2P trust simulator (West, Kannan, Lee, & Sokolsky, 2010) to evaluate the proposed SecTrust system while Table 3 lists the simulation parameters.

We have selected the EigenTrust model to compare against SecTrust due to the following reason: EigenTrust is based on a P2P network design and is efficient, lightweight, scalable and deployable as an embedded system. Furthermore, EigenTrust as a global reputation management system, has been implemented in file sharing networks like Gnutella. Although various improvements to EigenTrust have been put forward by researchers, we are unaware however, of any that has been adopted as a replacement to EigenTrust. Therefore, verifying the efficacy of SecTrust against EigenTrust in this study is appropriate and it provides a meaningful comparison. Furthermore, even though security covers confidentiality, integrity and availability, this study addresses availability by isolating malicious nodes that undermine the network and inhibit the availability of resources among nodes. Hence, the comparison in this study is based on the trust effectiveness between SecTrust and EigenTrust. Thus, a higher trust effectiveness between nodes depicts better reputation among nodes and hence, better availability of resources and communication between nodes hence, improved network performance.

4.1 Embedding SecTrust into the QTM P2P Trace Simulator
The P2P-trace simulator has pre-developed Application Programming Interfaces (API’s) that makes the inclusion of new trust and recommendation systems easy. The trace simulator comprises of four main modules namely: the CORE_LIB, the GENERATOR_LIB, the SIMULATOR_LIB and the TRUST_SYSTEM_LIB. SecTrust is developed in Java and included in the TRUST_SYSTEM_LIB of the simulator.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Users and Number of Transactions</td>
<td>50 users with 10,000 transactions</td>
</tr>
<tr>
<td></td>
<td>100 users with 10,000 transactions</td>
</tr>
<tr>
<td></td>
<td>500 users with 100,000 transactions</td>
</tr>
<tr>
<td></td>
<td>1,000 users with 100,000 transactions</td>
</tr>
<tr>
<td>Number of Files</td>
<td>5,000</td>
</tr>
<tr>
<td>Max. User Connections</td>
<td>2</td>
</tr>
<tr>
<td>Bandwidth Period</td>
<td>1</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.005</td>
</tr>
<tr>
<td>Zipf Constant</td>
<td>0.4</td>
</tr>
<tr>
<td>Pre-Trust;ed Users</td>
<td>5</td>
</tr>
<tr>
<td>Good Behaving Users</td>
<td>40</td>
</tr>
<tr>
<td>Purely Malicious Users</td>
<td>10</td>
</tr>
</tbody>
</table>

**Table 3: Trust Simulation Parameter Settings**

The QTM simulator provides some key metrics that are used as indices in determining the effectiveness of the different trust and recommendation systems. According to the framework in (West et al., 2010), the effectiveness of a trust and recommendation system is given as the ratio of the number of valid transactional files received by users (nodes) to the number of transactions attempted by good users (despite the malicious operations of bad users in the network). Equation 3 denotes this.

$$\text{Effectiveness Metric} = \frac{\text{Number of good user successes}}{\text{Number of good user transactions}}$$

(3)

**4.2 Simulation Setup and Scenarios**

The simulation study demonstrates the behavioural reactions of the various trust and recommendation systems. Further, it presents the different aspects of malicious users (nodes) simulated under two malicious scenario types: i) the purely naïve scenario (a scenario in which, malicious nodes do not collaborate with each other, but rather maliciously operate by themselves), and ii) the purely collective scenario (scenario whereby malicious nodes collaborate among themselves to perpetrate attacks). While the former uses the global interaction data of user nodes as the basis for the computation of trust values (and hence, no other implicit data is considered or used for trust evaluation), the latter simulates a scenario where malicious peers share honest information amongst themselves.

**4.3 Simulations Results**

A summary of simulation results is presented under the *purely naïve* and *purely collective* scenarios.

**4.3.1 Effectiveness Metric for 50-Nodes**

Figure 2 presents a comparison of the effectiveness metric of EigenTrust and SecTrust systems in a 50 P2P node network, for the purely naïve and purely collective scenarios. Under the purely naïve scenario, SecTrust had a better trust effectiveness metric of 0.9028 in 0.32 seconds compared to EigenTrust (0.9016). However, EigenTrust had a slightly lower completion time (0.31 seconds) over SecTrust (0.32 seconds). Under the purely collective scenario, SecTrust had the best trust effectiveness (0.9008) in 1.29 seconds, while EigenTrust had 0.8983 in 1.33 seconds.
4.3.2 Effectiveness Metric for 100-Nodes
Like the effectiveness metric for 50 nodes, Figure 3 shows the simulation results of the purely naïve and collective effectiveness metrics of the trust algorithms for a 100-node network. For the purely naïve scenario, SecTrust showed better effectiveness (0.9439) and completion time (1.04 seconds). In addition, under the purely collective scenario, SecTrust still had better trust effectiveness performance (0.9426) over EigenTrust (0.9358).

Figure 2: Trust model effectiveness of a 50-node network in purely naïve and purely collective scenarios.

Figure 3: Trust model effectiveness of a 100-node network in purely naïve and purely collective scenarios.

4.3.3 Effectiveness Metric for 500-Nodes
Figure 4 presents the simulation results of the purely naïve and purely collective effectiveness metric of the trust algorithms for a 500-node network. In the purely naïve scenario, SecTrust gives a superior performance with respect to both the effectiveness metric (0.9506) and completion time (4.65 minutes). Under the purely collective scenario, SecTrust equally gives better effectiveness metric performance (0.9427) with a completion time of 2.46 hours.

Figure 4: Trust model effectiveness of a 500-node network in purely naïve and purely collective scenarios.

4.3.4 Effectiveness Metric for 1000-Nodes
Under the purely naïve scenario in the 1,000–node setting (Figure 5), SecTrust with a 0.9585 effectiveness value and a completion time of 23.59 minutes, showed better performance over
EigenTrust. The EigenTrust effectiveness metric of 0.9277 indicates that with the growth in network size, its effectiveness metric starts to decline. The difference in the effectiveness metric recorded between SecTrust and EigenTrust under the purely naïve scenario shows a 3.32% performance increase for SecTrust over EigenTrust. The result of the purely collective scenario is shown in Figure 5. SecTrust showed better trust effectiveness (0.9550) over EigenTrust, which recorded 0.9466.

**Figure 6:** Effectiveness of Trust models against Sybil attacks under purely naive setting.

**Figure 7:** Trust effectiveness against Sybil attacks under purely collective setting.

4.3.5 Trust Effectiveness against Sybil Attacks Under a Naïve Scenario

Figure 6 shows the relative performances of EigenTrust and SecTrust under the Sybil naïve scenario. Again, SecTrust maintained better trust effectiveness, consistent performance and scalability over EigenTrust across the 50, 100, 500 and 1000 nodes during the simulation.

4.3.6 Trust Effectiveness against Sybil Attacks Under a Collective Scenario

Under the Sybil collective scenario, Figure 7 provides the effectiveness metric of EigenTrust and SecTrust. Again, SecTrust showed better consistent performance and scalability over EigenTrust and EigenTrust Incremental across the 50 to 1000 nodes during the simulation.

5. Conclusions

This paper proposes SecTrust, a trust-based system for P2P networks based on the quality of the packet forwarding behaviour of peer nodes. Using the framework described in (West et. al., 2010), the trust-based effectiveness and performance testing of SecTrust against EigenTrust were compared. Simulation results under both scenarios (the purely naïve, and purely collective) shows that performance wise, SecTrust proved to be superior to EigenTrust. Based on the simulation findings we conclude that SecTrust has shown promising effectiveness and efficiency in trust management even in the presence of adversaries as presented in this paper. In our future work, we hope to present threat models to demonstrate the robustness of SecTrust against attacks in comparison to other trust and recommendation systems.

**References**


