Secure Prophet Address Allocation for Mobile Ad-hoc Networks

Hongbo Zhou
Dept. of Computer Science
Slippery Rock University
Slippery Rock, PA 16057
USA
hongbo.zhou@sru.edu

Abstract

A mobile node in a MANET must be assigned with a free IP address before it may participate in unicast communications. This is a fundamental and difficult problem in the practical application of any MANET. There have been several solutions proposed, among which prophet address allocation outperforms others in terms of communication overhead, latency, and scalability. However, none of the approaches can survive attacks in an insecure environment. Based on studies of insecure scenarios, attack schemes, and our previous work, a secure autoconfiguration algorithm, namely secure prophet address allocation, is proposed in the paper. The proposed approach is able to maintain uniqueness of address assignment in the presence of IP spoofing attacks, "state pollution" attacks, and Sybil attacks. The invulnerability of the scheme is supported by both theoretical analysis and simulation results.

1. Introduction

A Mobile Ad-hoc Network (MANET) is a temporary wireless network composed of mobile nodes, in which an infrastructure is absent. Due to the abundance of mobile devices, the speed and convenience of deployment, and the independence from a network infrastructure, a MANET will find many applications in military uses, search-and-rescue operations, meeting rooms, and "smart transportations". In such an IP-based network, IP address assignment to mobile devices is one of the most important network configuration parameters.

For small scale closed MANETs, it may be easy and efficient to allocate free IP addresses manually. However, the procedure becomes difficult and impractical for an open system where mobile nodes are free to join and leave. The automatic configuration of IP addresses (autoconfiguration) for MANETs is more difficult than that in hardwired networks because of instability of mobile nodes, low bandwidth of wireless links, openness of the MANET, and lack of centralized administration. Therefore, additional overhead occurs to avoid address conflicts in comparison to the protocols for hardwired networks, such as DHCP [1].

Several autoconfiguration algorithms for MANETs have been proposed ([2] – [6]), among which prophet address allocation outperforms the others in terms of communication overhead (i.e., the total number of control packets generated to ensure the uniqueness of a new IP address) and latency (i.e., the time needed to generate a unique IP address). Thus, prophet address allocation has better scalability that the other schemes.

However, all of these approaches are based upon the assumption that the application scenario is secure. None of these approaches takes any security mechanism into consideration. When applied in real situations in which malicious nodes may exist, these autoconfiguration schemes will fail to function properly: either no new nodes will be allowed to join the MANET or there will be duplicate addresses assigned in the network.

Security mechanisms are extremely important for MANET due to the following reasons:

(1) The wireless link between two nodes is a broadcast channel, so the communication is vulnerable to eavesdropping;

(2) The assumption underlying the MANET is that all the nodes (or most nodes) cooperate to function properly. A malicious node can undermine routing fabrics and other services passively (by dropping the packets that need to be forwarded) or actively (by injecting false information into the network or altering the packets in transit);

(3) It is more difficult to identify the source of a message in the MANET than in the hardwired network because of the absence of an infrastructure;

In addition to the above-mentioned reasons, mobility adds more complexity to the design of secure protocols and applications. The most common and

978-0-7695-3354-4/08 $25.00 © 2008 IEEE
DOI 10.1109/NPC.2008.67

simplest situation is that a node joins a dense MANET in which malicious nodes may exist, as illustrated in Fig. 1 (N represents a new node, M represents a malicious node). Even there is only one malicious node in some network and we may adopt "majority votes" policy, the policy can be easily circumvented with Sybil attacks [7] when the malicious node impersonates many non-existent nodes. There are other situations, such as division and merger of the MANET, which can be regarded as extensions of this scenario. For example, when two MANETs merge, we can let the nodes from one MANET join the other one by one. Thus, we focus on this scenario in the paper.

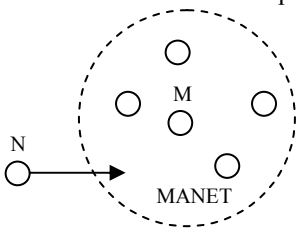

Figure 1. New nodes join a dense MANET

Most research effort on security in the MANET has been focused on secure routing protocols ([8] — [10]), and key management ([11] [12]), whereas the study on secure autoconfiguration in the MANET has not been explored. Note that the schemes for secure routing protocols may not be appropriate in secure autoconfiguration because the new node does not have a valid IP address before completion of autoconfiguration. It has to rely on one-hop or multi-hop broadcast, which is unrelated to unicast routing.

The paper is structured as follows. Section 2 gives a brief description about all the pre-existing autoconfiguration schemes, including the prophet address allocation. The misbehaviors of all the autoconfiguration schemes in the presence of different attacks are analyzed in Section 3. Based on the authors' previous work, an extended algorithm, namely secure prophet address allocation, is proposed in Section 4. The proposed autoconfiguration algorithm is able to survive IP spoofing attacks, "state pollution" attacks, and Sybil attacks, which is demonstrated in Section 4, as well. The analysis is supported by the simulation results in Section 5. Section 6 suggests future work and concludes the paper.

2. Autoconfiguration algorithms

There have been several autoconfiguration schemes proposed for MANETs, which can be divided into four groups.

2.1. Conflict-detection Allocation

The conflict-detection allocation adopts a "trial and error" policy to find a free IP address for a new node in the MANET. The new node chooses a random IP address tentatively, makes an announcement with flooding of a conflict detection message, and waits for approval from all the other members in the MANET. If the address that it chooses is already used by another member, it is going to receive a veto from that member and choose an address again.

2.2. Conflict-free Allocation

In conflict-free allocation, every node in the MANET maintains a disjoint address pool. When a new node joins the network, one of its neighbors divides its address pool into halves and gives one half to the new node. Thus, every node allocates unique addresses to new nodes. An example of conflict-free allocation is proposed in [3].

2.3. Best-effort Allocation

In this scheme [4], every node maintains a global allocation state to track down which IP addresses have been allocated and which IP addresses are still free. When a new node joins the network, one neighbor chooses a spare address for the new node. However, since the same spare address can be chosen for two new nodes joining at different parts of the network simultaneously, the scheme still requires conflict detection mechanism to ensure uniqueness of address allocation.

2.4. Prophet Address Allocation

Prophet address allocation ([5] [6]) utilizes a partition function f(n) as a sequence generator to assign a unique IP address to a mobile node. The partition function f(n) is a stateful function with the initial state called the seed. Different seeds lead to different integer sequences. These sequences satisfy the following properties:

(1) The interval between two occurrences of the same number in a sequence is extremely long;

(2) The probability of more than one occurrence of the same number in a limited number of different sequences initiated by different seeds during some interval is extremely low.

Because the integers in the sequences can be computed locally, which includes the addresses that have been allocated and are going to be allocated, conflicts can be avoided without broadcasts of conflict detection packets.

To be more specific, the procedures of prophet address allocation work as follows:

(1) The first node in the MANET, say node A, chooses a random number as its IP address and uses a random state value or a default state value as the seed for its f(n);

(2) When a new node, say node B, approaches A and asks A for a free IP address, A uses f(n) to obtain another integer, say n_2, and a state value, and provides them to B. Node A updates its state accordingly;

(3) Node B uses n_2 generated by A as its IP address and the state value obtained from node A as the seed for its f(n);

(4) Now node B is able to assign IP addresses to other new nodes, as node A.

The core issue of prophet address allocation is the design of the partition function f(n). Based on the fundamental theorem in number theory, every positive integer may be expressed uniquely as a product of primes. Apart from the rearrangement of terms, the canonical form of a positive number n is

$$n = \prod_{i=1}^{k} p_i^{e_i}$$

where the primes p_i satisfy $p_1 < p_2 < \ldots < p_k$ and the exponentials are non-negative integers. The partition function uses different exponential arrays to generate different numbers in the integer sequences.

The partition function can be illustrated with an example. Suppose $k = 4$. The first node obtains a random address of a and an initial state of (0, 0, 0, 0). Fig. 2 shows the procedure of generating new states and updating old states for the other three nodes. Suppose that a node is represented by (address, (e_1, e_2, e_3, e_4)), with address = $(a + 2^{e1}3^{e2}5^{e3}7^{e4}) \bmod P + 1$ where P is the largest prime number less than the address range. The parameters sent from the allocator to the new node include: (1) seed value (a); (2) the exponential array (e_1, e_2, e_3, e_4); (3) the index of increasing exponential (the place of the underlined element in the 4-tuple). The rules of state generation and update during the allocation are: (1) the increasing exponential (the underlined element in the 4-tuple) of the allocator is increased by 1; (2) the state of the new node is copied from the allocator, but the index of the increasing exponential is increased by 1 (as the underline moves to the right).

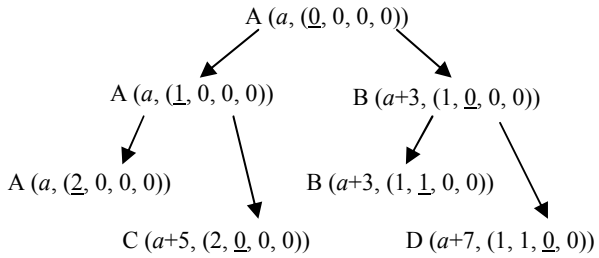

Figure 2. An example of the partition function in prophet address allocation

3. Attacks on autoconfiguration

All existing autoconfiguration algorithms for MANETs assume secure environments. They cannot function properly in the case of attacks from malicious nodes. This section classifies the attacks and demonstrates the algorithms' misbehaviors.

3.1. IP Spoofing Attacks

In a conflict-detection allocation, the new node chooses a random address (say x) and broadcasts a conflict detection packet throughout the MANET. Any veto from a node will prevent it from using this address. If the malicious node always impersonates a member that has occupied the same IP address and keeps replying with vetoes, it is called an IP spoofing attack.

Any allocation algorithms that utilize the conflict detection mechanism cannot defeat the IP spoofing attack, including best-effort allocation. However, this attack is easy to launch because the malicious node can construct a whole IP packet including the IP header, so it can put a fake source IP address in the response. In Linux, the programmer can even build up a data link layer packet, which may contain a fake source MAC address as well.

IP spoofing attacks on prophet address allocation takes another form. Because the seed value (a) is known to all the nodes, a malicious node (say, node M) can forge a seemingly reasonable exponential array that in fact belongs to another node (say, node G), so nodes M and G will generate duplicate address to new nodes separately.

IP spoofing attacks are difficult to detect and prevent even with the aid of secured switches and routers in hardwired networks, let alone in MANETs where an infrastructure is absent. There are some methods proposed to detect IP spoofing attacks. However, these methods can be easily circumvented.

One method that seems promising is a cryptographic method such as using a digital signature.

For example, the bindings of the IP addresses and public keys are stored in the Certificate Authority (CA) in the MANET and the public keys are certified by the CA, like the scheme used in [11]. However, this scheme may not fit the scenario where nodes are free to join and leave and have dynamic assigned IP addresses. The public-key management proposed in [12] is not secure either because the malicious node can generate a public/private key pair for the fake IP address and then certify the public key with its own.

Actually, the CA and the digital signature scheme cannot be utilized in secure autoconfiguration because the CA itself needs a valid IP address first. Thus, to obtain a valid address for the CA, we need another CA to validate the control messages signed by the allocators, which forms a circle. In some special scenarios we can specify that the first node in the MANET be the CA server and let it choose a random address. Because it is the first node, no conflict detection is necessary, and thus there will be no IP spoofing attacks against it. However, this is not the general case, and a centralized CA is not desirable in the MANET. We can also place CA server(s) in the MANET and specify the fixed address(es), which is beyond our topic on secure autoconfiguration.

Even if there is a CA in the MANET, the allocation algorithm with conflict detection mechanism cannot prevent IP spoofing attacks, which can be illustrated with the following example. Suppose a new node requests the address of x in the conflict detection message. The new node cannot use x in communications unless it is sure that x is not occupied by others. Once the malicious node receives the conflict detection message, it immediately changes its address to x temporarily, generates a public/private key pair and registers the public key with address x at the CA, and then sends the veto signed with the public key to the new node.

The reason that the CA is difficult to be applied in secure autoconfiguration is that it regards the IP address as a node's identifier and binds the public/private key pair with the IP address. However, this kind of identifiers is allocated dynamically in the scenario of autoconfiguration. Moreover, there seem no other versatile candidates for identifiers. For example, the physical address is not appropriate due to the following reasons: (1) most wireless NICs are detachable so the user may change the wireless NIC during the session of the MANET; (2) the user is able to modify the MAC address of the wireless NIC; (3) some wireless NICs are manufactured from small companies that may not allocate the addresses according to the IEEE standard, so the uniqueness of MAC addresses cannot be guaranteed. Although the user is most unlikely to change the CPU during the middle of communications, the built-in identifiers in some CPUs are not proper either, because in a heterogeneous MANET, not all CPUs have identifiers. Moreover, the built-in identifiers are hidden from outsiders. Thus, it is still easy for a malicious node to forge the built-in identifiers.

3.2. "State Pollution" Attacks

If a malicious node gives incorrect parameters in the reply, it is called the "state pollution" attack. For example, in best-effort allocation, a malicious allocator can always give the new node an occupied address, which leads to repeated broadcasts of conflict detection messages throughout the MANET and the rejection of the new node. This attack is difficult to be differentiated from the case when the global allocation states are not properly synchronized among all the mobile nodes because of the unreliable broadcast transmission. In conflict-free allocation, a malicious allocator can give a non-disjoint address pool to the new node, which affects the allocation to prospective new members and causes a huge problem for the MANET. In prophet address allocation, if the malicious node does not update its state, a previously allocated address will be re-assigned to the new node.

3.3. Sybil Attacks

If a malicious node impersonates some non-existent nodes, it will appear that several malicious nodes conspire together, which is called a Sybil attack [7]. This attack aims at network services when cooperation is necessary and affects autoconfiguration as well. However, there is no practical way to defeat Sybil attacks.

Table 1 summarizes vulnerabilities of the four autoconfiguration schemes.

TABLE 1 Summary of vulnerabilities

	IP spoofing attacks	"State pollution" attacks	Sybil attacks
Conflict-detection allocation	Yes	N/A	N/A
Conflict-free allocation	Yes	Yes	N/A
Best-effort allocation	Yes	Yes	Yes
Prophet allocation	Yes	Yes	Yes

4. Secure prophet address allocation

In this section, we describe the extension added to the original scheme.

4.1. Verification of Seed Value

In a secure environment, a new node can immediately configure itself with the parameters contained in the first response. However, in an insecure environment, this response may come from a malicious node, so the new node must wait for a period to collect as many responses as its neighbors to validate them.

With all the responses, the new node needs to find the correct seed value (a) first. Because the seed value is chosen by the first node in the network and kept constant during allocation, the seed value in all the replies should be the same in secure environment. If a malicious neighbor puts a different seed value, it seems to the new node that it is on the edge of two neighboring MANETs. It chooses either one to join. If it cannot talk to anyone in the MANET forged by the malicious neighbor, it then leaves the forged MANET and joins the other. Thus, the authenticity of seed value is not important. The uniqueness of addresses within the MANET relies on uniqueness of exponential arrays, which is enforced with the scheme in the following section.

4.2. Extension

In the original prophet allocation, there is an implicit assumption that all the nodes update their internal states correctly after allocation. In an insecure environment, a malicious node can refrain from updating its state or forge a false state, which renders difficulties in determining the authenticity of the other parameters in the reply. However, we can change the implicit assumption to an explicit rule by broadcasting an announcement after the allocation. If the new node follows the rule in computing its address, the scheme guarantees that no duplicate address is generated.

Thus, we require that the reply contain the following parameters in addition to the seed value and the index of the increasing exponential:

(1) The initial exponential array, which is the exponential array that the allocator (new node's parent node) received from its own allocator (the new node's grandparent node, except for the first node in the MANET) and was used to compute its own address. This parameter is used to validate the relationship between the parameters and allocator's address in the reply.

(2) A new state variable: priority. The priority indicates the freshness of the state contained in the reply. The larger the number, the fresher the state. The new node always chooses the highest priority value among the replies, and then adds some random value to the priority in computing its address. After allocation, the new node broadcasts an ACK message containing the updated priority value as an announcement. On receipt of the ACK message, all the configured nodes update their priority values.

Because the priority keeps increasing and is synchronized by multi-hop broadcast that is unreliable, even a good node may miss an ACK message containing the up-to-date priority. To solve this problem, we let the new node determines the highest priority value without discarding the replies that contain lower priority value. For those replies, the new node can still verify and utilize other parameters. Other solutions include reliable broadcast protocols ([14] [15]), or extension of the watchdog scheme [16] to monitor the broadcast of ACK messages of the neighbors. For example, with the aid of the neighbor detection mechanism in the routing protocol, each node can compare the number of retransmission of the broadcast packet with the number of neighbors to make sure that the former exceeds a threshold percentage of the latter.

If the malicious node presents a priority value that is greater than the highest value in the network, all the good nodes will update theirs to the high priority value. The scheme still guarantees that unique addresses will be generated, with some states skipped.

If all the neighbors are malicious nodes, they can conspire to provide an obsolete priority value. Since the new node is also going to choose a random value in computing its address, the malicious node cannot predict that number, thus they cannot allocate duplicate exponential array.

In summary, the reply contains the following parameters:

(1) The seed value for the whole MANET (a);
(2) The index of the increasing exponential (c);
(3) The source address of the responder (x).
(4) The initial exponential array ($i[1..n]$);
(5) Priority (p).

The relationship among these state variables is

$x = f(a, i[1..n])$ Equation (1)

which means the allocator's source address can be computed with the seed value and the initial exponential array as input to the stateful partition function.

The new node chooses a random value (r) to calculate its address (y):

$y = f(a, e[1..n])$ where

$$e[j] = \begin{cases} i[j], j < c \\ p + r, j = c \\ i[j] = 0, j > c \end{cases} \quad \text{Equation (2)}$$

Although all these parameters may be forged, the highest priority value plus the random value *r* will ensure different exponential arrays during allocations, which leads to different addresses.

4.3. Protocol

The new protocol works as follows:

(1) When the node switches into the ad-hoc mode, it begins to broadcast state request messages periodically, and changes from the UN-INITIALIZED state to the WAITING state.

(2) The mobile node stays in the WAITING state and repeats state requests for less than or equal to m times;

(3) If it receives a reply during that time, it changes from the WAITING state to the COLLECTING state, and stays in the latter to collect as many replies as possible;

(4) When the COLLECTING state times out, the node validates all the replies that it has received: first, the new node determines the correct seed value (a) and priority (p) for the whole MANET. If there are different seed values, the new node chooses one randomly; if there are different priority values in the remaining replies, it chooses the highest value among all the replies, and then discards all the replies with different seed values. For the remaining replies, it first determines the relationship of parameters according to Equation (1), and discards all the replies that fail the test. For all the remaining replies, it determines the highest priority value, chooses a random reply, and a random value (r), then calculates its address according to Equation (2). In the end, it broadcasts an ACK message with the priority value plus r, and enters the CONFIGURED state.

(5) If there are no valid replies, it starts to broadcast state request messages again and returns to the WAITING state;

(6) If the node fails to receive any (valid) replies for m times, it chooses itself an IP address and NID randomly and a default state value as its initial state value, and changes to the CONFIGURED state;

(7) Within the CONFIGURED state, the mobile node repeats broadcasting HELLO messages, sends back replies on receipt of state request packets from other nodes, and updates its own state accordingly;

(8) When the configured node receives an ACK message with a larger priority, it updates its state according to the new priority value;

(9) When the mobile node ends its session in the MANET, it switches out of the ad-hoc mode and changes to the UN-INITIALIZED state.

The parameter m and the value for timeout could be preset heuristically and adjusted according to the parameters such as node density, bandwidth, and packet loss rate.

4.4 Invulnerability Analysis

The invulnerability of the secure prophet address allocation can be demonstrated with some typical examples, as illustrated in Fig. 3. In Fig. 3, both N_1, N_2, and N_3 are new nodes joining the MANET simultaneously, M is a malicious node, and G is a good node. We also suppose that the state includes a random seed value a and a 4-tuple, like the state used in Fig 2.

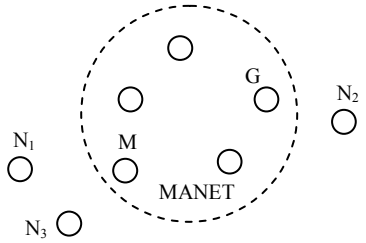

Figure 3. A simple MANET

In IP Spoofing attacks on Prophet Address Allocation, node M impersonates another node (say node G) and uses its state. Suppose the state of G is (2, 1, 0, 0), both N_1 and N_2 will get the same address: $(a+2^2 3^1 5^1) \bmod P + 1$, which is $a+61$ if we omit the modulus operation. However, in Secure Prophet Address Allocation, a new parameter, priority (*p*), together with a random value (*r*) are both utilized. If there is a small interval between N_1 and N_2 joining the network, they are going to receive different priority values. If they join at exactly the same time, they will choose different random values, so they are going to have different addresses generated.

In "state pollution" attacks on Prophet Address Allocation, node M does not update its state at all. Suppose the state of M is still (2, 1, 0, 0), both N_1 and N_3 will get the same address of $a+61$ (again, we omit the modulus operation). However, in Secure Prophet Address Allocation, even node M uses the same state in the two address assignment operations, nodes N_1 and N_3 will choose different random values in calculation.

In Sybil attacks on Prophet Address Allocation, node M impersonates many non-existent nodes. They can conspire to provide a false seed value a, which looks like a different MANET to the new node. The new node can go ahead joining the network and then later it can just leave the network when it finds out there is none to talk with. They can also conspire to

provide an obsolete priority value, which is defeated with the random value chosen by the new node.

5. Simulation experiments

According to our analysis, conflict-detection allocation and best-effort allocation will fail to function in the presence of IP spoofing attacks because there will be repetitious flooding of conflict detection messages throughout the MANET. Therefore, they are ignored in our simulation. To validate the survivability of secure prophet address allocation, we only need to compare it with the original prophet address allocation to see if it achieves uniqueness of IP address allocation in the presence of different kinds of attacks.

The simulation was done in ns-2 (version 2.31) with the CMU extension for ad hoc networks [17]. Statistics about the number of duplicate address pairs were collected to show that the secure prophet allocation is able to survive IP spoofing attacks, "state pollution" attacks, and Sybil attacks in insecure environments.

Both allocation schemes are compared in a simulation of 50 nodes moving around randomly in the area of 800 m × 800 m with the random waypoint mobility model. The size of the area is chosen to make the network densely connected. In the simulation, we choose k = 209. However, the simulation results show that at most 6 exponentials are used.

Table 2 shows the number of duplicate address pairs with IP spoofing attacks. Table 3 shows the number of duplicate address pairs with "state pollution" attacks and Sybil attacks. The number of malicious nodes varies during each run of simulation. In both simulations, no duplicate address is generated with secure prophet address allocation.

Table 2. The number of duplicate address pairs with IP spoofing attacks

Percentage of malicious nodes	Prophet address allocation	Secure prophet allocation
10%	1	0
20%	2	0
25%	1	0
33%	5	0
50%	1	0

Table 3. The number of duplicate address pairs with "state pollution" attacks and Sybil attacks

Percentage of malicious nodes	Prophet address allocation	Secure prophet allocation
10%	2	0
20%	6	0
25%	3	0
33%	16	0
50%	9	0

6. Conclusion

Secure autoconfiguration assures uniqueness of address allocation in the MANET in insecure environments, which is the first step towards secure MANETs in practical applications. However, most research effort has been focused on secure routing protocols, secure communications, and key management, whereas secure autoconfiguration has been neglected. Nevertheless, the latter is still difficult because we cannot simply combine cryptographic methods with pre-existing address allocation schemes due to the reasons in Section 3.

This paper is the first effort to propose a secure autoconfiguration for MANETs. Based on studies of insecure scenarios, categories of attack schemes, and our previous work, we extended our prophet address allocation so that it survives IP spoofing attacks, "state pollution" attacks, and Sybil attacks. Thus, new nodes will be able to join the MANET without being assigned duplicate addresses in insecure environments. In addition to the simplicity of the algorithm, secure prophet address allocation is especially suitable for the environments in which a CA does not exist or a pre-existing trust relationship among nodes cannot be built. Both theoretical analysis and simulations were conducted to demonstrate the survivability of the proposed scheme.

We are still studying the other kinds of attack patterns, the impact of packet loss rate on the secure allocation scheme, and more complicated simulations, which will be our future work.

7. References

[1] R. Droms, "Dynamic host configuration protocol," Network Working Group RFC 2131, March 1997

[2] C. Perkins, J. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun, "IP address autoconfiguration for ad hoc networks," draft-ietf-manet-autoconf-01.txt, November 2001 (work in progress)

[3] A. Misra, S. Das, A. McAuley, and S. K. Das, "Autoconfiguration, registration, and mobility management for pervasive computing," IEEE Personal Communication System Magazine, Vol. 8, pp. 24-31, August 2001

[4] S. Nesargi and R. Prakash, "MANETconf: configuration of hosts in a mobile ad hoc network," In Proceedings of the 21st Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM 2002), New York, NY, June 2002

[5] H. Zhou, L. M. Ni, and M. W. Mutka, “Prophet address allocation for large scale MANETs,” In Proceedings of the 22nd Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM 2003), San Francisco, CA, April 2003

[6] H. Zhou, L. M. Ni, and M. W. Mutka, “Prophet address allocation for large scale MANETs,” Ad Hoc Networks Journal, Vol. 1, Issue 4, pp 423-434, November 2003

[7] J. Couceur, “The sybil attack,” In Proceedings of the 1st Workshop on Peer-to-Peer Systems (IPTPS’02), Cambridge, MA, March 2002

[8] S. Yi, P. Naldurg, and R. Kravets, “A security-aware routing protocol for wireless ad hoc networks,” In Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, FL, July 2002

[9] B. Dahill, B. N. Levine, E. M. Royer, and C. Shields, “A secure routing protocol for ad hoc networks,” In Proceedings of the 10th International Conference on Network Protocols (ICNP’02), Paris, France, November 2002

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM 2002), Atlanta, GA, September 2002

[11] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” IEEE Network, Vol. 13, No. 6, pp. 24-30, 1999

[12] Čapkun, L. Buttyán, and J. P. Hubaux, “Self-organized public-key management for ad hoc networks,” In IEEE Transactions on Mobile Computing, Vol.2, No. 1, January-March 2003

[13] C. Perkins, E. M. Belding-Royer, and S. R. Das, “Ad hoc on-demand distance vector (AODV) Routing,” Network Working Group RFC 3561, July 2003

[14] E. Pagani and G. P. Rossi, “Reliable Broadcast in Mobile Multihop Packet Neworks,” In Proceedings of the 3rd Annual International Conference on Mobile Computing and Networking (MOBICOM 1997), pp. 34-42, Budapest, Hungary, September 1997

[15] W. Lou and J. Wu, “A Reliable Broadcast Algorithm with Selected Acknowledgements in Mobile Ad Hoc Networks,” In Proceedings of IEEE 2003 Global Communications Conference (GLOBECOM 2003), San Francisco, CA, December 2003

[16] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM 2000), pp. 255-265, Boston, MA, August 2000

[17] K. Fall and K. Varadhan (editors), The ns manual - the VINT Project. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html, April 2007