International Journal of Computer and Electrical Engineering, Vol. 1, No. 2, June 2009
1793-8163

SACOM: Secure Ant Colony Optimization for MANETs

Ramkumar. K.R, Ravichandran. M, Hemachandar. N, ManojPrasadh. D, GaneshKumar. M

***Abstract* - Ant colony optimization (ACO), a swarm intelligence technique takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving routing problem in MANETs. Mobile ad hoc networks (MANETs) are infrastructure-less networks consisting of wireless mobile nodes which are organized in peer-to-peer and autonomous fashion. Initial work in ad hoc routing using ACO has considered only the problem of providing efficient mechanisms for finding paths in very dynamic networks, without considering security. Because of this, there are a number of attacks that hinders the systems normal behavior. In this paper we introduce SACOM framework which incorporates security mechanisms into routing protocols using ACO for ad hoc networks. In addition, SACOM is developed for preventing Wormhole Attack in the system without using specialized hardware.**

***Index Terms* – Security, Ant Colony Optimization, Mobile Adhoc Network, Cryptography.**

I. INTRODUCTION

A mobile ad-hoc network (MANET) is a collection of nodes capable of movement and connected dynamically in an arbitrary manner. Nodes of these networks function as routers which discover and maintain routes to other nodes in the network. The issue in MANETs is that routing protocols must be able to respond rapidly to topological changes in the network. At the same time the amount of control traffic generated by the routing protocols must be kept at a minimum due to the limited available bandwidth through radio interfaces. Several protocols dealing with the problems of routing in mobile ad-hoc networks have been developed. These protocols may generally be categorized as (a) proactive or table driven and (b) reactive or on demand driven. Proactive routing protocols attempts to maintain consistent, up-to-date routing information from each node to every other node all times. Theses protocols require each node to maintain on or more tables to store routing information and respond to topological changes by propagating updates through the network. Thus using a proactive protocol, a node is immediately, able to route or drop a packet. Examples of proactive protocols are TBRPF ''Topology Broadcast based on Reverse Path Forwarding'' and OLSR (Optimized Link State Routing protocol')'.

Ant colony optimization (ACO) [1] is a stochastic approach for solving combinatorial optimization problems like routing in computer networks. The idea of this optimization is based on the observation of how ants optimize food gathering in the nature. Ant colony optimization algorithms use artificial ants to iteratively construct a solution for an optimization problem. [2] A pheromone trail and a heuristic pheromone value is been used. A folk of ants move on the adjacent paths concurrently and asynchronously to find an optimum solution. Each ant selects the next hop by making a stochastic decision using the existing pheromone trails and heuristic information.
The solution is built incrementally as the ants move from one node to another node. While moving on the path, an ant evaluates this solution and deposits pheromone on its way. This pheromone trail will be used by the future ants to make a routing decision [4]. Ad-hoc wireless networks are increasing in popularity, due to the spread of laptops, sensor devices, PDA and other mobile electronic devices. These devices will eventually need to communicate with each other [5]. In some cases, without an adequate infrastructure to rely on the network must work properly. That's why we need routing protocols that can work without any central gateway to connect with [9]. At the same time, swarm intelligence has been used to solve optimization problems applied to data networks. Routing is one such optimization problem where swarm intelligence has been applied. Several routing protocols take advantage of that, i.e. Ant Net [8], ARA [3], AntHocNet [7] and PERA [6].

Securing protocols for mobile ad hoc networks presents unique challenges due to characteristics such as lack of predeployed infrastructure, centralized policy and control. In this paper, we make a number of contributions to the design of secure ad hoc routing protocols. The rest of the paper is organized as follows. In Section II, we discus about the possible attacks that can happen in MANET environment. Next we present our proposed model SACOM in Section III, followed by a detailed security analysis in Section IV.

II. ATTACKS USING MODIFICATION

Malicious nodes can cause redirection of network traffic and DoS attacks by altering control message fields or by forwarding routing messages with falsified values. Below we briefly describe several modification attacks against ARA and AntHocNet

A. Attacks Using Modification

1) Redirection by Modified Route Sequence Numbers: Protocols such as AntHocNet assign monotonically increasing sequence numbers to routes towards specific destinations. A route with a higher pheromone is preferred over one with a lower pheromone. Thus, in AntHocNet, any node may divert traffic through itself by advertising a route to a node with a destination sequence num greater than the authentic value.

2) Redirection with Modified Hop Counts: In AntHocNet, a redirection attack is possible by modification of the hop count field in route discovery messages. Malicious nodes can increase the chances they are included on a newly created route by resetting the hop count field of the Forward ant to zero. Similarly, by setting the hop count field of the same to infinity, created routes will tend to not include the malicious node.

3) Denial-of-service with Modified Source Routes: ACO utilizes source routes, thereby explicitly stating routes in data packets. These routes lack any integrity checks and a simple denial-of-service attack can be launched in ACO by altering the source routes in packet headers, such that the packet can no longer be delivered to the destination.

4) Tunneling: Ad hoc networks have an implicit assumption that any node can be located adjacent to any other node. A tunneling attack is where two or more nodes collaborate to encapsulate and exchange messages along existing data paths. Such collaborating nodes can pretend to be neighbors, and falsely represent the length of available paths by preventing honest intermediate nodes from correctly incrementing the path length metric. It is also possible that instead of tunneling through existing multi-hop routes, the malicious nodes can use a long-range directional wireless link or a wired link between them. Such a link gives the attackers an unfair advantage towards occurring on the shortest delay route between a source and destination. This has been referred to as the wormhole attack in recent literature [10], [11]. However, if the malicious nodes truly lie on the shortest delay path, it could be argued that the selection of this path is not a subversion of the routing protocol. A mechanism for defending against wormhole attacks is presented in [11].

B. Attacks Using Impersonation

Spoofing occurs when a node misrepresents its identity in the network, such as by altering its MAC or IP address in outgoing packets, and is readily combined with other attacks, such as those based on modification. The advantage
of spoofing is that the attack cannot be traced back to the malicious node.

C. Attacks Using Fabrication

Fabrication attacks involve the generation of false routing messages. Such attacks can be difficult to verify as invalid constructs, especially in the case of fabricated error messages that claim a neighbor cannot be contacted.

1) Falsifying Route Errors in AntHocNet: In AntHocNet, if the destination node or an intermediate node along an active path moves, the node upstream of the link break broadcasts a route error message to all active upstream neighbors. This message causes the corresponding route to be invalidated in all upstream nodes. A denial-of-service attack can be launched by continually sending route error messages indicating a broken link on the route, thereby preventing the source from communicating with the destination.

2) Route Cache Poisoning in AntHocNet: In AntHocNet, a node overhearing any packet may add the routing information contained in that packet's header to its own route cache, even if that node is not on the path from source to destination. An attacker could easily exploit this method of learning routes and poison route caches by transmitting packets containing invalid routes in their headers.

III. Security In MANETS Using ACO

A. Route Maintenance

SACOM is an on-demand protocol, which uses Certified HELLO packets for identification of its neighbors. These are short broadcast every T_h seconds by the nodes. If a node receives a hello message from a new node n, it will add n as a destination in its routing table. After that it expects to receive a hello from n every T_{hello} seconds. After missing a certain number of expected hello packets, n will be removed. Using these messages, nodes know about their immediate neighbors and have pheromone information about them in their routing table. So when an ant arrives in a neighbor of the destination, it can go straight to its goal. All HELLO messages must be signed.

$$A \rightarrow *: [\mathrm{HELLO}, \mathrm{IP}_A]K_{A-}, \mathrm{CT}_A \qquad (1)$$

This message is broadcasted by A to it s neighbors so that, their neighbors can insert A in their routing table. Hello messages also serve another purpose: they allow detecting broken links. This allows nodes to clean up stale pheromone entries from their routing tables. It is extremely difficult to detect when HELLO messages are fabricated for links that are truly active and not broken. However, the signature on the message prevents impersonation and enables non-repudiation. A node that transmits a large number of HELLO messages, whether the HELLO messages are valid or fabricated, should be avoided.

B. Certification of Authorized Nodes

SACOM uses cryptographic certificates to bring authentication, message-integrity and non-repudiation to the route discovery process. Therefore SACOM requires the use of a trusted certificate server CSer, whose public key is known to all valid nodes (or multiple servers may be used [14]). Nodes use these certificates to authenticate themselves to other nodes during the exchange of routing messages. The use of public keys and certificates is common in many secure ad hoc routing protocols, but most assume the existence of such information without any explicit description of how it is transmitted. While SACOM may appear more expensive, it is in part because we account for the distribution of the cryptographic keying material.

In our algorithm, keys are a priori generated and exchanged through an existing, perhaps out-of-band, relationship between CSer and each node. Before entering the ad hoc network, each node must request a certificate from CSer. Each node receives exactly one certificate after securely authenticating its identity to CSer. A node A receives a certificate from CSer as follows:

$$CSer \rightarrow A: CT_A = [IP_A, K_{A+}, toc, exp]K_{T-} \quad (2)$$

The certificate contains the IP address of A (IP_A), the public key of A (K_{A+}), a timestamp toc (time of creation) of when the certificate was created, and a time exp at which the certificate expires. Table I summarizes our notation. These variables are concatenated and signed by CSer. All nodes must maintain fresh certificates with the trusted server.

C. *Authenticated Route Discovery – Forward Ant*

The goal of end-to-end authentication is for the source to verify that the intended destination was reached. The source trusts the destination to select the return path. The source node A, begins route instantiation to destination X by broadcasting the Forward Ant to its neighbors:

$$A \rightarrow *: [FA, IP_X, S_A]K_{A-}, CT_A \quad (3)$$

The Forward Ant includes a Ant identifier ("FA"), the IP address of the destination (IP_X), A's certificate (CT_A) and a sequence number S_A, all signed with A's private key. Note that the FA is only signed by the source and not encrypted, so the contents can be viewed publicly. The purpose of the S_A is to uniquely identify an FA coming from a source. Each time A performs route discovery, it monotonically increases the S_A. The S_A is 5 bytes in size, and is thus large enough that it will not need to be recycled within the lifetime of the network. Note that a hop count is not

TABLE I: NOTATIONS USED IN THE ALGORITHM

Symbol	Definition
K_{A-}	*Private Key of A*
[data] K_{A-}	*Data Signed by A*
CT_A	*Certificate of A*
exp	*Certificate Expiration Time*
S_A	*Sequence Number*
IP_A	*IP Address of A*
FA	*Forward Ant identifier*
BA	*Backward ant Identifier*
toc	*Time of Creation*

included with the message. When a node receives an FA, it sets up a reverse path back to the source by recording the neighbor from which it received the ant. This is in anticipation of eventually receiving a reply message that it will need to forward back to the source (Backward Ant). The receiving node uses A's public key, which it extracts from A's certificate, to validate the signature and verify that A's certificate has not expired. The receiving node also checks the (IP_A, S_A) tuple to verify that it has not already processed this FA; nodes do not forward messages with already-seen tuples. The receiving node signs the contents of the message, appends its own certificate, and forward broadcasts the message to each of its neighbors. The signature prevents spoofing attacks that may alter the route or form loops. Let H be a neighbor that has received from A the forward ant, which it subsequently forwarded.

$$H \rightarrow *: [[FA, IP_X, S_A]K_{A-}]K_{H-}, CT_A, CT_H \quad (4)$$

```
Forward(α)
{
Use CTA  to get KA+
Decrypt(Data, KA+);

If(α= = X)
{
//Destination is Reached
Backward();
}
else
{
Encrypt (Data, Kα+);
Append(Data, Certα);
broad();
}
}
broad()
{
For allα∈β[ neighbors ]
Broadcast forward ant
}
```

Fig 1 Algorithm for Forward Ant

Upon receiving the forward ant, H's neighbor R validates the signatures for both A, the FA, and H, the neighbor it received the forward ant from, using the certificates in the forward ant. R then removes H's certificate and signature, records H as its predecessor, signs the contents of the message originally broadcast by A and appends its own certificate. R then rebroadcasts the forward ant. Each intermediate node along the path repeats the same steps as R.

D. *Authenticated Route Setup – Backward Ant*

After receiving the forward ant, the destination unicasts a Backward Ant packet back along the reverse path to the source. Let the first node that receives the Backward Ant sent by Xbe node M

$$X \rightarrow M: [BA, IP_A, S_A]K_{X-}, CT_X \quad (5)$$

The Backward Ant includes a packet type identifier ("BA"), the IP address of (IP_A), the certificate belonging to X (CT_X) and the sequence number sent by A. Nodes that receive the Backward ant forwards the packet back to the predecessor from which they received the original Forward ant. Each node along the reverse path back to the source signs the Backward Ant and appends its own certificate before forwarding the Backward Ant to the next hop. Let M's next hop to the source be node L

$$M \rightarrow L: [[BA, IP_A, S_A]K_{X-}]K_{M-}, CT_X, CT_M \quad (6)$$

```
Backward()
{
foreach(hop in hopstack)
{
```

```
Decrypt(Data, K α+);

Check for Validity(Data);

Encrypt (Data, K α-);

Append(Data, Cert α );
}
}
```

Fig 2 Algorithm for Backward Ant

Lvalidates M's signature on the received message, removes the signature and certificate, then signs the contents of the message and appends its own certificate before unicasting the REP to the next node. Each node checks the sequence number and signature of the previous hop as the REP is returned to the source. This avoids attacks where malicious nodes instantiate routes by impersonation and re-play of X's message. When the source receives the backward ant, it verifies the destination's signature and the sequence number returned by the destination.

E. Add-ons

Many protocols need to commit to a sequence of random values. For this purpose, we repeatedly use a one way hash function to generate a one-way chain. One way chains are a widely-used cryptographic primitive. One of the first uses of one-way chains was for onetime passwords by Lamport [15]. Haller later used the same approach for the S/KEY one-time password system [16]. One-way chains are also used in many other applications. Figure 3 shows the one-way chain construction. To generate a chain of length *l* we randomly pick the last element of the chain S_l. We generate the chain by repeatedly applying a one-way function F. Finally, S_0 is a commitment to the entire one-way chain, and we can verify any element of the chain through S_0, e.g. to verify that element S_i is indeed the element with index i of the hash chain, we check that $F^i(S_i) = S_0$. More generally, S_i commits to S_j if i < j (to verify that S_j is part of the chain if we know that S_i is the ith element of the chain, we check that $F^{j-i}(S_j) = S_i$). We reveal the elements of the chain in this order $S_0, S_1, \ldots, S_{l-1}, S_l$. How can we store this chain? We can either create it all at once or store each element of the chain, or we can just store s` and compute any other element on demand. In practice, a hybrid approach helps to reduce storage with a small recomputation penalty. Jakobsson [17], and Coppersmith and Jakobsson [18] propose a storage efficient mechanism for one-way chains: a one-way chain with N elements only requires log(N) storage and log(N) computation to access an element.

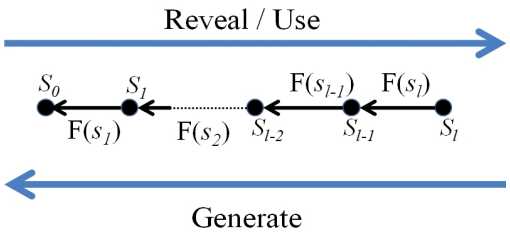

Fig 3 One way Hash chain example

We have specified the use of public certificates here; it is clear that intermediary can easily agree upon and exchange session keys using the certificates that authenticate their participation in route creation. Two nodes can easily share a symmetric key generated with their own private key and the public key of the other. A session key can last the duration of their juxtaposition and can be a symmetric key, KBC to reduce processing costs; equivalently, juxtaposed peers can create low-cost hash chains between themselves for authentication of future messages. Using these optimizations would decrease computational overhead and power consumption. However, even if these optimizations are used, we require that sources and destinations must include full public-key signatures for end-to-end route discovery and setup messages.

F. Key Revocation

In some environments with strict security criteria, the required certificate revocation mechanism must be very reliable and expensive. Due to the desired low overhead in wireless networks and the lower standards of security sought in the managed-open and open environments, a best-effort immediate revocation service can be provided that is backed up by the use of limited-time certificates. In the event that a certificate needs to be revoked, the trusted certificate server CSer, sends a broadcast message to the ad hoc group that announces the revocation. Calling the revoked certificate CT_r, the transmission appears as:

$$\mathrm{CSer} \rightarrow *: [\mathrm{revoke}, \mathrm{CT}_r]K_{\mathrm{CSer}-} \qquad (7)$$

Any node receiving this message re-broadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the un trusted node. This method is not failsafe. In some cases, the un trusted node that is having its certificate revoked may be the sole connection between two parts of the ad hoc network. In this case, the un trusted node may not forward the notice of revocation for its certificate, resulting in a partition of the network, that lasts until the un trusted node is no longer the sole connection between the two partitions. At the time that the revoked certificate should have expired, the un trusted node is unable to renew the certificate, and routing across that node ceases. Additionally, to detect this situation and to hasten the propagation of revocation notices, when a node meets a new neighbor, it can exchange a summary of its revocation notices with that neighbor; if these summaries do not match, the actual signed notices can be forwarded and re-broadcasted to restart propagation of the notice.

IV. Security Analysis

In this section, we provide a security analysis of SACOM by evaluating its robustness in the presence of the attacks introduced in Section III. As mentioned earlier, we do not consider denial-of-service attacks based on non-cooperation or aggressive participation, which are possible against all ad hoc routing protocols.

Unauthorized participation: Since all SACOM packets must be signed, a node cannot participate in routing without authorization from the trusted certificate server CSer. This access control therefore rests in the security of the trusted authority, the authorization mechanisms employed by the trusted authority, the strength of the issued certificates, and the revocation mechanism. Although we do not detail these functions explicitly, except for certificate revocation, they have been extensively studied by others. In practice, many single-hop 802.11 deployments already use VPN certificates; this is the case on the UMass campus.Mechanisms for authenticating users to a trusted certificate Authority are numerous; a significant list is provided by Schneier [30]. The trusted authority is also a single point of failure and attack; however, multiple redundant authorities may be used (e.g., as by Zhou and Haas [14]).

Spoofed Route Signaling: Forward Ant contain the certificate of the source node and are signed with the source's private key. Similarly, Backward ant include thedestination node' s certificate and signature, ensuring that only the destination can respond to route discovery. This prevents impersonation attacks where either the source or destination nodes is spoofed.

Fabricated Routing Messages: Since all routing messages must include the sending node's certificate and signature, SACOM ensures non-repudiation and prevents spoofing and unauthorized participation in routing. SACOM does not prevent fabrication of routing messages, but it does offer a deterrent by ensuring non-repudiation. A node that continues to inject false messages into the network may be excluded from future route computation.

Alteration of Routing Messages: SACOM specifies that all fields of forward ant and backward ant remain unchanged between source and destination. Since both packet types are signed by the initiating node and the use of Hash chain, any alterations in transit would be detected, and the altered packet would be subsequently discarded. Repeated instances of altering packets could cause other nodes to exclude the errant node from routing. Thus, modification attacks are prevented.

Securing Shortest Paths: The main advantage of integrating ant colony optimization and routing in MANETs is to identify the shortest route to the destination. In no way the SACOM changes this principle. SACOM identifies the shortest path using the hop count value but this advantage is not provided by ARAN[13]. ARAN assumes the route from which the first packet arrives is the shortest route. SACOM uses the Hash chains to identify the changes that takes place in the packet and takes necessary action.

Forwarding Attacks: End-to-end integrity can be ensured by the hash one way chains.

Denial-of-Service Attacks: Denial-of-service attacks can be conducted by nodes with or without valid SACOM certificates. In the certificateless case, all possible attacks are limited to the attacker' s immediate neighbors because unsigned route requests are dropped. There are more severe attacks available at the MAC and physical layer than ARAN provides. Nodes with valid certificates can conduct effective attacks, however, by sending many unnecessary route requests. Because these are broadcast and forwarded across the network, an attacker can cause widespread congestion and power-loss to all nodes in the network. Because it is difficult to infer the node' s intent at the network level, it can be hard to differentiate between legitimate and malicious RREQs.

V. Conclusion

This paper has presented the design and evaluation of SACOM a new ad hoc network routing protocol using ant colony optimization that provides security against attackers, and relies only on efficient asymmetric cryptography. With development in computing environments, the services based on ad hoc networks have been increased. However, wireless ad hoc networks are vulnerable to various attacks due to the physical characteristic of both the environment and the nodes. Most existing security mechanisms for MANETs thus far involve the heavy use of public-key certificates. In this regard, we believe that the findings of this paper would have much influence on the research paradigm of the whole community and stimulate many other fresh research outcomes. As our future work, we will try to integrate the security issues with other swarm intelligence method to increase the efficiency of the security as well as routing.

Reference:

[1] M. G. Hinchey, R. Sterritt, and C. Rouff, "Swarms and Swarm Intelligence, " IEEE Computer, Vo. 40, No. 4, pp. 111-1 13, April 2007.

[2] K. M. Sim and W. H. Sun, "Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions, " IEEE Trans. on Systems, Man, and Cybernetics, Vol. 33, No, 5, Sep. 2003, pp. 560-572.

[3] M. Guine, , U. Sorges, and I. Bouazzi, "ARA-the ant-colony based routing algorithm for MANETs, " in Proc. of IWAHN 2002, pp. 79-85, August 2002.

[4] H. Matsuo and K. Mori, "Accelerated Ants Routing in Dynamic Networks, " 2nd International Conf. on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, pp. 333-339, 2001.

[5] S. Marwaha, C. K. Tham, and D. Srinivasan, "Mobile agents based routing protocol for mobile ad hoc networks, " in Proc. ofIEEE ICON, pp. 27-30, August 2002.

[6] J. S. Baras and H. Mehta, "A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks, " in Proc. of WiOptO3, 2003.

[7] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks, " Tech. Rep. No. IDSIA-27-04-2004, IDSIA/USI-SUPSI, Sep.2004.

[8] Gianni Di Caro, Marco Dorigo, “AntNet: Distributed Stigmergetic Control for Communications Networks” Journal of Artificial Intelligence Research 9 (1998) 317-365.

[9] T. Maekawa, et. al. "An Ant-based Routing Protocol using Unidirectional Links for Heterogeneous Mobile Ad-Hoc Networks, " in Proc. ofICWMC '06, pp. 43 - 43, July 2006. ISBN

[10] Y. Hu, A. Perrig, and D. Johnson, “Ariadne: A Secure On-DemandRouting Protocol for Ad Hoc Networks, ” in Proc. MobiCom, Sep. 2002.

[11] Y. Hu, A. Perrig, and D. Johnson, “Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks, ” in Proc. Infocom, Apr. 2003.

[12] Sun Choi, Doo-young Kim, Do-hyeon Lee, Jae-il Jung “WAP:Wormhole Attack Prevention Algorithm in Mobile Ad Hoc Networks”, in IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2008

[13] Kimaya Sanzgiri, Daniel LaFlamme and Bridget Dahill, “Authenticated routing for Ad hoc Networks” IEEE Journal on Selected Areas in Communications, Volume 23, Issue 3, March 2005 Page(s): 598 - 610.

[14] L. Zhou and Z. J. Haas, “Securing Ad Hoc Networks, ” IEEE Network, vol. 13, no. 6, pp. 24–30, 1999.

[15] L. Lamport. “Password authentication with insecure communication.” Communications of the ACM, 24(11):770–772, November 1981.

[16] N. Haller. "The S/Key one-time password system." In Proceedings of the Symposium on Network and Distributed Systems Security, pages 151–157. Internet Society, February 1994.
[17] M. Jakobsson. "Fractal hash sequence representationand traversal". In Proceedings of the 2002 IEEE International Symposium on Information Theory (ISIT '02), pages 437–444, July 2002.
[18] D. Coppersmith and M. Jakobsson." Almost optimalhash sequence traversal." In Proceedings of the Fourth Conference on Financial Cryptography (FC '02), Lecture Notes in Computer Science, 2002.
[19] I. D. Chakeres and E. M. Belding-Royer, "The Utility of Hello Messages for Determining Link Connectivity, " in Proc. WPMC, Oct. 2002.
[20] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GlomoSim: A Scalable Network Simulation Environment, " UCLA, Tech. Rep. CSD Technical Report #990027, 1997.
[21] M. Jakobsson and A. Juels, "Proofs of Work and Breadpudding Protocols, " in Proc. Communications and Multimedia Security, Sep. 1999.
[22] [22]. S. Gwalani, K. Srinivasan, G. Vigna, E. Belding-Royer, and R. Kemmerer, "An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks, " in Proc. ACSAC, Dec. 2004
[23] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks, " in Proc. MobiCom, Aug. 2000.
[24] M. G. Zapata and N. Asokan, "Securing Ad hoc Routing Protocols, " in Proc. ACM WiSe, Sep. 2002.
[25] Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks, " in Proc. WMCSA, Jun. 2002.
[26] M. Just, E. Kranakis, and T. Wan, "Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks, " in Proc. ADHOC-NOW, Oct. 2003.
[27] L. Butty´an and J.-P. Hubaux. Report on a working session on security in wireless ad hoc networks. ACM SIGMOBILE Mobile Computing and Communications Review, 7(1):74– 94, Jan 2003.
[28] J. Zhen and S. Srinivas. Preventing replay attacks for secure routing in ad hoc networks. In ADHOC-NOW, LNCS 2865, pages 140–150, 2003.
[29] I. Khalil, S. Bagchi, and N. B. Shroff. LITEWORP: A lightweight countermeasure for the wormhole attack in multihop wireless networks. In Dependable Systems and Networks (DSN), pages 612–621, Jun 2005.
[30] I. Khalil, S. Bagchi, and N. B. Shroff. MOBIWORP: Mitigation of the wormhole attack in mobile multihop wireless networks. Securecomm and Workshops 2006, pages 1–12, Aug 2006.

Ramkumar K.R. Born in Madurai, Tamil Nadu, India at 1977, did the B.E. computer Science in the year of 1999 at Madurai Kamaraj University, Madurai, Tamil Nadu, India and then completed M.E. Computer Science in the year of 2007 at Sri Venkateswara College of Engineering, Anna University, India.He worked as a Lecturer at Sri Venkateswara College of Engineering, Chennai, Tamil Nadu, India for seven years and now working as a Senior Lecturer in the same college. He has published many papers in international journals.His research interests are Swarm Intelligence, Mobile Ad Hoc Networks and Security issues in routing algorithms.

M. Ravichandran earned his Master's degree in Optical Communication from college of Engineering, Anna University, Chennai in 2002. He acquired his Bachelor's degree in Electronics and Communication Engineering, from Sri Venkateswara College of Engineering, Sriperumbudur in 2000.
He is a lecturer of Computer Science and Engineering at Sri Venkateswara College of Engineering. He has over 6 years of teaching experience where he has guided many Undergraduate and Post graduate research projects. He has published many research papers and journals.
His research areas include Mobile Computing. High Speed Communication Networks, intelligent System and Digital Design. He is a Member of ISTE and IETE.

Hemachandar. N Born on December 9 1987, in Chennai, India. Is a under graduate student in the Department of Computer Science and Engineering Department at the Sri Venkateswara College of Engineering. His research interests are in Ad Hoc Mobile Networks and Swarm intelligence. He has published many research papers and journals.

Manojprasadh. D Born on November 8 1987 in Chennai, India. Is a under graduate student in the Department of Computer Science and Engineering Department at the Sri Venkateswara College of Engineering. His research interests are in Ad Hoc Mobile Networks and Swarm intelligence. He has published many research papers and journals He is a member of IEEE.

GaneshKumar. M Born on June 18, 1988, in Chennai, India. Is a under graduate student in the Department of Computer Science and Engineering Department at the Sri Venkateswara College of Engineering. His research interests are in Networking, Peer to Peer Networks, Ad Hoc Mobile Networks and Swarm intelligence. He has published many research papers and journals. He is a member if the IACSIT an International scientific association.