Triple Umpire System for Security of Mobile Ad Hoc Networks

Ayyaswamy Kathirvel [a] , Rengaramanujam Srinivasan[b]
a Assistant Professor, *Faculty of Computer Science and Engineering*
b Professor, Faculty of Computer Science and Engineering
EmailID: a kathir@crescentcollege.org, *b* drsrs@crescentcollege.org
B.S.Abdur Rahman University, Chennai – 600 048, Tamilnadu, India.

Abstract

A mobile ad hoc network (MANET) is a self-created self-organized and self-administering set of nodes connected via wireless links without the aid of any fixed infrastructure or centralized administrator. Protecting the network layer from malicious attacks is an important and challenging issue in both wired and wireless networks and the issue becomes even more challenging in the case of MANET. In this paper we propose a solution of triple umpiring system (TUS) that provides security for routing and data forwarding operations. In our system each node in the path from source to destination has dual roles to perform: packet forwarding and umpiring. In the umpiring role, each node in the path closely monitors the behavior of its succeeding node and if any misbehavior is noticed immediately flags off the guilty node. For demonstration, we have implemented the umpiring system by modifying the popular AODV protocol. Extensive simulation studies using QualNet 4.5 establish the soundness of the proposal.

Keywords : MANET, AODV, TUS, SCAN and malicious nodes.

1. Introduction

A mobile ad hoc network (MANET) is a self-created self-organized and self-administering set of nodes connected via wireless links without the aid of

A mobile ad hoc network (MANET) is a self-created self-organized and self-administering set of nodes connected via wireless links without the aid of any fixed infrastructure or centralized administrator. Each node moves and operates in a distributed peer-to-peer mode, generating independent data and acting as a router to provide multi-hop communication. MANET is ideally suited for potential applications in civil and military environments, such as responses to hurricane, earthquake, tsunami, terrorism and battlefield conditions. Security is an important aspect in such mission critical applications.

In this paper we tackle the problem of securing the network layer operations from malicious nodes. Malicious nodes may disrupt routing algorithms by transmitting a false hop count; they may drop packets, route the packets through unintended routes and so on. Our work rests on the foundations of a system already proposed: SCAN [1]. A SCAN system as follows:

In SCAN [1] two ideas are exploited to protect the mobile ad hoc network: (i) local collaboration where the neighboring nodes collectively monitor each other and (ii) information cross-validation by which each node monitors neighbors by cross-checking the overheard transmissions.

In SCAN, each node monitors the routing and packet-forwarding behavior of its neighbors and independently detects the existence of malicious nodes in its neighborhood. This is made possible because of wireless nature of the medium and all the involved nodes are within each other's transmission. In order to enable cross-checking they have modified AODV protocol and added a new field *next_ hop* in the routing messages so that each node can correlate the overheard packets accordingly.

While each node monitors it neighbors independently all the nodes in the neighborhood collaborate to convict a malicious node. An agreement between a minimum of k neighboring nodes is required for convicting a malicious node. Once its neighbors convict a malicious node the network reacts by depriving it of its right to access the network. In SCAN each node must possess a valid token in order to interact with other nodes. They have used asymmetric key cryptography to prevent forgeries of tokens. A group of nodes (minimum-k) can collaboratively sign a token, while no single node can do so. Further each node has to get its token renewed

periodically by its neighbors. A node which behaves continuously in a good manner can get its token renewed at less frequent intervals as compared to a fresh entrant node.

Our triple umpiring system has been strongly influenced by the above schemes. In our system all the active nodes have dual roles just as in SCAN; we also exploit promiscuous hearing functionality as done by SCAN. We have adopted the token concept from SCAN. We achieve the avoidance of malicious nodes by a system of tokens, which is similar to the ones used in SCAN [1]. Token is a pass or validity certificate enabling a node to participate in the network. It contains two fields: nodeID and status bit; nodeID is considered to be immutable. Initially the status bit of all participating nodes is set as 0 indicating "green flag" with freedom to participate in all network operations. It is assumed that a node cannot change its own status bit. When an umpiring node finds its succeeding node misbehaving it sends a M-Error message to the source and malicious node's status bit is changed using M-Flag message (set to 1 indicating "red flag"). With "red flag" on the culprit node is prevented from participating in the network.

Our objective is designing the security system is to keep the overhead as minimum as possible while optimizing the throughput. We do not use encryption or key algorithms as done by SCAN. We find that token issuing and token renewals and broadcasts to announce convictions create very large communication overheads and also degrade energy performance, which SCAN has completely over looked. There is no token renewal feature in our system. In our system all the nodes are pre issued with green tokens. They continue to enjoy the status until any immediate ancestor node, in its umpiring mode finds its next node misbehaving, sends the M-Error and M-Flag messages and red flag is set.

Just like SCAN in order to facilitate cogent promiscuous hearing we have used "*next_hop*" field with our AODV implementation. Our umpiring system can detect any false reporting of hop count during the route reply process RREP. In SCAN it is done by a set of 'k' neighbors. In our system, we take 'k' neighbors value is three. In our system the designated predecessor node in its umpiring role carries out both detection and conviction. The performance of the node is monitored by three umpires. Hence we called 'Triple Umpire System (TUS)'.

The rest of the paper is organized as follows: section 2 provides a models and assumptions; section 3 discusses triple umpire system models. Section 4 gives implementation of TUS. Section 5 presents simulation results; Section 6 discusses analysis of results; Section 7 discusses the related work and Section 8 gives the conclusions.

2. Models and Assumptions

Assumptions made in the design of triple umpiring system are as follows:

1.A wireless ad hoc network where nodes are free to move about or remain at stand still, at their will is assumed.
2.The source and the destination node are not malicious.
3.Nodes may fail at any time.
4.There exists a bi-directional communication link between any pair of nodes, which is a requirement for most wireless MAC layer protocols including IEEE 802.11 for reliable transmission.
5.Wireless interfaces support promiscuous mode of operation.

Promiscuous hearing means, over hearing by a node say A, messages not addressed to it, transmitted by a second node B, situated in the communication range of A, to a third node C. Most of the existing IEEE 802.11 based wireless cards support such promiscuous mode of operations, to improve routing protocol performance.

3. Triple Umpiring System Model

In the triple umpiring system each node is issued with a token at the inception. The token consists of two fields: NodeID and status. NodeID is assumed to be unique and deemed to be beyond manipulation; status is a single bit flag. Initially the status bit is preset to zero indicating a green flag. The token with green flag is a permit issued to each node, which confers it the freedom to participate in all network activities [1].

Each node in order to participate in any network activity, say Route Request RREQ, has to announce it's token. If status bit is "1" indicating "red flag" protocol does not allow the node to participate in any network activity.

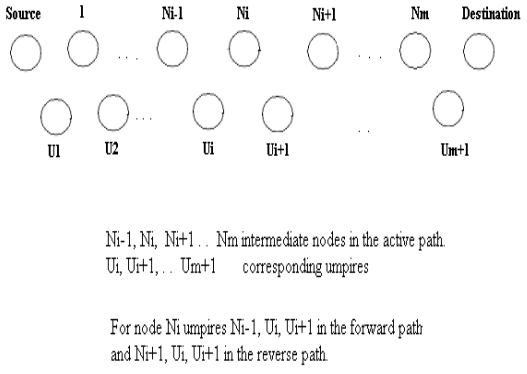

Fig. 1 Triple Umpiring System Model

We investigate a Triple Umpiring System (TUS) for securing the mobile ad hoc networks from attacks of the malicious nodes. In Fig. 1. N_{i-1}, N_i, N_{i+1} are the nodes in the active path; U_{i-1}, U_i, U_{i+1} are corresponding umpires. Each intermediate node in the active path is monitored by three umpires. For node N_i, N_{i-1}, U_i and U_{i+1} will be umpires in the forward path and N_{i+1}, U_i and U_{i+1}will be umpires in the reverse path. Any node in the active path, misbehaves as determined by the umpires is conjunction decide and the status bit of misbehave node is changed to red flag.

When Ni is found to be misbehaving – say dropping packets or changing Hop_count or sequence number, umpire nodes U_i , U_{i+1} and N_{i-1} in the forward path and N_{i+1} in the reverse path sends a M-ERROR message to the source and sets the status bit of guilty node Ni to "1" indicating red flag by M-Flag message.

In our system there is no change in the token – it can be used for the full lifetime of the node, if the node continuously behaves correctly. At the instance of the first offence the status of the guilty node is set to 1 preventing its further participation in the network.

We assume that no node can alter its own status bit. Only the designated umpire corresponding to the forward or reverse path under consideration can change the status bit. It is also assumed that a node cannot announce wrongly its token particulars – NodeID and status bit.

4. Implementation of TUS

We implement TUS on top of traditional AODV protocol, but its principal is applicable to other routing protocol as well. In order to enable such umpiring cross verification, we modify the famous AODV routing protocol and add a new field, next_hop, in the routing messages, so that a node can correlate the overheard packets accordingly. We implement of TUS is based on three important algorithms. Algorithm 1 describes TUS route request procedure; Algorithm 2 and Algorithm 3 discuss route reply procedure and packet forwarding procedure respectively.

Each node in order to participate in any network activity, says Route Request (RREQ), has to announce it's token as described in Algorithm 1. If the node status bit is "1" indicating red flag protocol does not allow the node to participate in any network activity. Otherwise, the token of the node status bit is "0" indicating green flag is a permit issued to each node, which confers it the freedom to participate in all network activities.

Algorithm 1: While sending an TUS RREQ packet

1: **for** each TUS RREQ packet (P) sent do
2: **if** each node status is green flag then
3: broadcast RREQ
4: nodeprevhop ← ▪ nodecurrenthop [node address]

5: neighhop1← prevhop[node address]
6: neighhop2← nexthop[node address]
7: repeat the steps from step 2 to step 6 until it reaches the destination node
8: **else**
9: drop umpire RREQ packet (P) sent
10: **endif**
11: **endfor**

In the triple umpiring system, three umpiring nodes are used to convict the malicious node. TUS all the nodes have dual roles – packet forwarding and umpiring. For node N_i, N_{i-1}, U_i and U_{i+1} will be umpires in the packet forwarding operation and N_{i+1}, U_i and U_{i+1}will be umpires in the route reply operation. Route reply process (RREP) as given in Algorithm 2.

Algorithm 2: While sending an TUS RREP packet

1: **for** each TUS RREP packet (P) sent do
2: **if** node status is green flag then
3: set designated umpires
4: neighhop1← prevhop[node address]
5: neighhop2← nexthop[node address]
6: nodenexthop ← ▪ nodeprevhop [node address]
7: unicast RREP to previous node
8: repeat the steps from step 2 to step 7 until it reaches the the source node
9: **if** nodecurrenthopcount and neighhop1 and neighhop2 is equal to nodenexthopcount then
10: process this RREP as specified in the standard protocol
11: **end if**
12: **endif**
13: **endfor**

When a node is found to be misbehaving – say dropping data packets, the corresponding umpires immediately sends a M-ERROR message to the source and the status bit of guilty node is set to "1" – red flag using M-Flag message as shown in algorithm 3. In order to correctly correlate the overheard messages an additional field next_hop has been introduced in all routing messages as done in SCAN [1]. Though there are several kinds of misbehavior that could be captured by promiscuous hearing we are focusing only on two types of malicious actions: dropping packets and transmitting false hop count.

Algorithm 3: While sending an TUS data packet

1: **for** each TUS DATA packet (P) sent do
2: **if** node status is green flag then
3: send a packet to the next forwarded node
4: it tampered with the payload or header of the currently sent packet
5: nodenexthop ← ▪ nodecurrentpacketheader
6: neighhop1← nodecurrentpacketheader
7: neighhop2← nodecurrentpacketheader
8: it keeps this header information until next packet is forwarded to the node
9: **else**
10: nodenextnode has dropped the packet, thus, the malicious node
11: nodeprevnode, neighhop1 and neighhop2 is umpire node for next immediate forwarded node
12: **if** nodenexthop← ▪ currentpacketheader and neighhop1← nodecurrentpacketheader and neighhop2← nodecurrentpacketheader
is not equal to prevhop ← ▪ currentpacketheader
13: it has marked as malicious node
14: it broadcast MERR packet to 1-hop or 2-hop node distance
15: nodenextnode status is marked as red flag
16: Umpires node sent link error message to the source node
17: process this RERR message as specified in the standard protocol
18: **endif**

19: **endif**
20: **endfor**

5. Simulations and Results

We use a simulation model based on QualNet 4.5 in our evaluation [2-3]. Our performance evaluations are based on the simulations of 100 wireless mobile nodes that form a wireless ad hoc network over a rectangular (1500 X 600 m) flat space. The MAC layer protocol used in the simulations was the Distributed Coordination Function (DCF) of IEEE 802.11 [4-5]. The performance setting parameters are given in Table 1.

Before the simulation we randomly selected a certain fraction, ranging from 0 % to 40 % of the network population as malicious nodes. We considered only two attacks – modifying the hop count and dropping packets. Each flow did not change its source and destination for the lifetime of a simulation run.

There are some critical issues to be discussed:

What happens when the umpires themselves are malicious?

We have investigated two types of malicious behavior of umpires (i) Umpires who after accepting umpireship are not acting as umpires, possibly, they are selfish they are conserving their own energy. These umpires, if they detect any misbehaving node they simply ignore them. We call them 'sleeping umpires'. The second category of umpires is strongly malicious in that, they totally behave in a malicious way. If they detect malicious behavior of a node they ignore; but they go to book innocent nodes.

Our experiments are based on four important parameters:

Simulation Time	1500 seconds
Propagation model	Two-ray Ground Reflection
Transmission range	250 m
Bandwidth	2 Mbps
Movement model	Random way point
Maximum speed	0 – 20 m/s
Pause time	0 seconds
Traffic type	CBR (UDP)
Payload size	512 bytes
Number of flows	10 / 20

Table 1 Parameter Settings

5.1. Throughput

In the world of MANET, packet delivery ratio has been accepted as a standard measure of throughput. Packet delivery ratio is nothing but a ratio between the numbers of packets received by the destinations to the number of packets sent by the sources. We present in Tables 2 the packet delivery ratios for TUS of 0 - 40 percentage of malicious node, with node mobility varying between 0 to 20 m/s.

Mobility (M/s)	**Percentage of Malicious nodes**				
	0 %	**10 %**	**20 %**	**30 %**	**40 %**
0	**98.28**	**82.88**	**76.97**	**70.95**	**64.51**
5	**96.28**	**75.55**	**67.69**	**61.51**	**45.59**
10	**95.10**	**74.40**	**65.39**	**58.42**	**44.42**
15	**94.12**	**73.54**	**64.07**	**58.26**	**40.86**
20	**93.73**	**75.06**	**63.46**	**56.29**	**39.12**

Table 2 Packet delivery ratios for TUS

From Tables 2 the following conclusions can be drawn:

1. In general packet delivery ratio decreases as mobility and percentage of malicious nodes increase.
2. In the case of 0% malicious nodes, packet delivery ratio drops from 98.28%, when the nodes are stationary to 93.73%, when the nodes are moving at 20 m/s. In presence 30 % and 40 % malicious nodes with 20 m/s mobility, the corresponding values are 56.29 and 39.12 percentages respectively.
3. TUS have low throughput because it have very high security.
4. In general packet delivery ratio decreases as security increase.

From the above results we conclude that SCAN leads to a substantial improvement over TUS, from the point of view of throughput.

5.2. Failure to deduct (False Negatives) Probability

False Negatives Probability can be defined as:

False Negatives Probability = number of malicious nodes left undetected/total number of malicious nodes.

Mobility (M/s)	Percentage of Malicious nodes				
	0 %	10 %	20 %	30 %	40 %
0	0	0.1018	0.1721	0.1731	0.1849
5	0	0.0878	0.1462	0.1471	0.1408
10	0	0.0571	0.0512	0.0618	0.0628
15	0	0.0618	0.0831	0.0871	0.0842
20	0	0.0754	0.1013	0.0873	0.0936

Table 3 False negatives for TUS

From Tables 3 the following conclusions can be drawn:

1. In general false negative probability increases as percentage of malicious nodes increases.
2. In the case of 20% malicious nodes when the nodes are moving at 20 m/s, false negative probability has high because number of malicious node in that particular area is high.

From the above results we conclude that TUS has the least false negative probability when compared with SCAN.

5.3. False Accusation (False Positives) Probability

We find false positive probability increases with increased mobility speed. The values vary from 0 to 0.0809 and are similar to the patterns obtained for SCAN [2].

Mobility (M/s)	Percentage of Malicious nodes				
	0 %	10 %	20 %	30 %	40 %
0	0	0	0	0	0
5	0	0.0078	0.0084	0.0091	0.0115
10	0	0.0112	0.0312	0.0354	0.0364
15	0	0.0224	0.0491	0.0511	0.0667
20	0	0.0564	0.0592	0.0612	0.0809

Table 4 False positives for TUS

It can be seen that lowest false positives probability is obtained with TUS. In other words innocent node booking is minimum with TUS.

5.4. Communication Overhead

Communication overhead can be evaluated based on the number of transmissions of control messages like RREQ, RREP, RERR in the case of plain AODV and in addition M_ERROR, M-Flag messages in the triple

umpiring system. RREQ are to be decimated to the entire network, where as RREP messages are unicasts. We present the communication overhead details in Table 5.

Mobility (M/s)	Percentage of Malicious nodes				
	0 %	10 %	20 %	30 %	40 %
0	9046	11422	13832	15390	16855
5	9618	12215	14882	16206	17697
10	10025	13084	15705	17151	18609
15	10576	14011	16659	18025	19808
20	11998	14661	17348	18713	20610

Table 5 Communication overhead for TUS

We find that communication overhead increases with mobility and TUS has the lowest communication overhead when compared with SCAN.

6. Analysis of Results

We present the plain AODV, SCAN and TUS results in Table 6.

Mobility (M/s)	Throughput for malicious node = 30 %		
	Plain AODV	SCAN	TUS
0	70.44	90	70.95
5	45.18	85	61.51
10	37.89	83	58.42
15	32.55	81	58.26
20	32.07	80	56.29

Table 6 Throughput for plain AODV, SCAN and TUS.

We find that SCAN yield much higher output as compared to plain AODV and TUS. The increase in communication overhead ranges from 8.8 % (plain AODV, 0 m/s mobility) to 16.4 % (plain AODV 20 m/s mobility). The corresponding values for SCAN are from 11 to 28 %.

Clearly with TUS, with more umpires involved in detection, false negatives and false positives probabilities decrease. Thus with TUS we have better rounding up of malicious nodes with very low communication overhead.

7. Related Works

The Key Distribution Center (KDC) architecture is the main stream in wired network because KDC has so many merits: efficient key management, including key generation, storage, distribution and updating. The lack of Trusted Third Party (TTPs) key management scheme is a big problem in ad hoc network [6 - 25].

Kong et al. [7] describe a solution that supports ubiquitous services to mobile hosts. In their design they distribute the certification authority functions through a threshold secret sharing mechanism, in which each entity holds a secret share and multiple entities in a local neighborhood jointly provide complete services. Thus no single entity in the network knows or holds the complete system secret (e.g. - a certification authority's signing key). Instead, each entity holds a secret share of the certification authority's secret key. Multiple entities, say k in one hop network locality jointly provide complete security services, as if a single omni present certification authority provided them.

Yong et al. [9][12] propose a novel cryptography for ad hoc network security, where they present a new digital signature algorithm for identity authentication and key agreement scheme. Their scheme has no central administrator. They have shown that their scheme can withstand man-in-middle and Byzantine mode conspiracy attacks.

Hubaux et al. [17] make a survey of threats and possible solutions for one security of ad hoc network. They extend the idea of public key infrastructure. Their system is similar to Pretty Good Policy (PGP) in the sense public key certificates are issued by the users. However they do not rely on certificate directories for the distribution of certificates. They present two algorithms in this connection.

All the above schemes only try to protect the system from the attacker, but not bother about quarantining attackers. The twin systems of *watchdog* and *pathrater* [18] not only detect the mischievous nodes but also prevent their further participation in the network. SCAN [1] also has similar action, but is more comprehensive, in the sense not only packet dropping but also other misbehaviors like giving wrong hop count are covered. Our TUS is an extension of the above two works.

8. Acknowledgements

We express our thanks to Dr. P. Kanniappan, the Vice Chancellor, Prof. V. M. Periasamy, the Register and Prof. K.M.Mehata, the Head, Department of CSE & Dean, School of Computer and Information Science B.S.A.Crescent Engineering College Chennai, Tamilnadu, India for the encouraging environment provided.

References

1. Hao Yang, James Shu, Xiaoqiao Meng and Songwu Lu, "SCAN: Self-Organized Network-Layer Security in Mobile ad hoc networks", IEEE Journals on selected areas in communications, vol. 24, No. 2, February 2006.
2.L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia, and M. Gerla, "Glomosim : A scalable network simulation environment. Technical Report 990027, 1999.
3.Scalable Networks Technologies: QualNet simulator version 4.5. http://www.scalable-networks.com
4.IEEE 802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, August, 1999.
5.A.Kathirvel and R.Srinivasan, "Performance Enhancement on demand routing protocol in mobile ad hoc networks", in Proc. Second National Conference, PSG tech, 2006.
6.Marianne A. Azer, Sherif M. El-Kassas, and Magdy S. El-Soudani, "Certification and revocation schemes in ad hoc networks survey and challenges, in proc. IEEE ICSNC 2007.
7.J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for MANET", in Proc. IEEE ICNP, 2001, pp. 251-260.
8. Lei Feng-Yu, Cui Guo-Hua, and Liao Xiao-Ding, "Ad hoc Networks security mechanism based on CPK", in proc. IEEE ICCISW, 2007, pp. 522 – 525.
9.Pi Jian Yong, Liu Xin Song, Wu Ai, Liu Dan, "A Novel Cryptography for Ad Hoc Network Security", in Proc. IEEE 2006, pp. 1448 -1451.
10.Michael Hauspie, and Isabelle Simplot-Ryl, "Enhancing nodes cooperation in ad hoc networks", in proc. IEEE 2007, pp. 130 – 137.
11.S. Capkun, L. Buttyan and J. Hubaux, "Self-organized public-key management for mobile ad hoc networks", IEEE Trans. Mobile Computing, vol. 2, No. 1, pp. 52-64, January, 2003.
12. Pi Jian Yong, Liu Xin Song, Wu Ai, Liu Dan, "A Novel Cryptography for Ad Hoc Network Security", in Proc. IEEE 2006, pp. 1448 -1451.
13.J. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in Mobile ad hoc networks", in Proc. ACM MobiHoc, 2001, pp. 146-155.
14.William Stallings, "Cryptography and network Security principles and Practices", Pearson Education, First edition, 2007.
15. Y. Zhang and W. Lee, "Intrusion detection in wireless ad hoc networks", in Proc. ACM MobiCom, 2000, pp. 275-283.
16. S. Capkun, J.Hubaux, and L. Buttyan, "Mobility helps security in ad hoc networks", in Proc. ACM MobiCom, 2003, pp 46-56.
17.J. Hubaux, I. Buttyan and S. Capkun, "The quest for security in mobile ad hoc networks", in Proc. ACM MobiHoc 2001, pp. 251-260.
18.Sergio Marti, T.J. Giuli, Kevin Lai and Mary Baker, "Mitigating routing misbehavior in mobile ad hoc networks", in proc. ACM MobiCom, 2000, pp- 255-265.
19. Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks", in Proc. IEEE WMCSA, June 2002, pp. 3-13.

20. Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure on-demand routing for ad hoc networks", in Proc. ACM MobiCom, 2002, pp. 12-23.
21. P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks", in Proc. CNDS, 2002, pp. 193-204.
22. K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Royer, "A secure protocol for ad hoc networks," in Proc. IEEEICNP, 2002, pp. 78-89.
23. M. Zapata and N. Asokan, "Securing ad hoc routing protocols", in Proc. ACM Wise, 2002, pp.1-10.
24.Azeddine Attir, Farid Nait Abdesselem, Brahim Bensaou, and Jalel Ben-Othman, "Logical Wormhole Prevention in Optimized Link State Routing Protocol", in proc IEEE GLOBECOM 2007, pp. 1011 – 1016.
25.Nidal Nasser and Yunfeng Chen, "Enhanced Intrusion Detection System for discovering malicious nodes in mobile ad hoc networks", in proc. IEEE ICC, 2007, pp. 1154- 1159.

Author Biographies:

A.Kathirvel - born in 1976 in Erode, Tamilnadu, India, received his B.E. degree from the University of Madras, Chennai, in 1998 and M.E. degree from the same University in 2002. He is currently with B.S.A. Crescent Engineering College in the Department of computer science and Engineering and pursing Ph.D. degree with the Anna University, Chennai, India. He is a member of the ISTE. His research interests are protocol development for wireless ad hoc networks, security in ad hoc networks.

Rengaramanujam Srinivasan -- born in 1940 in Alwartirunagari, Tamilnadu, India, received B.E. degree from the University of Madras, Chennai, India in 1962, M.E. degree from the Indian Institute of Science, Bangalore, India in 1964 and Ph.D. degree from the Indian Institute of Technology, Kharagpur, India in 1971. He is a member of the ISTE and a Fellow of Institution of Engineers, India. He has over 40 years of experience in teaching and research. He is presently working as a Professor of Computer Science and Engineering at BSA Crescent Engineering College, Chennai, India and is supervising doctoral projects in the areas of data mining, wireless networks, Grid Computing, Information Retrieval and Software Engineering.