ISSN:0975-9646

A Strategy Implementing System for Secure Ad hoc Networks

[1]Nikesh Kumar Sharma, [2]Aamir Mohammad, [3]Shilpi Jain

[1]Asst. Prof, CSE Deptt. IITM, Gwalior, India
[2] Asst. Prof, IT Deptt. IITM, Gwalior, India
[3]Asst. Prof, IT Deptt. IITM, Gwalior, India

Abstract-In most existing trust evaluation studies, a single value is computed based on the ratings given to a service provider to indicate the current trust level. In MANETs, multicast group members frequently change due to node mobility; therefore, supporting secure authentication and authorization in a multicast MANET is more critical than that in a wired network with a centralized certificate authentication (CA) server. This paper thus proposes a two-step secure authentication approach for multicast MANETs. In particular, we examine routing attacks, such as link spoofing and colluding misrelay attacks, as well as countermeasures against such attacks in existing MANET protocols. We seek to combine the notions of "social trust" derived from social networks with "quality-of-service (QoS) trust" derived from information and communication networks to obtain a composite trust metric. A distributed scheme is designed to acquire, maintain, and update trust records associated with the behaviors of nodes' forwarding packets and the behaviors of making recommendations about other nodes. Simulations show that the proposed trust evaluation system can significantly improve the network throughput as well as effectively detect malicious behaviors in ad hoc networks.

Index Terms: Ad hoc networks, Security, Trust modeling and evaluation.

1 INTRODUCTION

In recent years, Service-Oriented Computing (SOC) has emerged to be an increasingly important research area attracting attention from both the research and industry communities. In SOC applications, various services are provided to clients by different providers in a loosely coupled environment. In such context, a service can refer to a transaction, such as selling a product online (i.e., the traditional online services), or a functional component implemented by web service technologies [2]. However, when a client looks for a service out of a large pool of services provided by different service providers, in addition to functionality, the trust of service is also a key factor for service selection. Conceptually, trust is the measure taken by one party on the willingness and ability of another party to act in the interest of the former party in a certain situation. If the trust value is in the range of it can be taken as the subjective probability by which, one party expects that another party can perform a given action . In SOC environments, the trust issue is very important. An effective and efficient trust management system is highly desirable and critical for service clients to identify potential risks, providing objective trust results, and preventing huge financial loss .In general, in a trust management enabled system, service clients can provide feedback and trust ratings after completed transactions. A good reputation results from the high quality of delivered services in a certain time period. Based on the ratings, the trust management system can calculate the reputation-based trust value of a service provider to reflect the quality of services in a certain time period, with more weights assigned to later transactions [3]. In most existing trust evaluation models a single trust value (e.g., a value in the range of [0, 1]) is computed to reflect the global trust level of a target accumulated in a certain time period (e.g., in the latest six months). The calculation of the final trust value is based on either all the ratings given for the latest time period [or the current trust value for previous transactions and the rating for the latest transaction. Single trust value systems are easy to use in trust oriented service comparison and selection. However, a single trust value computed by a service management authority cannot depict the real trust level very well under certain circumstances. For example, if there are two service providers A and B with their final trust values TA _ 0:7 and TB _ 0:7 (each of them is in the range of [0, 1]), does it mean that both A and B have the same trust level? It is not true I A's trust values are turning worse with an accumulated value of 0.7 while B's trust values are becoming better. In this case, B is better than A in terms of predicting the trust level of a forthcoming transaction. In order to observe the trend of trust changes, the complete set of trust ratings for a certain time period would be required. However, service clients are usually interested in a long service history (e.g., recent one month, three months, six months, or one year). Therefore, in such a situation, transferring a complete set of trust ratings to a client will be too costly in terms of communication overhead**.**

Trust Definition*:* Although definitions of trust have been borrowed from the social science literature, there is no clear consensus on the definition of trust in distributed computer networks. Trust has been interpreted as reputation, trusting opinion, probability , etc. *Trust Metrics:* As a nature consequence of the confusion in trust definition, trust has been evaluated in very different ways. Some schemes employ linguistic descriptions of trust relationship, such as in PGP, PolicyMaker distributed trust model [4], trust policy language [5], and SPKI/SDSI public-key infrastructure .In some other schemes, continuous or discrete numerical values are assigned to measure the level of trustworthiness. For example, in an entity's opinion about the trustworthiness of a certificate is described by a continuous value in [0, 1]. In a two-tuple in describes the trust opinion. In the metric is a triplet in , where the elements in the triplet represent belief, disbelief, and uncertainty, respectively. In [4], discrete integer numbers are used.

Currently, it is very difficult to compare or validate these trust metrics because a fundamental question has not been well understood. What is the physical meaning of trust? We need trust metrics to have clear physical meanings, for establishing the connection between trust metrics and observation (trust evidence) and justifying calculation/policies/rules that govern calculations performed upon trust values.

***Quantitative Trust Models**:* Many trust models have been developed to model trust transit through third parties. For example, the simplest method is to sum the number of positive ratings and negative ratings separately and keep a total score as the positive score minus the negative score. This method is used in eBay's reputation forum. In subjective logics are used to assess trust values based on the triplet representation of trust. In fuzzy logic provides rules for reasoning with linguistic trust metrics. In the context of the "Web of Trust," many trust models are built upon a graph where the resources/entities are nodes and trust relationships are edges, such as In [6] and [7]. Then, simple mathematics, such as minimum, maximum, and weighted average, is used to calculate unknown trust values through concatenation and multipath trust propagation. In , a Bayesian model is used to take binary ratings as input and compute reputation scores by statistically updating beta probability density functions. Although a variety of trust models are available, it is still not well understood what fundamental rules the trust models must follow. Without a good answer to this question, the design of trust models is still at the empirical stage. We approach the trust evaluation problem from a definition of trust given by Gambetta in [8]. It states that trust is a level of likelihood with which an agent will perform a particular action before such action can be monitored and in a context in which it affects our own actions. It is clear that trust relationship, involves two entities and a specific action. The concept of trust exists because we are not sure whether the agent will perform the action or not in some circumstances. In the proposed information theoretic framework of trust modeling and evaluation, trust is a measure of uncertainty; as such trust values can be measured by entropy. From this understanding of trust, we develop axioms that address the basic rules for establishing trust through a third party (concatenation propagation) and through recommendations from multiple sources (multipath propagation). Based on these axioms, we develop techniques that calculate trust values from observation and design two models that address the concatenation and multipath trust propagation problems in ad hoc networks. The proposed models are then applied to improve the performance and security of ad hoc routing protocols. In particular, we investigate trust relationship associated with packet forwarding as well as making recommendations. We develop a distributed scheme to build, maintain, and update trust records in ad hoc networks. Trust records are used to assist route selection and to perform malicious node detection. Simulations are performed to evaluate the effectiveness of theproposed models in ad hoc networks. Individual users obtain the trust values of forwarding packets and making recommendations in a distributed way. The malicious nodes can be detected and their types can also be identified. The proposed scheme can also track the dynamics of the networks adaptively. Compared with a baseline scheme without trust evaluation, the proposed scheme can select the route with higher recommended quality so that the packet dropping rates are greatly reduced**.[IEEE 2006]**

2.RELATED WORK

Routing Protocols in MANETs

The goal of routing in a MANET is to discover the most recent topology of a continuously changing network to find a correct route to a specific node. Routing protocols in a MANET can be classified into two categories: reactive routing protocols (e.g., AODV) and proactive routing protocols (e.g., OLSR). In reactive routing protocols, nodes find routes only when they must send data to the destination node whose route is unknown. On the other hand, in proactive protocols, nodes periodically exchange topology information, and hence nodes can obtain route information any time they must send data. In this section, we describe two standard routing protocols that currently are being researched actively, that is, AODV and OLSR.

AODV

AODV [9] is a reactive routing protocol designed for a mobile ad hoc network. In AODV, when a source node S wants to send a data packet to a destination node D and does not have a route to D, it initiates route discovery by broadcasting a route request (RREQ) to its neighbors. The immediate neighbors who receive this RREQ rebroadcast the same RREQ to their neighbors. This process is repeated until the RREQ reaches the destination node. Upon receiving the first arrived RREQ, the destination node sends a route reply (RREP) to the source node through the reverse path where the RREQ arrived. The same RREQ that arrives later will be ignored by the destination node. In addition, AODV enables intermediate nodes that have sufficiently fresh routes (with destination sequence number equal or greater than the one in the RREQ) to generate and send an RREP to the source node.

OLSR Protocol

OLSR [10] is a proactive routing protocol, that is, it is based on periodic exchange of topology information. The key concept of OLSR is the use of multipoint relay (MPR) to provide an efficient flooding mechanism by reducing the number of transmissions required. In OLSR, each node selects its own MPR from its neighbors. Each MPR node maintains the list of nodes that were selected as an MPR; this list is called an MPR selector list. Only nodes selected as MPR nodes are responsible for advertising, as well as forwarding an MPR selector list advertised by other MPRs.

Routing Message in OLSR — generally, in the OLSR protocol, two types of routing messages are used, namely, a

HELLO message and a topology control (TC) message. Neighbor A HELLO message is the message that is used for neighbor sensing and MPR selection. In OLSR, each node generates a HELLO message periodically. A node's HELLO message contains its own address and the list of its one-hop neighbors. By exchanging HELLO messages, each node can learn a complete topology up to two hops. HELLO messages are exchanged locally by nodes and are not forwarded further to other nodes. A TC message is the message that is used for route calculation. In OLSR, each MPR node advertises TC messages periodically. A TC message contains the list of the sender's MPR selector. In OLSR, only MPR nodes are responsible for forwarding TC messages. Upon receiving TC messages from all of the MPR nodes, each node can learn the partial network topology and can build a route to every node in the network.

MPR Selection — For MPR selection, each node selects a set of its MPR nodes that can forward its routing messages. In OLSR, a node selects its MPR set that can reach all its two-hop neighbors. In case there are multiple choices, the minimum set is selected as an MPR set.

3. Routing Attacks Against Manet Protocols

Flooding Attack

The aim of the flooding attack [11] is to exhaust the network resources, such as bandwidth and to consume a node's resources, such as computational and battery power or to disrupt the routing operation to cause severe degradation in network performance. For example, in AODV protocol, a malicious node can send a large number of RREQs in a short period to a destination node that does not exist in the network. Because no one will reply to the RREQs, these RREQs will flood the whole network. As a result, all of the node battery power, as well as network bandwidth will be consumed and could lead to denial-of-service. In [12], the authors show that a flooding attack can decrease throughput by 84 percent.

Blackhole Attack

In a black hole attack, a malicious node sends fake routing information, claiming that it has an optimum route and causes other good nodes to route data packets through the malicious one.

For example, in AODV, the attacker can send a fake RREP (including a fake destination sequence number that is fabricated to be equal or higher than the one contained in the RREQ) to the source node, claiming that it has a sufficiently fresh route to the destination node. This causes the source node to select the route that passes through the attacker. Therefore, all traffic will be routed through the attacker, and therefore, the attacker can misuse or discard the traffic.

Link Spoofing Attack

In a link spoofing attack, a malicious node advertises fake links with non-neighbors to disrupt routing operations. For example, in the OLSR protocol, an attacker can advertise a fake link with a target's two-hop neighbors. This causes the target node to select the malicious node to be its MPR. As an MPR node, a malicious node can then manipulate data or routing traffic, for example, modifying or dropping the routing traffic or performing other types of DoS attacks.

Replay Attack

In a MANET, topology frequently changes due to node mobility. This means that current network Topology might not exist in the future. In a replay attack [13], a node records another node's valid control messages and resends them later. This causes other nodes to record their routing table with stale routes. Replay attack can be misused to impersonate a specific node or simply to disturb the routing operation in a MANET.

Wormhole Attack

A wormhole attack [14] is one of the most sophisticated and severe attacks in MANETs. In this attack, a pair of colluding attackers record packets at one location and replay them at another location using a private high speed network. The seriousness of this attack is that it can be launched against all communications that provide authenticity and confidentiality.

Countermeasures Against Attacks In A Manet

In this section, we discuss solutions that are proposed to counter against routing attacks described in the previous section.

Solutions To The Flooding Attack

In [11], the authors proposed a simple mechanism to prevent the flooding attack in the AODV protocol. In this approach, each node monitors and calculates the rate of its neighbors' RREQ. If the RREQ rate of any neighbor exceeds the predefined threshold, the node records the ID of this neighbor in a blacklist. Then, the node drops any future RREQs from nodes that are listed in the blacklist. One limitation of this approach is that it cannot prevent against the flooding attack in which the flooding rate is below the threshold. Another drawback of this approach is that if a malicious node impersonates the ID of a legitimate node and broadcasts a large number of RREQs, other nodes might put the ID of this legitimate node on the blacklist by mistake. In [12], the authors proposed an adaptive technique to mitigate the effect of a flooding attack in the AODV protocol. This technique is based on statistical analysis to detect malicious RREQ floods and avoid the forwarding of such packets. Similar to [11], in this approach, each node monitors the RREQ it receives and maintains a count of RREQs received from each sender during the preset time period. The RREQs from a sender whose

RREQ rate is above the threshold will be dropped without forwarding. Unlike the method proposed in [11], where the threshold is set to be fixed, this approach determines the threshold based on a statistical analysis of RREQs. The key advantage of this approach is that it can reduce the impact of the attack for varying flooding rates.

Solutions To The Blackhole Attack

In [15], the authors introduce the route confirmation request (CREQ) and route confirmation reply (CREP) to avoid the blackhole attack. In this approach, the intermediate node not only sends RREPs to the source node but also sends CREQs to its next-hop node toward the destination node. After receiving a CREQ, the next-hop node looks up its cache for a route to the destination. If it has the route, it sends the CREP to the source. Upon receiving the CREP, the source node can confirm the validity of the path by comparing the path in RREP and the one in CREP. If both are matched, the source node judges that the route is correct. One drawback of this approach is that it cannot avoid the blackhole attack in which two consecutive nodes work in collusion, that is, when the next-hop node is a colluding attacker sending CREPs that support the incorrect path. In [16], the authors proposed a solution that requires a source node to wait until a RREP packet arrives from more than two nodes. Upon receiving multiple RREPs, the source node checks whether there is a shared hop or not. If there is, the source node judges that the route is safe. The main drawback of this solution is that it introduces time delay, because it must wait until multiple RREPs arrive. In [17], the authors analyzed the black hole attack and showed that a malicious node must increase the destination sequence number sufficiently to convince the source node that the route provided is sufficiently enough. Based on this analysis, the authors propose a statisticalbased anomaly detection approach to detect the blackhole attack, based on differences between the destination sequence numbers of the received RREPs. The key advantage of this approach is that it can detect the attack at low cost without introducing extra routing traffic, and it does not require modification of the existing protocol. However, false positives are the main drawback of this approach due to the nature of anomaly detection.

Solutions To The Link Spoofing Attack

To detect a link spoofing attack, the author of [18] proposed a location information-based detection method by using cryptography with a GPS and a time stamp. This approach requires each node to advertise its position obtained by the GPS and the time stamp to enable each node to obtain the location information of the other nodes. This approach detects the link spoofing by calculating the distance between two nodes that claim to be neighbors and checking the likelihood that the link is based on a maximum transmission range. The main drawback of this approach is that it might not work in a situation where all MANET nodes are not equipped with a GPS. Furthermore, attackers can still advertise false information and make it hard for other nodes to detect the attack. In [19], the authors show that a malicious node that advertises fake links with a target's two-hop neighbors can successfully make the target choose it as the only MPR. Through simulations, the authors show that link spoofing can have a devastating impact on the target node. Then, the authors present a technique to detect the link spoofing attack by adding two-hop information to a HELLO message. In particular, the proposed solution requires each node to advertise its two-hop neighbors to enable each node to learn complete topology up to three hops and detect the inconsistency when the link spoofing attack is launched. The main advantage of this approach is that it can detect the link spoofing attack without using special hardware such as a GPS or requiring time synchronization. One limitation of this approach is that it might not detect link spoofing with nodes further away than three hops.

Solutions To The Replay Attack

In [20], the authors proposed a solution to protect a MANET from a replay attack by using a

time stamp with the use of an asymmetric key. This solution prevents the replay attack by comparing the current time and time stamp contained in the received message. If the time stamp is too far from the current time, the message is judged to be suspicious and is rejected. Although this solution works well against the replay attack, it is still vulnerable to a wormhole attack where two colluding attackers use a high speed network to replay messages in a far-away location with almost no delay. This attack will be discussed in the next subsection.

Solutions To The Wormhole Attack

In packet leashes are proposed to detect and defend against the wormhole attack. In particular, the authors proposed two types of leashes: temporal leashes and geographical leashes. For the temporal leash approach, each node computes the packet expiration time, *te*, based on the speed of light *c* and includes the expiration time, *te*, in its packet to prevent the packet from traveling further than a specific distance, *L*. The receiver of the packet checks whether or not the packet expires by comparing its current time and the *te* in the packet. The authors also proposed TIK, which is used to authenticate the expiration time that can otherwise be modified by the malicious node. The main drawback of the temporal leash is that it requires all nodes to have tightly synchronized clocks. For the geographical leash, each node must know its own position and have loosely synchronized clocks. In this approach, a sender of a packet includes its current position and the sending time. Therefore, a receiver can judge neighbor relations by computing distance between itself and the sender of the packet. The advantage of geographic leashes over temporal leashes is that the time synchronization needs not to be highly tight. In [18], the authors offer protection against a wormhole attack in the OLSR protocol. IEEE 07

4 PROPOSED WORK

In this section, two trust based routing protocols are presented. Each of them makes use of different trust quantification and embedded trust in different context. The goals that two of them intend to achieve are also not the same.

4.1 Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks

This protocol is described in [1]. It provides a complete framework from trust evaluation to trust routing. In order to have a good understanding of trust, the whole general idea of this paper will be discussed as following.

4.1.1 Trust Evaluation

T {subject: agent, action} is used to denote the trust value that subject has for action with regard to agent. Similarly, P {subject: agent, action} denotes the probability subject estimate if agent will perform action correctly. In order to get a correct comprehensive trust, both first-hand and second-hand evidence should be considered. Therefore, we need to concatenate the trust. Two contexts exist in this situation. One is whether another node will transmit the packet correctly while the other is whether it will give a good recommendation. The respective trust values are presented as T {subject: agent, transmit} and T {subject: agent, recommend}. The simplest model of concatenation trust is shown in figure 1.The final trust is

T (A: C, action) = RAB TBC

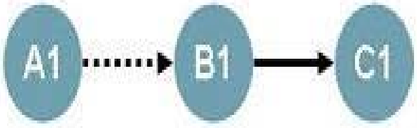

Figure 1: Concatenation trust propagation

If there is more than one recommendations, just like the situation shown in figure 2. The final trust is

T {A: C, action} = w1 (RAB TBC) + w2 (RAD TDC)
Where
w1= RAD / RAB+RAD,
w2= RAB / RAB+RAD

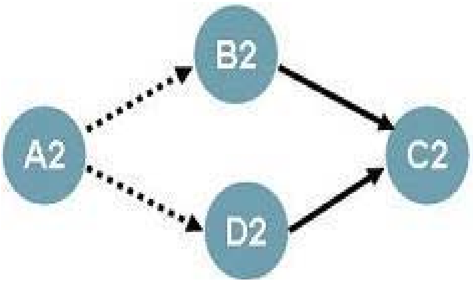

Figure 2: combining multiple recommendations.

Now, assume that A is going to evaluate the firsthand trust of B. suppose A requires B to perform the action N times while B actually performs k times. There is a common approach in probability from which we get

P {A: B, action} = k/n

Then according to this paper, using Bayesian method, at last we get

P {A: B, action} = k+1/n+2

From P {A: B, action} we can get the corresponding trust values through the entropy formula.

4.1.2 Trusted routing

The routing process can be summarized into the following steps:

1. Route discovery: it is just like the route discovery in DSR. Suppose A starts this process to communicate with D. At the end, A collects all the available routes to D;
2. Validate routes: Node A check the trust values of the intermediate nodes along the path. Assuming node B's trust value is missing in A's trust table or its trust values is below a certain threshold, put B into a set X;
3. During the transmission, node A updates its trust table based on the observations. When some malicious behavior is found, A will discard this path and find another candidate path or restart a new discovery.
4. Compute trust values for every node in X based on the trust graph.

4.2 Trust Based Adaptive On demand Ad Hoc Routing Protocol

This section gives a general analysis of [3]. This paper aimed to hide the source node's identity from intermediate nodes in route discovery. There is an assumption that there are well-defined cryptographically mechanisms and each node has several mechanisms to choose. It is certain that different mechanisms have different complexity and consume different amount of power. Therefore, trust is introduced to determine which mechanism to use. The discipline is that if the next node is more trustworthy, a simpler method will be choosing. Of course the choice is also based on the security level demanded by the application. As is shown in table 1, the security level and the trust levels cooperate to decide the encryption policy. The protocol proposed is based on AODV as we discussed above. In order to give a more detailed example of routing in MANETs, the route discovery process will be described as follows

1. Source S wants to communicate with node D. It broadcasts the request message RREQ. RREQ includes the level of security it requires and D's id, a sequential number and S's id encrypted by D's public key. RREQ is like this :{ RREQ, seqnum, Pb D [Si d], Di d, SL}
2. Node A receives RREQ. It looks up its trust list for the trust values of the neighbors. And A will encrypt if own id with proper policy and append in the message. The message which will sent by A is like this:{RREQ, seqnum, Pb D[Pv A[Aid], Pb D[Sid], Did , SL} where Pv A is the private key of A.
3. D receives RREQ. It uses its private key and the public key of the intermediate nodes to authenticate them. D checks if there are any bad nodes. If they are all trusted, D generates a number for the flow Fid , and broadcasts the following message(suppose A and B are the intermediate nodes): {RREP,Pb B[Fid , Pb A[Fid , Pb S[Pv D[Fid]]]]};
4. Intermediate node that receives the RREP uses its private key to decrypt the message and gets the flow id. Then it updates its route table with Fid designated to destination D;
5. S receives RREP, uses its private key to decrypt the message and D's public key to identify the destination. Afterwards, it will send message with the flow id Fid. Thus, the intermediate nodes will never know who the source is and just pass data according to Fid.

5. TRUSTED PATH SELECTION

I find that the path selection in the above document is not convincing in some situations. Let us see an extreme example in figure 3. There are two paths and the trust of either path equals 0.216. However, it is easy for us to choose the former one. For the node with trust 0.3 is more likely to break sometime later. Therefore, we have to find some methods to choose the better path automatically.

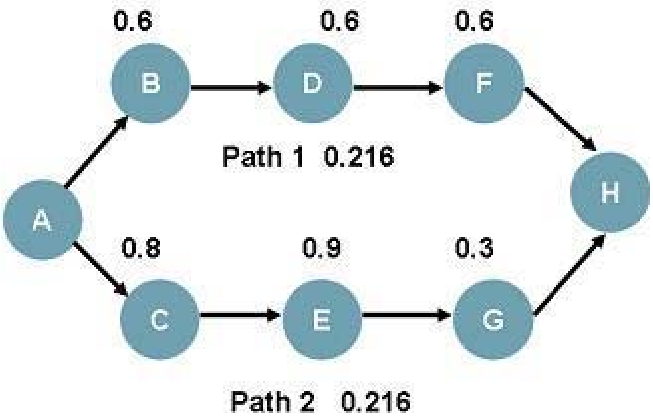

Figure 3: An extreme example

Firstly, suppose Ti is the ith node's trust value along the path. Then the initial trust value of the path is computed as:

$$T' = \frac{1}{n}\sum_{i=1}^{n} T_i$$

A parameter that can reflect the fluctuation of the trust values need to be introduced. Let σ2 denote the variant:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (T_i - T')^2$$

Last we can combine the two above parameters together to show the trust of the path. The lengths of paths are also taken into consideration. What we want to get is the one with fewer nodes and bigger trust value. The final path trust is like follows:

$$T = T' - \frac{\mu n \sigma^2}{\mathrm{hop(max)}}$$

Where hop max is the maximal number of hops among all available paths. μ is a punishment factor. Finally, we will choose the path with the biggest path trust value.

CONCLUSION

This paper presented a mechanism for MANETs to enforce application communication policies. Under this mechanism, nodes supporting the same set of applications and enforcing the same policies construct a trusted multitier application centric network. Each tier of the network runs one application and enforces its associated policy. The application of the upper tier depends on the applications of the lower tiers to communicate. Only trusted nodes are allowed to join the network. Moreover, communication between them is regulated by the policies at every tier. To ensure trusted policy enforcement, we augment each node with a trusted kernel agent based on the TCG TPM. We evaluated the method through a prototype based on an IEEE 802.11 ad hoc network and through network simulations. The results demonstrate the feasibility of the proposed method as well as its low overhead.

REFERENCES

[1] Yan L. Sun, Wei Yu, "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks", 2006 IEEE, pp305-317

[2] A. J. Tardiff and J.W. Gowens, Editors, "ARL Advanced Telecommunication and Information Distribution Research Program (ATIRP)," Final Report, 1996-2001, June 2001.

[3] S. Staab (Editor), "The Pudding of Trust," IEEE Intelligent Systems, vol. 19, no. 5, pp. 74-88, 2004.

[4] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in Proc. 1997 New Security Paradigms Workshop, 1998, pp. 48–60.

[5] A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure: Or assigning roles to strangers," in Proc. IEEE Symp. Security and Privacy, May 2000, pp. 2–14.

[6] U. Maurer, "Modeling a public-key infrastructure," in Proc. Eur. Symp. Res. Comput. Security, vol. 1146, Lecture Notes in Computer Science, 1996, pp. 325–350.

[7] M. K. Reiter and S. G. Stubblebine, "Resilient authentication using path independence," IEEE Trans. Comput., vol. 47, no. 12, pp. 1351–1362, Dec. 1998

[8] D. Gambetta, "Can we trust trust?," in Trust: Making and Breaking Cooperative Relations, D. Gambetta, Ed. Oxford, U.K.: Dept. Sociology,Univ. Oxford, 2000, pp. 213–237.

[9] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc Ondemand Distance Vector (AODV) Routing," IETF RFC 3561, July 2003

[10] Th. Clausen et al., "Optimized Link State Routing Protocol," IETF Internet draft, draft-ietf-manet-olsr-11.txt, July 2003

[11] P. Yi et al., "A New Routing Attack in Mobile Ad Hoc Networks," Int'l. J. Info. Tech., vol. 11, no. 2, 2005.

[12] S. Desilva, and R. V. Boppana, "Mitigating Malicious Control Packet Floods in Ad Hoc Networks," Proc. IEEE Wireless Commun. and Networking Conf., New Orleans,LA, 2005.

[13] C. Adjih, D. Raffo, and P. Muhlethaler, "Attacks Against OLSR: Distributed Key Management for Security," 2nd OLSR Interop/Wksp., Palaiseau, France, July 28–29, 2005.

[14] Y-C. Hu, A. Perrig, and D. Johnson, "Wormhole Attacks in Wireless Networks," IEEE JSAC, vol. 24, no. 2, Feb. 2006.

[15] S. Lee, B. Han, and M. Shin, "Robust Routing in Wireless Ad Hoc Networks," 2002 Int'l. Conf. Parallel Processing Wksps., Vancouver, Canada, Aug. 18–21, 2002.

[16] M. Al-Shurman, S-M. Yoo, and S. Park, "Black Hole Attack in Mobile Ad Hoc Networks," ACM Southeast Regional Conf. 2004.

[17] S. Kurosawa et al., "Detecting Blackhole Attack on AODV-Based Mobile Ad Hoc Networks by Dynamic Learning Method," Proc. Int'l. J. Network Sec., 2006.

[18] D. Raffo et al., "Securing OLSR Using Node Locations," Proc. 2005 Euro. Wireless, Nicosia, Cyprus, Apr. 10–13, 2005.

[19] B. Kannhavong et al., "A Collusion Attack Against OLSR-Based Mobile Ad Hoc Networks," IEEE GLOBECOM '06.

[20] C. Adjih, D. Raffo, and P. Muhlethaler, "Attacks Against OLSR: Distributed Key Management for Security," 2nd OLSR Interop/Wksp., Palaiseau, France, July 28–29, 2005.