Preprint: Jim Parker, Anand Patwardhan, Filip Perich, Anupam Joshi and Tim Finin, Trust in Pervasive Computing, in The Handbook of Mobile Middleware, P. Bellavista and A. Corradi (eds.), pp. 473-496, CRC Press, (ISBN 0-849-33833-6), May 2006.

Trust in Pervasive Computing

Jim Parker, Anand Patwardhan, Filip Perich, Anupam Joshi, and Tim Finin
CSEE Department
University of Maryland Baltimore County
1000 Hilltop Circle,
Baltimore, MD 21250

Abstract

Pervasive environments are comprised of resource-constrained mobile devices "limited" in their connectivity to other devices or networks due to the inherent dynamic nature of the environment. Limited connectivity to the Internet precludes the use of conventional security mechanisms like Certifying Authorities and other forms of server-centric authentication. Under these circumstances peer-to-peer interactions are well-suited for information interchange. However, practical solutions for protecting mobile devices, preserving privacy, evaluating trust, and determining reliability and accuracy of peer-provided data in such interactions are still lacking. Our research is directed towards providing stronger assurances of reliability and trustworthiness of information and services with practical implementation considerations for pervasive environments.

Contents

1 Introduction

The idea of ad hoc networking and pervasive environments is now more than a decade old. A significant amount of research on trust and privacy has been accomplished in the area of Social Sciences; however, since ad hoc networks have thus far not been popularly adopted in commercial products there has been little application research on trust and privacy in this area. Recent advances in wireless and storage technology, and the consequent proliferation of highly capable portable devices and wireless appliances is expected to lead to widespread use of ad hoc networking technologies. Even so, practical solutions for achieving security, privacy, and trust are still lacking. The highly invasive nature of some of these technologies pose a threat to the security and privacy of personal data and the area of pervasive computing.

Mobile devices with small form factors, yet with computing power comparable to desktops only years old, are now common. Enhanced multi-modal user interfaces like touch screens, biometric security devices, and accelerometers have significantly improved the usability of these devices. Integration of GPS receivers, cameras, recorders in cell phones and PDAs have ushered in a new generation of *converged* mobile devices. We are now witnessing a continuous proliferation of wireless appliances in everyday life – crib monitors, home security alarms, fire alarm annunciators, surveillance cameras, and automobiles. These technological advances are helping create resource rich environments in which personal mobile devices can seamlessly integrate – utilize and provide services. Moreover these mobile devices will be capable of sharing their capabilities via wireless means. Peer-to-peer relationships will enable devices to dynamically form collaborative relationships and perform complex tasks leveraging available resources either shared among the peers or those present in their surrounding environment.

Thus far, wireless networking has primarily served to extend the reach of the Internet. Most of the prevalent wireless technologies and their applications are infrastructure based. In traditional mobile computing environments, devices mostly adhere to the basic client-server model in which the devices act as clients and access non-transient information on trusted servers. In the client-server model, the server is anchored and a client can verify through several authentication and integrity schemes that the information originated from the server, forcing accountability. Mobile devices lack a "common sense" that people often employ to decide the reliability of both the source and information provided by the source. Consequently, devices need a mechanism to evaluate the integrity of their peers and the accuracy of information provided by their peers, as there is otherwise no scheme for protecting a device from malicious peers that deliberately provide unreliable information.

A Mobile Ad hoc Network (MANET) is a self-organized collection of wireless mobile nodes lacking a fixed network infrastructure and having no central authority. The flexibility and openness of MANETs make them very appealing as an information gathering and exchange medium; however, these two properties can also lead to security vulnerabilities. To fully realize the potential of the mobile ad hoc paradigm there must be an autonomous approach to mitigating risk and/or place users in control of risk evaluation and usage. Along

with enabling devices to estimate their trust in other devices and the accuracy in the information obtained from them, a mechanism must be provided that enables devices to detect and distinguish between *malicious peers*, that purposely provide incorrect information, *ignorant peers*, that are unable to guarantee a reliable level of provided information, and *uncooperative peers*, that have reliable information but refuse to make it available to other devices. This mechanism would also implicitly support an *incentive model*, in which all devices must provide only reliable information and provide this information often, otherwise they risk losing the ability to communicate with other devices in the environment.

In MANETs, a server-centric mechanism of identification and authentication is not suitable. Even with limited Internet connectivity, total reliance on conventional security mechanisms involving Key Distribution Centers (KDCs), Certificate Authorities (CAs) or similar forms of remote *trusted* sources impose serious limitations on the functioning of these devices, in effect limiting them to function only when those remotes sources can be contacted. In pervasive environments, the number of devices either embedded in the surrounding infrastructure, or personal mobile devices, will be immense. Thus, it is not possible to pre-enumerate all possible devices that may be encountered, nor will it be feasible to centrally register all such devices and then later identify and authenticate them on every encounter.

This chapter presents research work that addresses some of the concerns in protecting the privacy and security of mobile devices. The inherent vulnerabilities of pervasive networks have thus far restricted their use. Providing strong assurances of reliability and trustworthiness of information and services with practical implementation considerations for pervasive environments will be the most significant contribution of our research and will be another step toward making the vision of *anytime-anywhere* computing a viable reality.

2 Social Communities in Pervasive Networks

2.1 Pervasive trust

Using locally available information, collected from the surrounding pervasive environment or peers in the vicinity, introduces several trust and security issues. Due to the inherent nature of pervasive environments, conventional mechanisms of providing security are not suitable. Devices must be made self-reliant to make trust evaluations and use reputations to guide their behavior.

However, since mobile devices are potentially innumerable, it is not possible to be able to cache the identities and reputations of all the encountered devices, nor can we expect all devices to be cooperative. However the abundant storage capacities of the mobile devices will be sufficient to cache specific device identities, e.g., it will be sufficient to remember only devices that are of future potential value in forming social networks and those that will be most likely to cooperate.

Scenario: *Peter is flying a red-eye from LA to NY. His calendar has a meeting in his NY office at 9:00 am. His portable device notices other people present in the airplane and finds his colleagues Clark and Lex, who are also attending the same meeting, though Peter can't see them from his seat. Each of their devices interfaces with the personal display screen in front of them, and the built-in cameras in their devices allow them have a live video conversation, exchange notes for the next day's presentation – in-flight. They later decide to watch Peter's presentation, to provide feedback. Peter grants them the right to access his device and make changes to the presentation. Neither of Peter, Clark or Lex's devices have the capability to edit video content, however Clark finds one of his old friends Bruce also in the airplane whose device has the required capability. On Clark's request and recommendation, Bruce allows Peter to use the video-editing capability. Peter is able to improve and finish his presentation without even leaving his seat.*

In the above scenario, neither of the participants were initially aware of each others presence. Authenticating each other's devices is usually not possible unless prior security associations exist. Pre-enumerating all possible devices that can be encountered is not a feasible option. Thus, distributed trust management becomes a necessity for the survival of the network. In the following sections we propose a distributed trust management scheme that utilizes activity monitoring and reputation management to evaluate trust. We propose to employ mobility patterns and distinguishing landmarks/beacons to evolve trust and establish a scalable pervasive reputation management framework. Reputations of known devices in addition to activity monitoring can be used to compute trust in that device.

The networking layers can benefit from knowing who the reliable or trusted peers are within the local neighborhood for preferential consideration in forming routes and for peer discovery. The application layer can benefit from reports of malicious activity detected by the lower layers and appropriately modify their trust assessments. Further recommendations by trusted devices can be then used to create new trust relationships or modify existing ones.

Connectivity provided by ad hoc networking entails that peers in the pervasive environment be cooperative. Due to the security threats posed to individual mobile devices, collaborative efforts in countering intrusive behavior are required. Most of the response mechanisms we have described in [20] are reactionary. Since scopes of intrusion detection mechanisms deployed on individual devices are limited to their radio-range, collaborative mechanisms are required for communicating suspicious activity and intrusions to other devices in the vicinity. We propose to use *reputations* to pro-actively detect and deny resources to devices that have been deemed malicious. Also while sharing information and services amongst devices, complex processes of trust evolution can be simplified using recommendations amongst trusted peers – which are again motivating factors for forming local collaborative groups. We propose the concept of ***Pack Formation*** that uses accounts of prior encounters, evolved trust, and recommendations, to form local packs.

Using context information and notions of neighborhoods that can be identified by specific unique landmarks, devices need only store trust information pertaining to the relevant context, e.g., a portable device owned by a student should only remember the most frequently encountered devices in the vicinity of the University campus to deduce that those particular entities are frequent visitors of that neighborhood/community. Furthermore, if malicious activity is attributed to any such known entity, this fact can be reported back in the community where that entity is known to be a frequent visitor.

Without assurances of the reliability and trustworthiness of retrieved data, the utility and effectiveness of the completed tasks is questionable. Metrics to evaluate reliability of data and trustworthiness for peer-provided information must be available. Further trust evaluation and reputation management mechanisms will allow devices to function autonomously with minimal user intervention. To achieve these goals it is necessary to have a holistic approach in addressing issues of device security, secure routing, peer discovery, data management, and trust relationships – since these issues are highly interdependent.

We propose giving MANET nodes the ability to independently evaluate *Trust* in the nodes with which they interact. This solution involves a reputation management system through which nodes can evaluate, maintain, and distribute information about trust relationships within a MANET. Each node can make autonomous decisions about the trustworthiness of other nodes, providing an alternative to third party authentication during periods of disconnection. Ding *et al.* [13] propose using two kinds of trust, viz., "domain trust" and "referral trust." Nodes can ask other trustworthy nodes for providing information (domain trust) or trust them to provide referrals to other devices who might have that information (referral trust).

Since MANETs rely on cooperation from all nodes, detection, and isolation of malicious nodes is a must for a MANET to function. Malicious and non-cooperative nodes can cause disruption in MANETs and potentially disable the network. Each node must be able to identify malicious activity since centralized Intrusion Detection (ID) schemes and firewalls cannot be effective in a MANET environment [19, 20]. Also at the application level, devices should be able to make autonomous assessments (reliable, corrupt or unknown) about data provided by peers. For trust management at the application level we present results from our work in distributed reputation management and accuracy beliefs in section 3. In section 4 we describe some activity monitoring techniques that we use to detect intrusive/malicious behavior at the lower networking levels.

2.2 Services to-go

Continuous improvements in compact storage technologies including semiconductor memory – CompactFlash, MMC cards etc., and miniature harddisks and microdrives have spawned a generation of mobile devices with substantial storage capacities. Abundant on-board storage relieves the burden of requesting services or data from remote servers thereby freeing devices from the dependency on connectivity to remote servers. Devices guided by their profiles [21, 11] can cache large amounts of potentially useful information and keep required information updated by asking other trusted devices in the vicinity and require connectivity to the Internet only when absolutely necessary. To be able to guide themselves, the devices will need to sense their contexts (e.g., spatial and temporal). Reading local information from reliable sources, the devices can compose locally available services and use their existing knowledge bases to service their needs – be largely self-reliant. Moreover, all such devices will be capable of providing useful services to other (mobile) devices in their vicinity. The collective resources comprised of the the individual data storage capacities and unique sensory and effector capabilities and the individual trust relationships will enable complex tasks to be performed and improve the overall performance of collective and individual tasks. Long range wireless services are often not suitable for high data rates and at times are not cost-effective. We propose to harness the immense storage capacity of the mobile devices, optimize use of available connectivity to merely keep the knowledge base updated, and enable

devices to function autonomously.

2.3 Pack formation and collaborative queries

As exemplified in the above scenario, mobile devices are often bound by commonalities in the physical world. Common goals can be deduced from the profiles of the users and their devices. Thus there exist natural incentives to collaborate. The pack formation mechanism that we have proposed has several advantages – faster response times, increased scope of search, distributed trust and reputation management. Also collaborative mechanisms will prove useful when collective action against colluding adversaries is needed. We present some of our preliminary results from our simulations.

Collaboration in query processing leads to improved response times. We simulated an environment with 50 nodes spread in random locations in a two dimensional square area using GlomoSim [28]. We present some of the interesting performance results from two separate sets of simulations. In the first case, each device assigned a task set of distinct questions, individually searches for answers. Later, the same set of devices with the same task set of questions search for the answers collaboratively.

For simplicity, we assumed that some initial trust already exists to be able to form collaborative groups. We present results with pack sizes of 5 in a total population of 50 devices in a 150 sq. m. area. Each device had a transmission range of 25 m. and follows a random waypoint model (speeds varying from 1 to 5 m/s and pauses of 5 seconds). Each device tried to find answers for its assigned task set of 100 questions and the answers were randomly distributed amongst the remaining nodes.

To simulate the serendipitous nature of the environment we varied the percentage of the knowledge base present in the neighborhood from 40% to 100% in increments of 20 percentage points. We ran the simulation using five different starting positions for the devices, for five runs of the simulation.

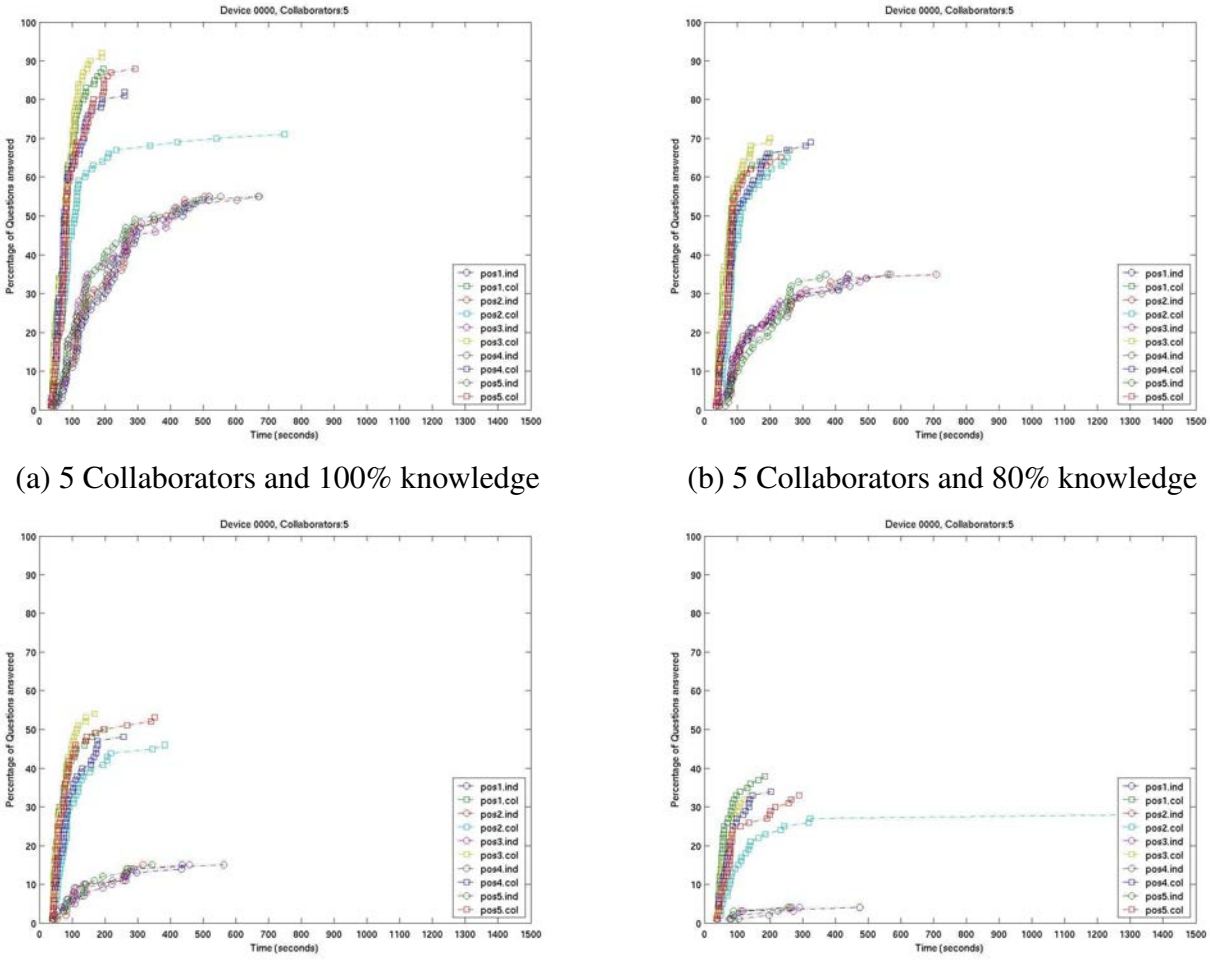
Since our focus was on the effectiveness and response time of the search, we assumed that all the sources of information were reliable and would only provide accurate answers. In the collaborative version, pack members help each other find answers to their questions. When an answer for a collaborator's question is found, the device tries to send it back to that collaborator.

The plots in Fig. 1 depict two sets of trends each, representing the 5 different starting positions of the 5 collaborators, each having a task set of 100 questions (not common with other collaborators). Also, the devices themselves cannot answer their own questions. Fig. 1 (a), (b), (c), and (d) have decreasing knowledge bases 100% to 40% in decrements of 20 percentage points. The collaborative version consistently outperforms the individual searches. In Fig. 1 (a), the collaborative version is able to find twice as many answers in under a minute since the start of the querying process. In the non-collaborative version where devices independently try to query other devices in their radio-range, they manage to find approximately 50% of the answers and took upto 10 mins.

In Fig. 1 (d), the performance difference is more pronounced where there is now only 40% knowledge in the vicinity, i.e., answers to only 40% of the questions are available. Here it is seen that the non-collaborative version was able to find no more than 5% of the answers, whereas the collaborating devices managed to find as many as 30% of the answers in less than 4 minutes.

All the simulations showed promising results in terms of faster responses and search

effectiveness, in case of the collaborative models. We observed that as the pack size was increased from 5 to 10, the control overhead for communication between the pack members increased and introduced minor increase in latency to query responses, yet the number of successfully answered queries were consistently more than the non-collaborative version.



(a) 5 Collaborators and 100% knowledge

(b) 5 Collaborators and 80% knowledge

(c) 5 Collaborators and 60% knowledge

(d) 5 Collaborators and 40% knowledge

Figure 1: **Preliminary results from simulations with 5 collaborators**

3 Belief and Reputation in MANETs

This section introduces a *distributed reputation model*, which extends the traditional query processing model [22] in order to allow devices to capture their beliefs on reputation of their peers and accuracy beliefs of information obtained from those peers. To mitigate negative effects of malicious and "ill-informed" devices, the model categorizes peers as *reliable* and *unreliable*. In the model, the accuracy of an answer is a function of the trustworthiness of the information source and its belief in the accuracy of its answer. Devices assign trust to an information source based upon past experience and from the recommendations of those devices that it trusts.

3.1 Related work

Trust and belief management models can be divided into two categories: mathematical and logical. Jonker *et al.* [15] propose a mathematical model for capturing trust in multi-agent systems. Their model consists of beliefs, and trust is a function of the values of these beliefs. The trust function is based on initial trust, experiences, and trust dynamics. The types of trust dynamics determine how past experiences affect the newly computed trust value. Richardson *et al.* [25] present a mechanism for calculating the trustworthiness of users on the Semantic Web by developing a "web of trust" based on web algorithms like Google's PageRank [1]. In this approach, every user maintains trust values for a small set of users and uses the belief values of these users and her trust in the users to calculate her own beliefs. Abdul-Rahman *et al.* [2] define a formal trust model based on trust and recommendations. Users store trust values for other users and ask trusted users for recommendations when dealing with unknown users. However, once a trust value is calculated it is not updated.

There is also a significant amount of work on developing logical trust models. Blaze *et al.* [6, 7] define trust management as creating policies and assigning credentials. They use a PolicyMaker engine for checking if users' credentials conform to policies before granting them access. Keynote [5] is designed along the same lines as PolicyMaker, however, it has been designed to be simpler, to provide more support for PKI, and to allow policies and credentials to be transported over insecure communication channels. Referee is a similar trust management system that is designed to facilitate security decisions for the web [12]. Kagal *et al.* [16, 17] also describe a policy based infrastructure for security and trust management in multi-agent systems and the Semantic Web. In this system, every entity has a policy that reflects its current binary trust values and exchanges them with other entities via speech acts.

The model described in this section employs the mathematical approach. This is because a mathematical model requires fewer computing resources than logical models, which require reasoning engines and a certificate verification. Additionally, unlike logical models that only describe conditions when devices are "trusted" enough to access a certain information, mathematical models can also be employed to represent answer accuracy and to handle situations when more than one answer is provided for a given query. This model differs from other mathematical models in that the new model proposes several trust learning schemes based on experience and recommendations, allow information sources to specify their trust in the information being provided, and use both kinds of values to compute belief. Most other schemes either provide trust learning algorithms based on experience or on recommendations but do not combine the two. They also ignore the believed accuracy of the information source, whereas the proposed model uses it as a factor for rating the trustworthiness of a source.

3.2 Reputation model

A successful model evaluating integrity and information accuracy of a device must address the inherent limitations of mobile ad hoc networks and of mobile devices, including power, memory, and computation constraints as well as network reachability and wired infrastructure support limitations. The reputation model described in this section overcomes these issues because it does not rely on any wired infrastructure nor does it assume connectivity among all devices. The model also does not assume that each device can maintain belief information about every other device or information the other devices can provide.

The model only assumes that every device is able to assign an accuracy degree to any information the device provides to its peers and that every maintains trust degrees about a

subset of devices in the environment representing how much a device trusts the other devices for providing accurate answers to queries. The accuracy degree represents the device's belief about the correctness of the information, which can range from *distrust* to *undecided* to *trust* value. A device, when asked, can provide a its recommendation for some other device in question. Similar to accuracy degree, the recommendation can range from *distrust* to *undecided* to *trust* value.

The model functions as an extension to a traditional query processing model for mobile ad hoc networks. In this reputation-driven model, a querying device collects responses from peers but also computes their trust degrees. It has been advocated [23] that this approach is superior to the alternative where a device first computes reputation of its peers and then queries those peers for information.

3.2.1 Information Source Discovery

When a device needs to obtain an answer for a query, it first attempts to discover which of its peers may have the necessary answer. The device does so by evaluating its cache of advertisements received from its peers and by broadcasting a source discovery request messages to its peers up to n-hop away. The discovery message consists of the device's identity ID_Q, the question Q, and a nonce for differentiating it from other discovery messages sent by this device. A device may sent out the discovery messages more than once based on the responses it receives from its peers.

3.2.2 Information Advertisement

When a cooperating, non-malicious peer receives a source discovery, it checks its cache to find an answer matching the question. If the peer has a cached answer, it will respond by sending an advertisement message containing the identifier of the device ID_S and the question it can answer Q, where the ID is some globally unique string (e.g. the MAC address), or a cryptographically secure scheme that prevents id spoofing such as those presented by Gligor *et al.* [18]. A device may optionally pro-actively broadcast bulk advertisements at random intervals.

3.2.3 Querying Peers

The querying device evaluates all advertisements in its cache in order to determine possible sources for its query. If a device is unable to discover a sufficient number of information sources that could provide answer to its question, the device simply broadcasts the question to all peers in its vicinity, again up to n hops away. If, however, the device is able to collect some information sources, the device sends a query to only those peers.

3.2.4 Collecting Answers

When a cooperating, non-malicious peer receives a query message, and has a matching answer, it will respond with a message containing its ID, the answer A_i, and the accuracy degree $A_D(i)$ of the answer from -1 to 1.

3.2.5 Recommendation Request

Each querying device Q has a lower limit n on the number of trusted peers that must provide an answer to a given query. A trusted peer R is any peer for which the device Q has a *trust degree* $T_Q(R)$ above a certain trust threshold τ. While a device has not received enough answers from at least that many trusted peers, it computes the trust degree of every peer ID_R that sends it an answer, using its initial trust belief function α_D and current trust values. If the device is unable to determine if the answering peer is *trusted*, and it has not reach the minimum number of trusted responses, the device may initiate a recommendation session about the answering peer.

In the model, the device can either ask only those devices who it believes are its *trusted peers* or the device can ask anyone in n-hop distance for recommendations about the answering device. The querying device ID_Q does so by sending out a recommendation request message to some remote peer ID_R with the identity of the answering peer:

3.2.6 Recommendation Response

When a cooperating, non-malicious peer receives a recommendation request, it looks up its trust beliefs to determine if the querying device ID_Q is one of its *trusted peers*. If this is the case, the device responds with its trust degree in ID_S by sending a recommendation response message.

3.2.7 Calculating Final Answer

Once a device receives all responses from all peers to whom it sent its query message or once its session timeout period ends, the querying device proceeds by calculating the final answer.

For every different answer value it has received, the device calculates the combined accuracy degree of the particular value based on the suggested accuracy degree of the information sources and their trust degrees using its *trust-weighting* and *accuracy-merging* functions $\otimes$ and $\bigoplus$, resp.

$$\boxed{\forall answer_i : A_{combined}(i) = \bigoplus_S \Big(A_D(i) \otimes T_Q(D)\Big)} \tag{1}$$

The model defines three accuracy-merging functions - AFV, MIN, and MAX. These functions are similar to computing membership degrees in boolean combinations of fuzzy variables.

The querying device uses the merging function to compute a combined accuracy degree for every distinct value it received as a possible answer.

$$\boxed{PA = \Big\{(answer_i, A_{combined}(i))\Big\}} \tag{2}$$

If all trusted devices provide the same answer to the original query, i.e., there is only one tuple in PA, the querying device will simply use that answer as the final value if its combined accuracy degree is above a certain threshold τ. This similar to the threshold concept for a *trusted peer*.

In many cases, however, the querying device may receive multiple conflicting answers from trusted peers. To address this problem, in this model the querying device can apply two different techniques:

The querying device may accept an answer only if precisely one of the suggested answer has a combined accuracy level above τ. This technique is referred to as *only-one* answer (OO). Formally:

$$\boxed{\begin{array}{l} answer_x \text{ is } OO \Leftrightarrow \\ \forall(answer_i, A_{combined}(i)) \in PA : \begin{cases} A_{combined}(i) > \tau & i = x \\ A_{combined}(i) \leq \tau & i \neq x \end{cases} \end{array}} \quad (3)$$

This is a pessimistic approach as the device chooses precisely one answer, e.g. when the device is querying for the current stock price of a certain company. There is only one possible answer. The querying device will not cache an answer, if there is more than one answer above the threshold or if there are no answers above the threshold. At the same time, this approach will limit the amount of uncertain or distrusted data kept in the cache.

Alternatively, a device may employ a more optimistic technique that considers the possibility of a question having multiple valid answers, e.g. the list of Chinese restaurants in a given location. In this case, the device will choose the answer with the highest combined accuracy degree above τ. If there are multiple answers with the highest accuracy degree, the device will randomly choose one. This is called the **highest-one** answer (HO). Formally:

$$\boxed{\begin{array}{l} answer_x \text{ is } HO \Leftrightarrow \exists max : \\ \Big(max > \tau\Big) \wedge \Big((answer_x, max) \in PA\Big) \wedge \\ \Big(\forall(answer_i, A_{combined}(i)) \in PA : A_{combined}(i) \leq max\Big) \end{array}} \quad (4)$$

3.2.8 Updating Trust Belief

If a querying device is able to determine its final answer Λ, it uses that fact to evaluate the interaction experiences it had with the answering peers. A querying device has a *positive experience* with an answering peer, if the answering peer provided the same answer as the final one and if the answering peer suggested a non-negative accuracy degree for the answer. A querying device has a *negative experience* with an answering peer, if the answering peer provided a different answer to the query and suggested a positive accuracy degree. The querying device also has a negative experience if the querying peer provided the same answer but suggested negative accuracy degree. Lastly, for all other cases, the querying devices has an undecided experience.

When a querying device has either a positive or negative experience with any answering device, it should update its trust degree for that device for future interactions by using one of the available trust learning functions Δ, which were adapted from [15].

The first category of trust learning functions employs all history information to predict a future trust degree of a device. In one technique referred to as *blindly-positive*, a device with absolutely trust its peer if it had at least n positive experiences. Similarly, one can define a *blindly-negative* approach.

The other category employs only the current experience of the querying devices with the answering peer and the current trust degree the querying devices has of its peer to calculate the future trust degree. The querying device does so by either increasing or decreasing the trust degree using slow, fast or exponential steps. The model uses *fast-positive/slow-negative*, *slow-positive/fast-negative*, *balanced-slow*, *balanced-fast* and exponential techniques. The first four techniques use a fixed fast or slow step to increase or decrease a trust degree. The

last technique always increases by a half step necessary to move from current trust degree to an absolute trust degree of 1.0 and similarly to decrease to an absolute distrust at -1.0.

The update of the querying device will affect who the device sends its query in the future as well as how the device recommends the answering devices to its peers.

3.2.9 Answering Peers

For an honest device to return an answer, provided it has an answer, the device must be able to determine whether the querying device is one of its *trusted peers*. The answering device evaluates its beliefs to determine if its local trust degree of the querying device is above the trust threshold τ. Otherwise, the device first initiates a recommendation session by either asking all of its n-hop peers for recommendations about the querying device or by selectively asking devices it determines using its local belief to be *trusted peers*.

Given a set R of recommendation tuples (Q, D, $T_D(Q)$), where Q is the querying source, D is the recommending peer and $T_D(Q)$ is the recommended trust degree representing how honest Q is according to D, then the combined recommendation trust degree is computed by the answering device A as:

$$\boxed{R_{combined}(Q) = \bigoplus_R \Big(T_D(Q) \otimes T_A(Q)\Big)} \tag{5}$$

Similar to combining the weighted answer accuracy degree values, for combining recommended trust degrees here the AVG, MIN and MAX methods can be employed.

The device then sends back an answer to the querying device when $T_A(Q) > \tau$ or $R_{combined}(Q) > \tau$.

This mechanism implicitly creates an *incentive model*, in which it is the interest of every device to provide only reliable information and provide this information often because others maintain and share reputation degrees about this device.

The model computes trust and accuracy degrees using a two-level deep path algebra only. Therefore, in the model a mobile device calculates a trust degree of another device using only its trust degrees of its peers and their proposed trust of the other device. Similarly, a device computes an accuracy degree using only its combined trust degree of answering devices and the suggested accuracy values obtained from these answering devices.

The advantage of using only a limited depth is straightforward. Since each devices combines only up to two trusts values where one value is its own and one is of a remote device, one cannot introduce a possible cycle in the path computation. Secondly, since each device answer recommendation requests by evaluating only its local trust degrees, this model does not generate an additional traffic in the resource limited wireless ad hoc network. Last, it is a relatively easy exercise for a mobile device to automatically updates its beliefs about cached trust degrees and answer accuracy degrees when it changes a trust degree of any of its peers. Therefore, the model uses only a two-level deep path algebra.

4 Malicious Activity Detection and Trust

Mobile wireless networks can be divided into two architectures: Infrastructure and Ad hoc. In the infrastructure case, each network has a central node, or access point, through which all traffic must pass. The access points act not only to route traffic between nodes, but grant or deny access to the network based on policies or access lists. Wireless intrusion detection

devices have been developed to a limited extent for wireless traffic monitoring in infrastructure networks. Since all wireless traffic must transit a central node, the wireless intrusion device can be co-located the the access point and scan wireless channels looking for malicious activity. Devices trying to gain unauthorized access, and/or disrupt network activity, can be easily detected.

True MANETs have no central access point. Nodes connect together in and ad hoc fashion to form a mesh network for information routing. Single intrusion detection systems cannot effectively operate in this environment, since the dynamic nature of the network makes a central observation point very unlikely. For the remainder of the discussion, we will assume MANETs are being used for node connections

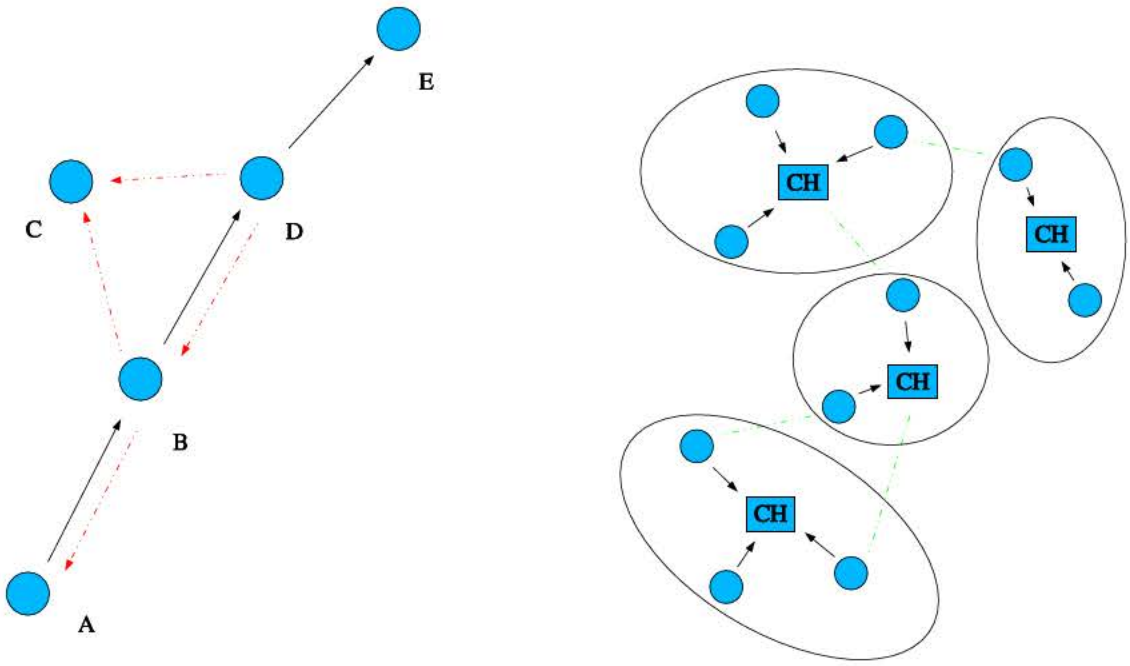
Current wired security mechanisms require either a third party authority for authentication, or *a priori* distribution of key material. MANETs offer no guarantee of a constant Internet connection and therefore third party authentication may not be available. Proposed solutions to this problem for MANETs involve the establishment of a security association (SA) either out of band or with a' priori knowledge of an encounter on which additional secure protocols are enabled [8], [3], [10], [27]. The problem is that SAs cannot be randomly established between two nodes that are previously unknown to one another in an Internet disconnected scenario. In order for MANETs to become widely accepted and used, there must be some mechanism giving confidence to users that security exists within the MANET. We believe the way for security to be established in MANETs is in the evaluation and fostering of *trust* between interacting nodes. Failure to provide this mechanism could have a negative effect on security, quality of service (QoS), and overall willingness to risk information exchange over MANETs.

4.1 Malicious activity detection

Our research into malicious node detection is based on promiscuous snooping of the network channel. Snooping leverages two properties inherent in most mobile ad hoc protocols. The first property is that each node in the network maintains a neighbor list containing the addresses of those nodes with which it is in immediate proximity or on the path from a source to a destination. The second property, as is the case in the 802.11 [14] and *MACAW* [4] link layer protocols, is that a node is able to "hear" the *RTS/CTS* negotiation of its neighbors. Accordingly, each node participates in the intrusion detection process and *"snoops"* on its neighbor's transmissions in order to ensure that they network packets have not been modified or mis-routed. The notion of "snooping" is also employed in *DSR*, which is used for "reflecting shorter routes" as an optimization of the route maintenance process.

Our initial research involved an extension, which is viable for many ad hoc routing protocols (e.g. DSR, AODV), where the snooping nodes listen to all other nodes in their proximity. In Fig. 2 promiscuous snooping is shown in an example where node *A* is sending traffic to node *E* via nodes *B* and *D*. In this example node *C* can monitor the traffic as it transits from node *B* to node *D* and then forwarded to node *E*. Node *C* is in the position to determine whether node *D* changes, re-routes or drops any packets in the data stream.

We drew a distinction between our work and others *Watchdog* [26] and *Neighborhood Watch* [9], which work primarily with *DSR*, watching the forward node on the patch from source to destination. We experimented with two response mechanisms when observing nodes recognized ongoing malicious activity. The first was a *passive* response mode, where a node, upon determining that another node is aberrant, unilaterally decides to cease interacting



(a) Snooping (b) Clustering

Figure 2: **Malicious Node Detection**

with the offending node. Although each node acts independently, eventually the intrusive node will be blocked from using all network resources.

The second response was *active*, where each node relied upon a *Cluster Based* hierarchy (Fig. 2). When a node detects an aberrant neighbor it informs its *Cluster Head*, who in turn initiates a voting procedure. If a voting majority determines that the suspect node is in fact intrusive, an alert will be broadcast throughout the network and the intrusive node will be denied network resources. The algorithms were written and simulated using GlomoSim version 2.03. The simulation environment was as follows:

1. Number of Nodes: 50 (16 nodes involved in constant bit rate (CBR) connections, and the number of malicious nodes was varied).
2. Grid Size: 2,000 by 2,000 meters.
3. Application Traffic: 10 CBR connections are generated simultaneously, where 4 nodes are source for two streams each, and 2 nodes were the source for a single stream each; destination nodes only receive one CBR stream.
4. Mobility: Random Way-point Model (max speed 20 meters per sec., pause time 15 sec.).
5. Radio: "no fading" radio model, with range of 376 meters.
6. MAC Layer: 802.11, peer-to-peer mode.
7. Routing Protocol: AODV or DSR.
8. Simulation Time: 200 sec.
9. Neighbor `Hello` period: 30 seconds.
10. Dropped Packet Time Out: 10 sec.
11. Dropped Packet Threshold: 10 packets.
12. Clear Delay (event expiration timer): 100 sec. (e.g.: the amount of time that a node considers an event without coming to a final determination).
13. Misroute Threshold: 5 events. Detectable only in routing protocols using Source Routes such as DSR.
14. Modification Threshold: 5 events.

Results were obtained by averaging 100 simulation run for 200 seconds each [19, 20]. The True positives, False positives, and successfully delivered packets as a percentage of the number of bad nodes in the network for DSR and AODV were measured (Fig. 3).

Node density of both malicious and normal nodes was a significant factor in rates of True positives. For a malicious node to be detected, it must act maliciously within the proximity to a good node. As expected, the performance of the both the Passive and Active response protocols improved, with respect to both True Positives and False positives, as the density of the malicious nodes increased. Likewise, and as expected, the number of successfully delivered packets decreased as the density of malicious nodes increased. This is attributable to the increased bandwidth requirements for the voting mechanism. It became apparent through the experiments, that the voting scheme worked better in a more dense environment of network nodes as opposed to a sparse environment.

During development, there were a number of issues raised for further work. The first issue raised was security/privacy concerns over nodes snooping network traffic. Even though

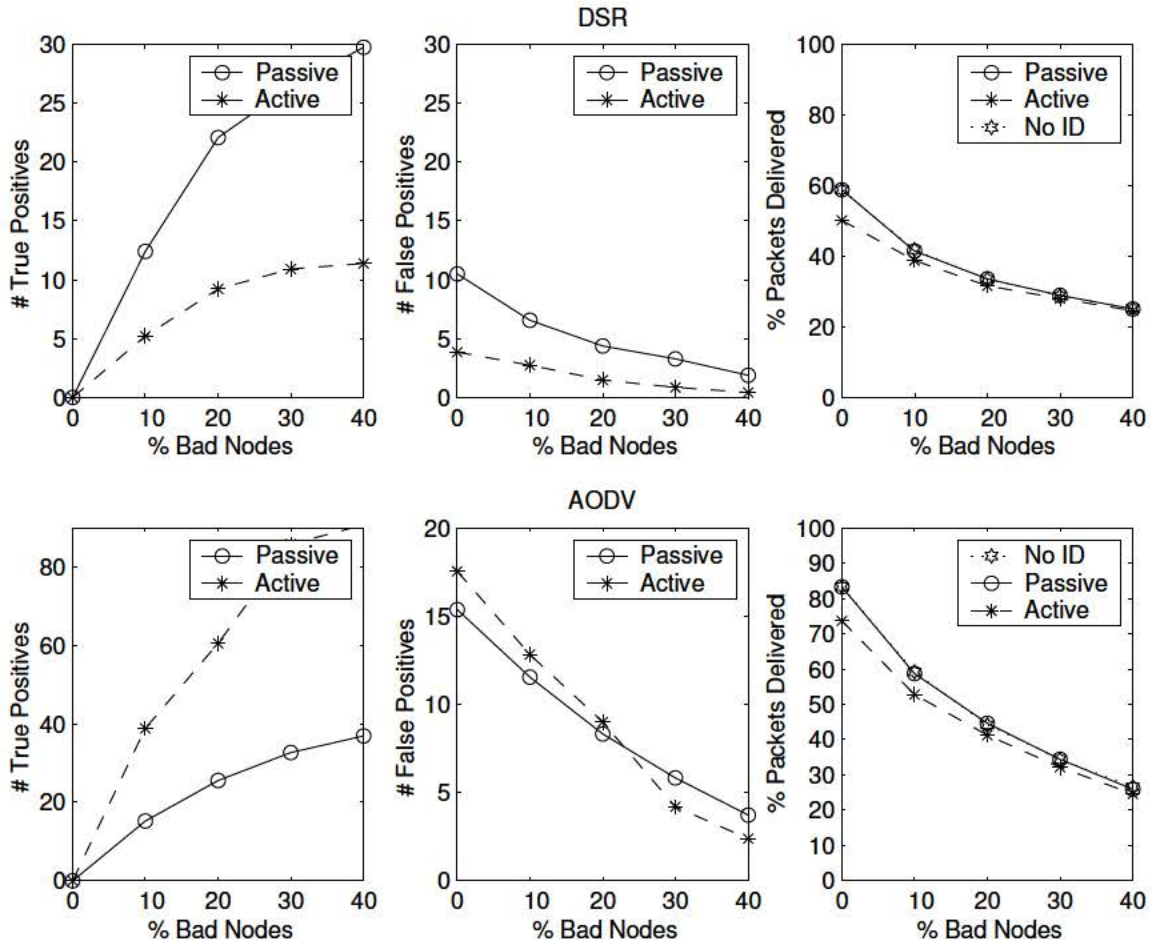


Figure 3: **Simulation Results of the Active and Passive Response Protocols for DSR (top) and AODV (bottom)**

this is always a possibility, users may tend to dislike the idea of their traffic being monitored by all other nodes in their vicinity. Another issue was performance while snooping and processing network traffic. Nodes such as PDAs that cannot do anything else while performing malicious node detection are of little use to their owners. Certainly there is a concern for malicious nodes raising alarms and causing problems such as denial of service attacks agains accused nodes. The active response with cluster voting helps make this more difficult. Finally, reliably detecting and separating malicious users from normal mobile node disconnection in a MANET is quite difficult. Thresholds for the various parameters of malicious detection depend on node density, average speed, etc. The idea of re-admittance into the network for falsely accused nodes is something that could be very important if snooping schemes are to be used.

4.2 Cross-layer information processing

Malicious node detection is an important step toward developing trust relationships between MANET nodes. Nodes found to be acting maliciously should definitely not be trusted. However, MANET nodes can also be malicious in other ways seen outside the network layer. We decided to start looking at observable events within other layers of the OSI stack.

Perich [24] developed a simulation for an application layer query processor in MANETs. Information is assumed to be probabilistically distributed throughout the MANET, and nodes then query other nodes for the information. Results are determined to be correct or incorrect in order to make trust evaluations about the replying nodes. After a period of time, each node maintains a table of nodes with whom it has interacted, along with trust ratings reflecting *reliable* or *unreliable*.

We believe that by combining the network layer malicious node detection, with the application layer query, a more robust approach to rating individual trust can be achieved. Network layer events are more numerous but less specific than query information gathered at the application layer. The ultimate goal is that by combining information from the two layers a more accurate representation of trust can be obtained. One of the results from [24] shows that evolving trust is heavily based on the initial trust rating. We will be providing the initial trust rating from observation information gathered at various layers of the OSI stack. Additionally, the information value and subsequent risk was the same for all nodes in [24]. We will be studying how trust may be evolved through first using low risk encounters with low value exchanges and gradually progressing to higher risk situations. The overall metric of success or failure would be a function of how much total value is lost over a period of time.

To expand on the original idea for querying we will be dividing nodes into classes. For instance when on vacation, there may be a willingness to trust information provided by a citizen of the current city, over that provided by another tourist. As another example there may be an unwillingness to trust information provided by someone having a vested interest in a sale, over an independent third party experienced opinion.

Throughout our research we are using a combination of simulation and testbed implementation to measure time for convergence of trust, accuracy in responses, malicious node detection time, and true/false positives for malicious node accusations. Results are to be be compared with test runs for malicious node detection and trust query application separately to determine the relative contribution of each in the combined effort. Evaluations will be conducted with scenarios for low, medium, and high mobility, and with varying degrees of malicious nodes..

The issue of resource management as it pertains to performance is very important. As part of our research we need to determine resource usage at each node in relation with scalability. This will actually involve the number of nodes in the MANET, the number of node encounters during some time period, and the number of new nodes entered into the MANET during interaction. All of these conditions affect the resources used at each node for the algorithms. Results will be compared with results of research being conducted at EPFL in the area of rumor spreading in MANETs.

We believe there to be additional information that can be used in the development of trust from other OSI layers. We plan on exploring MAC layer management and control messages (i.e. beacons, RTS, CTS, etc) for patterns indicating malicious use. Other approaches include exploring the physical layer to provide signature information associated with a particular node physical address. Depending on the granularity of attribute measurement, nodes may be able to match indicator attributes and alarm when malicious nodes try to spoof existing

trusted node addresses.

There is also room for modification of the query model to reflect certain real world actions. For instance, trusted nodes do make mistakes; trusted nodes may not be as accurate as a particular situation warrants. In the case of malicious nodes, they may not always act malicious but instead restrict malicious behavior to specific situations, or circumstances. We recognize the development of trust to be a difficult task and would like to examine how to best judge trust within MANETs given these scenarios.

5 Discussion

Trustworthy data management in pervasive environments is a challenging task. The trend of continuous improvements in the capabilities of embedded devices and their widespread acceptance are signs of the onset of a highly interconnected society where pervasive environments and portable devices form integral parts of our daily lives, creating cultures of their own, and unforeseen applications emerging. Due to the inherent open, dynamic and distributed nature of these environments, they cannot be secured by conventional security practices.

Our experience in trust management for pervasive computing has, thus far, only been in the lab setting. We have made attempts to anticipate user issues, as well as developer issues, in our simulations and experiments. By implementing algorithms on actual handheld devices, we have shown that such ideas can are viable with current technology, at least in a limited capacity. What we do not know, and can only speculate on, is how the devices will be used in a widespread pervasive network.

From our simulations and experiments we have determined that nodes must be expected to adhere to some pre-determined set of rules for each layer (physical through application) that govern acceptable behavior, and our capability to detect anomalies and attribute misbehavior to particular nodes. At the lower layers, the monitored data can be quite overwhelming to process directly. However such data can be filtered using the rules of acceptable behavior and provided to higher layers for further processing. Specifically such data acquired from across layers can be processed, and applied directly reputation management. By aggregating the data from multiple layers, application layer protocols can use monitored behavior from lower layers to report and react to overtly malicious acts. To protect the network from malicious or faulty devices, observed misbehavior, or non-compliance to protocols is used to penalize such nodes by denying them access to resources and/or excluding them from the network. In order to be successful, a majority of nodes must participate. Nodes failing to fully participate in information routing or message store and forward mechanisms, can lead to disruptions in the ad hoc network. Nodes failing to participate in monitoring for malicious activity can also leave the network vulnerable to attack.

One of the hard problems is trying to overcome false alarms. Our experiments have shown the importance of identifying one-hop neighbors in order to monitor traffic effectively. Incorrect neighbor table information can lead to false accusations, or ignoring actual malicious activity. An increase in “hello” packet rates can make neighbor tables more accurate over the short term, but also lead to increased traffic congestion. With congestion comes lower date throughput, and potential interfere with nodes monitoring for malicious activity.

We believe social communities will play an important role in reputation management and trust evolution. Our experiments with pack formation and results from preliminary work validates our approach. We have experimented with a distributed reputation management system that uses a mathematical model, enabling individual devices to compute the accuracy of peer-provided information. These concepts can be used to reliably determine and manage reputation and trust in a pervasive environment.

We recognize that trust management is not only a technical issue, but also one of social acceptance. Our experience has led us to believe that even though it may be technically possible to overcome the challenges of trust and reputation management in pervasive computing, success or failure will be determined by the willingness of people to purchase and use these devices. To fully realize the potential of the mobile ad hoc paradigm there must be an autonomous approach to mitigating risk and/or place users in control of risk evaluation and usage. We have presented techniques ranging from the application layer to the lower level networking layers that help mitigate risks stemming from the open, dynamic nature of pervasive environments; however there are still many open challenges to overcome.

References

[1] http://www.google.com/technology/.

[2] ABDUL-RAHMAN, A., AND HAILES, S. A Distributed Trust Model. In *Workshop on New Security Paradigms* (2003).

[3] BALFANZ, D., SMETTERS, D., STEWART, P., AND WONG, H. Talking to strangers: Authentication in adhoc wireless networks, Feb. 2002. In Symposium on Network and Distributed Systems Security (NDSS '02), San Diego, California.

[4] BHARGHAVAN, V., DEMERS, A., SHENKER, S., AND ZHANG, L. Macaw: a media access protocol for wireless lan's. In *Proceedings of the conference on Communications architectures, protocols and applications* (1994), ACM Press, pp. 212–225.

[5] BLAZE, M., FEIGENBAUM, J., IOANNIDIS, J., AND KEROMYTIS, A. The KeyNote Trust Management System Version. Internet RFC 2704, 1999.

[6] BLAZE, M., FEIGENBAUM, J., IOANNIDIS, J., AND KEROMYTIS, A. The Role of Trust Management in Distributed Systems. *Secure Internet Programming* (1999).

[7] BLAZE, M., FEIGENBAUM, J., AND LACY, J. Decentralized Trust Management. In *IEEE Conference on Privacy and Security* (1996).

[8] BOBBA, R. B., ESCHENAUER, L., GLIGOR, V., AND ARBAUGH, W. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. Tech. Rep. 2002-44, ISR, May 2002.

[9] BUCHEGGER, S., AND BOUDEC, J. L. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain: IEEE Computer Society, January 2002, pages 403-410.*

[10] CAPKUN, S., BUTTYAN, L., AND HUBAUX, J. Self-organized public-key management for mobile ad hoc networks. In *Proceedings of the ACM workshop on Wireless security (International Conference on Mobile Computing and Networking)* (Atlanta, GA, USA, September 2002), ACM, ACM Press New York, NY, USA.

[11] CHERNIAK, M., FRANKLIN, M., AND ZDONIK, S. Expressing User Profiles for Data Recharging. *IEEE Personal Communications* (July 2001).

[12] CHU, Y.-H., FEIGENBAUM, J., LAMACCHIA, B., RESNICK, P., AND STRAUSS, M. REFEREE: Trust management for Web Applications. *Computer Networks and ISDN Systems 29*, 8–13 (1997), 953–964.

[13] DING, L., ZHOU, L., AND FININ, T. Trust Based Knowledge Outsourcing for Semantic Web Agents. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence* (October 2003).

[14] IEEE. *IEEE Std 802.11, 1999 Edition*, r2003 ed., 1999.

[15] JONKER, C. M., AND TREUR, J. Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99)* (Berlin, 30– 2 1999), F. J. Garijo and M. Boman, Eds., vol. 1647, Springer-Verlag: Heidelberg, Germany, pp. 221–231.

[16] KAGAL, L., FININ, T., AND JOSHI, A. A Policy Based Approach to Security for the Semantic Web. In *2nd International Semantic Web Conference (ISWC2003)* (September 2003).

[17] KAGAL, L., FININ, T., AND PENG, Y. A Delegation Based Model for Distributed Trust. In *Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents, International Joint Conferences on Artificial Intelligence* (August 2001).

[18] LAURENT ESCHENAUER, VIRGIL D. GLIGOR, J. B. On trust establishment in mobile ad-hoc networks. In *Proc. of the Security Protocols Workshop, Cambridge, UK, April 2002*.

[19] PARKER, J., UNDERCOFFER, J. L., PINKSTON, J., AND JOSHI, A. On Intrusion Detection in Mobile Ad Hoc Networks. In *23rd IEEE International Performance Computing and Communications Conference – Workshop on Information Assurance* (April 2004), IEEE.

[20] PATWARDHAN, A., PARKER, J., JOSHI, A., IORGA, M., AND KARYGIANNIS, T. Secure Routing and Intrusion Detection in Ad Hoc Networks. In *Proceedings of the 3rd International Conference on Pervasive Computing and Communications* (Kauai Island, Hawaii, March 2005), IEEE, pp. 191–199.

[21] PERICH, F. MoGATU: Data Management in Pervasive Computing Enviroments. http://mogatu.umbc.edu/, 2001-2005.

[22] PERICH, F., JOSHI, A., FININ, T., AND YESHA, Y. On Data Management in Pervasive Computing Environments. *IEEE Transactions on Knowledge and Data Engineering* (May 2004). Accepted for publication.

[23] PERICH, F., UNDERCOFFER, J. L., KAGAL, L., JOSHI, A., FININ, T., AND YESHA, Y. In Reputation We Believe: Query Processing in Mobile Ad-Hoc Networks. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services* (Boston, MA, August 2004).

[24] PERICH, F., UNDERCOFFER, J. L., KAGAL, L., JOSHI, A., FININ, T., AND YESHA, Y. In Reputation We Believe: Query Processing in Mobile Ad-Hoc Networks. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services* (Boston, MA, August 2004).

[25] RICHARDSON, M., AGRAWAL, R., AND DOMINGOS, P. Trust Management for the Semantic Web. In *2nd International Semantic Web Conference (ISWC2003)* (2003).

[26] S. MARTI, T.J. GIULI, K. L., AND BAKER, M. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000, August 2000*.

[27] ZAPATA, M., AND ASOKAN, N. Securing ad hoc routing protocols. In *Proceedings of the ACM workshop on Wireless security (International Conference on Mobile Computing and Networking)* (Atlanta, GA, USA, September 2002), ACM, ACM Press New York, NY, USA, pp. 1 – 10.

[28] ZENG, X., BAGRODIA, R., AND GERLA, M. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *Workshop on Parallel and Distributed Simulation* (1998).

Jim Parker received his B.S. degree in Computer Science from James Madison University in 1985, and his M.S. degree in Computer Science from University of Maryland, Baltimore County (UMBC) in 1998. He is currently a Ph.D. candidate in Computer Science at UMBC. As a member of the eBiquity research group at UMBC, his research focus is in the area of security as applied to mobile ad hoc computing environments.

Anand Patwardhan received his B.E. degree in Computer Engineering from the University of Pune, India, in 2000, and his M.S. Degree in Computer Science and Engineering from the OGI School of Science and Engineering at OHSU, Portland, Oregon in 2002. He is currently a Ph.D. Candidate in the CSEE Department at the University of Maryland, Baltimore County, and is working on trustworthy data management for mobile devices in pervasive environments.

Filip Perich is a Senior Research Engineer at Cougaar Software, in McLean, Virginia, and an Adjunct Assistant Professor at University of Maryland, Baltimore County (UMBC). He received Ph.D. degree in Computer Science from UMBC in 2004, M.S. degree in Computer Science in 2002, and B.A. degree in Mathematics from Washington College, Maryland in 1999. Dr. Perich's primary research focus is on intelligent agent software and data management in distributed systems, particularly mobile / pervasive ad hoc networks. Currently, he is developing novel military and commercial intelligent-agent-based distributed middleware with an emphasis on C4ISR cognitive knowledge fusion, visualization, and control. As a member of the eBiquity research group at UMBC, Dr. Perich also continues working on research projects related to software agents, data management, wireless ad hoc networks, ubiquitous computing, and the Semantic Web. Dr. Perich has authored over 20 referred publications and served as a conference organization and committee member for multiple conferences and workshops.

Anupam Joshi is an Associate Professor of Computer Science and Electrical Engineering at UMBC. Earlier, he was an Assistant Professor in the CECS department at the University of Missouri, Columbia. He obtained a B. Tech degree in Electrical Engineering from IIT Delhi in 1989, and a Masters and Ph.D. in Computer Science from Purdue University in 1991 and 1993 respectively. His research interests are in the broad area of networked computing and intelligent systems. His primary focus has been on data management and security for mobile, pervasive, and sensor systems. He has created agent based middleware to support discovery, composition, and secure access of services/data over both infrastructure based and ad-hoc wireless networks, as well as systems that integrate sensors with the grid. He is also interested in Semantic Web and Data/Web Mining, where he has worked on creating personalized and secure web spaces using a combination of agents, policies, and soft computing. He has published over 100 technical papers, and has obtained research support from NSF, NASA, DARPA, DoD, IBM, AetherSystens, HP, AT&T and Intel. He has presented tutorials in conferences, served as guest editor for special issues for IEEE Personal Comm., Comm. ACM etc., and served as an Associate Editor of IEEE Transactions of Fuzzy Systems from 99-03. At UMBC, Joshi teaches courses in Operating Systems, Mobile Computing, Networking, Security, and Web Mining. He is a member of IEEE, IEEE-CS, and ACM.

Tim Finin is a Professor of Computer Science and Electrical Engineering at the University of Maryland, Baltimore County (UMBC). He has over 30 years of experience in the applications of Artificial Intelligence to problems in information systems and is currently working on the theory and applications of the semantic web, intelligent software agents, and pervasive computing. He holds degrees from MIT and the University of Illinois. Prior to joining the UMBC, he held positions at Unisys, the University of Pennsylvania, and the MIT

AI Laboratory. Finin is the author of over 225 refereed publications and has received research grants and contracts from a variety of sources. He is a former AAAI councilor and on the the board of directors of the Computing Research Association.