# Securing Data Delivery in Ad Hoc Networks

Wenjing Lou and Yuguang Fang

Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611

***Abstract*** *- A novel Secure Protocol for REliable dAta Delivery (SPREAD) is proposed to enhance the secure data delivery in a mobile ad hoc network. The basic idea is to divide a secret message into multiple shares by secret sharing and deliver them via multiple independent paths to the destination. By this means, an adversary (adversaries) will have more difficulty to compromise the message delivered therefore improved data confidentiality can be expected. This paper outlines the system architecture and the major design issues of SPREAD scheme. A multiple paths optimization technique is proposed to find as many as possible and at the same time as secure as possible paths. The simulation results justify the feasibility of the SPREAD approach and verify the effectiveness of the scheme by showing the significantly reduced message compromise probability.*

## 1 Introduction

Security is a critical issue in a mobile ad hoc network (MANET). As compared with an infrastructured or wired network, a MANET poses many new challenges in security. For example, wireless channel is more vulnerable to attacks such as passive eavesdropping, or active signal interference and jamming; the co-operative MANET protocols are more vulnerable to denial of service attacks; the lack of infrastructure and limited resources restrict the applicability of some conventional security solutions; and the un-predictable ad hoc mobility makes it more difficult to detect the malicious behavior [1].

Due to these new challenges, many security solutions that have been effective in a wired network become inapplicable in a MANET. Much effort has been made to develop applicable security solutions dedicated to a MANET environment. Among them, key management, probably the most critical and fundamental security issue in a MANET, has attracted much attention [2,3,4]. A number of secure routing protocols have also been proposed to protect the correctness of different types of ad hoc routing protocols, both table-driven/on-demand and distance vector/source routing types [5,6,7,8]. Some other issues that have been addressed in the current literature include handling node misbehavior [9,10,11], intrusion detection [12], and so on [1].

The scheme suggested in this paper addresses data confidentiality service in a MANET. Data confidentiality is the protection of transmitted data from passive attacks, such as eavesdropping. Sensitive information, such as tactical military information transmitted across a battlefield (an ad hoc network), requires confidentiality. Leakage of such information to enemies could cause devastating consequences. The wireless channel in a hostile environment is vulnerable particularly to the eavesdropping. Messages transmitted over the air can be eavesdropped from anywhere without having the physical access to the network components. Conventionally, confidentiality is achieved by cryptography. However, the limited resources, such as the limited battery power and processing capability, restrict the use of computationally intensive encryption schemes in a MANET. The computationally efficient encryption schemes sometimes are not secure enough. For example, the WEP (Wired Equivalent Privacy) protocol defined in IEEE 802.11 uses RC4 algorithm, which is a stream cipher and computationally efficient. However, it has been discovered that it can be decrypted through traffic analysis and dictionary-building attack that, after analysis of about a day's worth of traffic, allows real-time automated decryption of all traffic [13]. A more severe problem in a MANET is that, mobile nodes usually reside in an open and hostile environment. Nodes themselves might be compromised. For example, in the battlefield scenario, nodes might be captured. In this case, all the credential stored in the nodes would be compromised, including the keys. Any encryption scheme, no matter how secure it is, would not help.

Based on these observations, we propose a novel scheme, Secure Protocol for REliable dAta Delivery (SPREAD), to statistically enhance data confidentiality in a MANET. The fundamental idea of SPREAD is shown in Figure 1. Assume that we have a secret message, if we send it through a single path, the enemy could compromise it by compromising any one of the nodes along the path. However, if we divide it into multiple pieces, and send the

This work was supported in part by the Office of Naval Research Young Investigator Award under grant N000140210464 and the Office of Naval Research under grant N000140210554.

multiple pieces via multiple independent paths, then the enemy would have to compromise all the pieces from all the paths to compromise the message. Improved security can be achieved by this means.

Here, to compromise the message, the enemy must accomplish at least two things. First, the enemy must intercept all pieces of the message. This can be done by either eavesdropping or compromising nodes. Either way, by spreading the message pieces into multiple paths, the enemy would have more difficulty to collect all the pieces. Secondly, we assume link encryption between neighboring nodes, each link with different keys. The establishment of a shared session key between neighboring nodes is not difficult although the key management in a MANET is problematic. So even the enemy collected all the pieces, he has to decrypt them. The decryption can be done by either compromising the nodes or by brute-force type of attack or traffic analysis, while the latter requires a large amount of encrypted data by the same key. The more data, the better chance the decryption. By spreading the traffic onto multiple paths, it also makes it harder for the enemy to decrypt the message. Improved security can be expected from SPREAD scheme.

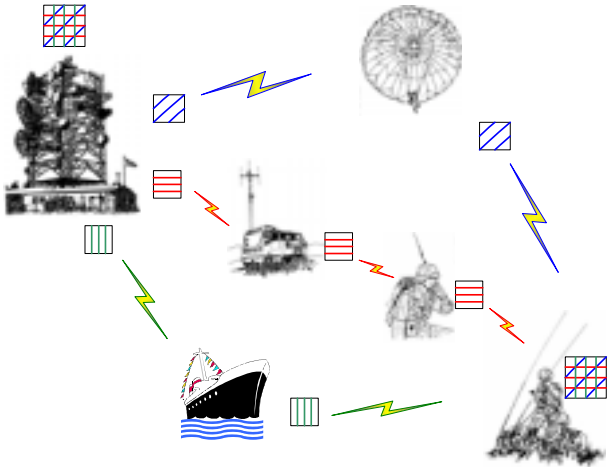

Figure 1 Illustration of SPREAD idea

In this paper, we address the improved security by dealing with the compromised nodes and eavesdropping problems. We consider both individual attacks and colluded attacks (multiple compromised nodes are working together to recover the message). We assume that the adversaries, after compromising the nodes, will attempt to remain in the network by launching only passive attacks in order to acquire more secure information. If the compromised nodes launch active attacks, such as stop forwarding packets for other nodes or altering the information when forwarding packets, some intrusion detection mechanism [12] or the misbehavior detection schemes such as a watchdog proposed in [9] can be used to identify the compromised nodes quickly so that it will be excluded from the network.

## 2 SPREAD Architecture

Several issues need to be addressed for SPREAD scheme in order to maximize the security. First, how do we divide the secret message into multiple pieces? Secondly, how the message pieces should be allocated onto each selected path? Thirdly, how do we discover the desired multiple paths in a MANET? We will briefly describe the first two issues as we have discussed them in other papers [14,15]. In this paper, we focus on the third issue.

### *2.1 Secret Sharing*

In our SPREAD scheme, we use the threshold secret sharing algorithm to divide the secret message into multiple pieces. Threshold secret sharing algorithms have been well studied in the literature. Assume that we have a system secret and we divide it into $N$ pieces, called *shares* or *shadows*. Each of $N$ participants of the system holds one share of the secret respectively. Any less than $T$ participants cannot learn anything about the system secret, while with an effective algorithm, any $T$ out of $N$ participants can reconstruct the system secret. This is called a *($T$,$N$) threshold secret sharing scheme.*

With a ($T$,$N$) secret sharing algorithm, the secret message can be divided into $N$ message shares such that in order to compromise the message, the enemy must compromise at least $T$ shares. With less than $T$ shares, the enemy could learn nothing about the message and he has no better chance to recover the secret than an outsider who knows nothing at all about the message. The generation of the message shares and the reconstruction of the message are all linear operations over a finite field. The computational overhead is trivial ($O(Tlog^2T)$). The detailed information on how to apply secret sharing algorithm in our SPREAD scheme can be found in [14].

### *2.2 Optimal Share Allocation*

The second issue is how to select the paths, how to choose an appropriate value of ($T$,$N$), and how to allocate the shares onto each selected path such that the maximum security can be achieved. The simplest and most intuitive share allocation scheme is to choose $N$ as the number of available paths, apply ($N$,$N$) secret sharing, and allocate one share onto each path. This will achieve the desired maximum security with least processing cost. However, in an ad hoc network, wireless links are instable and the topology changes frequently. Sometimes packets might be dropped. In the case that packet loss does occur, this type of non-redundant share allocation will disable the reconstruction of the message at the intended destination. To deal with this problem, we introduce redundant (i.e. $T<N$) SPREAD scheme to improve the reliability. In [15] we discussed the optimal share allocations. We formulated the share allocation into a constrained optimization problem, with the objective to minimize the message compromise probability. Our investigation to the optimal

share allocation reveals that, by choosing an appropriate (*T,N*) value and allocating the shares onto each path carefully, we could improve the reliability by tolerating certain packet loss without sacrificing the security. The maximum redundancy we can add to the SPREAD scheme without sacrificing security is identified as $r < 1/m \ (m \geq 2)$, where $r = 1 - T/N$ is the redundancy factor and $m$ is the number of paths selected to deliver the message. The optimal share allocation is proposed. Basically any allocation that conforms to the constraints

$$\begin{cases} N - T + 1 \leq n_i \leq T - 1, & \text{i} = 1,..., \text{m} \\ \sum_{i=1}^{M} n_i = N \end{cases}$$

is an optimal share allocation in terms of security. More details about share allocation can be found in [15].

### *2.3 Multipath Routing*

Routing in ad hoc networks presents great challenge because the nodes in ad hoc networks can move freely and the topology changes continuously and unpredictably. A great effort has been made to design ad hoc routing protocols. Multipath routing technique is a promising choice since the use of multiple paths in a MANET could diminish the effect of unreliable wireless links and the constant topological changes. Several multipath routing schemes have been proposed to improve the reliability, fault-tolerance, end-to-end delay for bursty traffic, as well as to achieve load balancing etc. [16,17,18].

For our SPREAD scheme, we need independent paths, more specifically, node disjoint paths, because we are dealing with compromised node problem. Several multipath routing protocols have been proposed in MANETs with the design goal to find node-disjoint paths, such as the diversity injection technique [17], and the on-demand multipath routing [18]. The dynamic source routing protocol itself is also capable of maintaining multiple paths from the source to a destination. Those proposed protocols are all on-demand, due to the network bandwidth limitation, and source routing type, as the source routing provides the source with the capability of controlling the disjointness of the paths. Those on-demand protocols work by broadcasting the route inquiry messages throughout the network and then gathering the replies from the destination and other nodes. Although those routing protocols are able to find multiple node-disjoint paths, the set of paths provided by them might not be optimal for our SPREAD scheme as the cost function they are based on is usually the hop count or propagation delay, not necessary the security.

For on-demand routing protocols, some type of cache is necessary to store the routes previously found so that the node does not have to perform the costly route discovery for each individual packet. In DSR and the multipath extension of DSR, the route replies back to the source contain the complete node list from the source to the destination. By caching each of these paths separately, a "path cache" organization can be formed. This type of cache organization has been widely used. However, the paths found by this means might not serve our purpose best. They are not necessary the most secure paths. In [19], we designed an alternative cache organization, called a "link cache", in which routes are decomposed into individual links and represented in a unified graph data structure. Given the same amount of route reply information, the routes existing in a path cache can always be found in a link cache. Thus a link cache has the potential to use the route information more efficiently. We also developed an adaptive stale link removal scheme to work together with the link cache. By using such a link cache, we could separate the routing and the selection of the paths. Although we rely on an underlying routing protocol to provide us with a partial view of network topology, the selection of the optimal paths can be done orthogonal of the routing protocols used, based on the discovered partial network topology. In the next section, we present the maximal paths finding algorithm that is trying to select a set of paths, when used to deliver the message shares, providing the maximum overall security.

## 3 Maximal Paths Finding Algorithm

Assume that we have totally $M$ node-disjoint paths available. The security can be maximized when we allocate the shares in such a way that the enemy has to compromise all the $M$ paths to compromise the necessary $T$ shares. Here we assume that the enemy compromises shares by compromising nodes where the shares are relayed. We use $P_{msg}$, the probability that the message might be compromised, to indicate the security of the SPREAD scheme. Then $P_{msg}$ can be calculated as follows,

$$P_{msg} = \prod_{i=1}^{M} p_i$$

where $p_i$ *(i=1,2,…,M)* is the probability that path $i$ is compromised, i.e., the probability that any intermediate node in path $i$ is compromised.

Assume that with probability $q_i$ that node $n_i$ might be compromised. Then the probability that a (*s,t*) path consisting of node *s*, $n_1$, $n_2$, …, $n_l$, *t* might be compromised equals to

$$p = 1 - (1 - q_1)(1 - q_2) \cdots (1 - q_l)$$

Since we consider the protection of messages when they are transmitted across the network, we assume that the source and the destination are safe with $q_s = q_d = 0$. Note that the probability $q_i$ indicates the security level of node $i$ and it could be estimated from the feedback of some security monitoring software and/or hardware such as firewalls and intrusion detection devices. It could also be assigned manually by administrators based on the level of physical protection to nodes, the positions of nodes, or the rankings

Step 1. Find the first most secure path by modified Dijkstra algorithm, select the path

Step 2. Perform a graph transformation as follows
For each selected path:

a. Replace the links used in the path with directed arcs – for the arc that is directed towards the source, make its cost the negative of the original link cost; make the cost of the arc directed towards the destination infinite (e.g. remove it)

b. Split each node on the selected paths (except the source and destination) into two collocated subnodes; Connect the two subnodes by an arc of cost 0 and directed towards the source node.

c. Replace each external link that is connected to a node in the selected paths by its two component arcs of cost equal to the link cost – let one arc terminate on one subnode and the other one emanate from the other subnode such that along with the zero-cost arc, a cycle does not result.

Step 3. Run the modified Dijkstra algorithm, find the most secure path in the transformed graph

Step 4. Transform back to the original graph; erase any interlacing edges; group the remaining edges to form the new path set.

Step 5. Go to step 2, until no more path can be found or the security of the path set does not increase..

Figure 2 Maximal node disjoint path finding algorithm

of nodes, etc.

Ideally, given a network, we wish to find an optimal path set, such that the probability $P_{msg}$ is minimized. Intuitively, since $p_i$ is a probability which is always less than 1. The more items of $p_i$, the less the probability, the better the security. So the general goal of our path finding algorithm is to find as many as possible paths while at the same time as secure as possible.

The maximal paths finding algorithm proposed for our SPREAD scheme is modified from the node disjoint shortest pair algorithm [20]. A modified Dijkstra algorithm is used so that negative links are allowed (but no negative loop) in the graph [20]. The modified Dijkstra algorithm modifies the standard Dijkstra algorithm by allowing the permanent labeled node change back to a tentative label when a smaller cost to that node is found. We define the following link cost function to convert the security characteristics into an additive link cost function so that the shortest path algorithm is readily used as most secure path finding algorithm.

We define the cost function of link between node $n_i$ and $n_j$ as

$$c_{ij} = -\log\sqrt{(1-q_i)(1-q_j)}$$

Then the cost of the the $(s,t)$ path using shortest path algorithm is

$$\begin{aligned}\cos t(s,t) &= c_{s1} + c_{12} + \cdots + c_{l-1,l} + c_{ld} \\ &= -\log(1-q_1) - \log(1-q_2) - \cdots - \log(1-q_l) \\ &= -\log\{(1-q_1)(1-q_2)\cdots(1-q_l)\}\end{aligned}$$

With the shortest path algorithm,

$$\begin{aligned}&\cos t(s,t) \text{ is minimized} \\ &\Rightarrow -\log\{(1-q_1)(1-q_2)\cdots(1-q_l)\} \text{ is minimized} \\ &\Rightarrow (1-q_1)(1-q_2)\cdots(1-q_l) \text{ is maximized} \\ &\Rightarrow p = 1-(1-q_1)(1-q_2)\cdots(1-q_l) \text{ is minimized}\end{aligned}$$

So the path found by the shortest path algorithm would be the most secure path when the proposed cost function is used.

The maximal paths algorithm is then an iterative procedure. The most secure path is found first and added to the path set. Then in each iteration, the number of paths in the set will be augmented by one. Figure 2 summarizes the steps taken to find the maximal number of paths. Each time a new path is added to the set of selected paths, a graph transformation is performed, which involves a vertex splitting of the nodes on the selected paths (except the source and destination node). Then the modified Dijsktra algorithm is executed to find the most secure path in the transformed graph. Then by transforming the split nodes back to the original one, erasing any interlacing edges, grouping the remaining edges, the new path set is formed. In each iteration, the number of paths will be augmented by one.

Figure 3 shows an example of the path finding algorithm. After finding the first two node-disjoint paths, the third one temporarily makes use of the selected nodes but using the

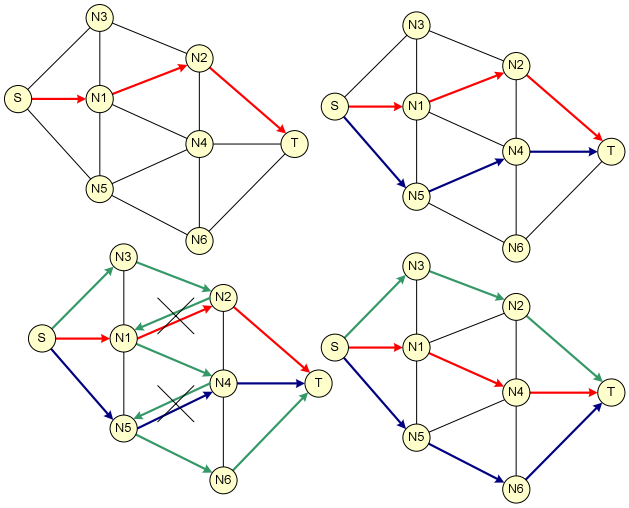


Figure 3 Illustration of the maximal node disjoint paths algorithm

link in the reverse direction. After the interlacing removal and regrouping, a path set consisting of 3 paths is found instead of 2.

Because of the regrouping of edges, the paths in the path set in each iteration might change. So we calculate $P_{msg}$ after each iteration. If $P_{msg}$ is not getting smaller in the iteration, the path set found in the previous iteration will yield the best security results. The path finding algorithm terminates.

## 4 Simulation Results

In this section we present the simulation results to show the effectiveness of the SPREAD scheme in enforcing the data confidentiality. We simulate an ad hoc network with 100 nodes randomly deployed in a 1000m by 1000m area. The transmission range of each node is set equal in each simulation and varies in different simulations. The simulation results are averaged over 20 randomly deployed networks. To factor out the effect of routing protocols, in the simulation we assume the network topology is known. In each network, we find 1, 2, …, till maximal node-disjoint paths for each source-destination pair which is at least three hops away. Two sets of simulations are executed. In the first set, each node is assumed equally likely to be compromised with probability $q_i$=0.152. In the second set of simulation, each node is assigned a probability randomly: 10% of nodes with probability $q_i$=0.50, 30% of nodes with $q_i$=0.20, 40% of nodes with $q_i$=0.10, and 20% of nodes with $q_i$=0.01. In the first set, all the links are of same cost. In the second set, we use the proposed link cost function to define the link cost based on the node security level ($q_i$).

Table 1 gives some basic idea of the network topology of simulated ad hoc networks. We see that ad hoc networks typically have dense connectivity which allows the exploitation of multipath routing techniques.

Table 1 Network parameters

| TR(m) | 200 | 250 |
|---|---|---|
| Node degree | 10.3 | 15.4 |
| Diameter | 9 | 6.8 |

Figure 4 shows the probability that multiple paths are found in the simulated network. It is observed that the probability that multiple node disjoint paths exist in an ad hoc network is pretty high. Since our SPREAD scheme depends on the availability of multiple paths, the existence of such multiple paths justifies the feasibility of our scheme.

Figure 5 shows the probability that the message is compromised when multiple paths are used. Here, we consider the case that the message is compromised due to compromised nodes. This probability is the probability for colluded attacks. One message is considered compromised

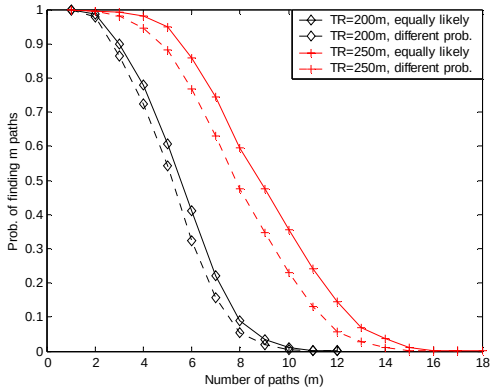


Figure 4 Capability of path finding

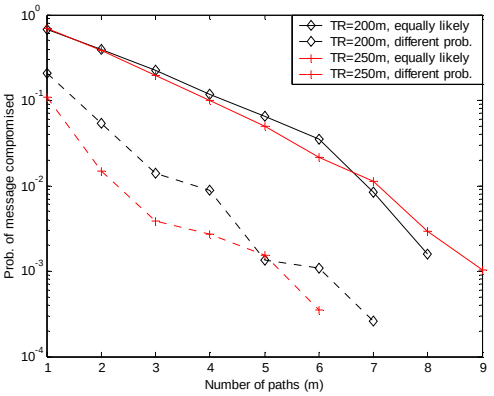


Figure 5 Message Compromise Probability

when at least one compromised node is located on each of the paths selected to deliver this message. This probability for individual attack is zero when multiple (>1) paths are used because no single node is able to relay all the necessary shares. Noticing the logarithmic scale of the probability, we observe that the probability drops quickly (actually exponentially fast) with the increase of the number of paths used. This result verifies the effectiveness of our SPREAD idea. We also noticed that when nodes are with different security level, our algorithm tends to select more secure paths that further decrease this probability significantly.

Figure 6 shows the probability that a message is eavesdropped when multiple paths are used. Since the wireless channel is a broadcast channel, anyone sits within the transmission range of a transmitting node is able to eavesdrop (overhear) the node's transmission. This figure actually presents the probability for individual attack. The probability for colluded attack is pretty high (almost 1) because in our simulation, we have about 15 compromised nodes among the totally 100 nodes. It is observed that, with the increase of the number of paths, this probability decreases. However, the decrease becomes less significant when more paths are used. In fact, there is a lower bound of this probability because anyone sits within the transmission

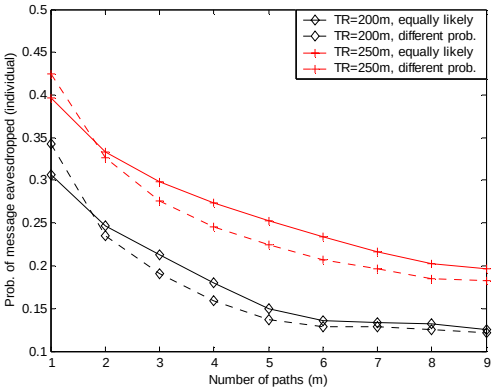


Figure 6 Message eavesdropped probability

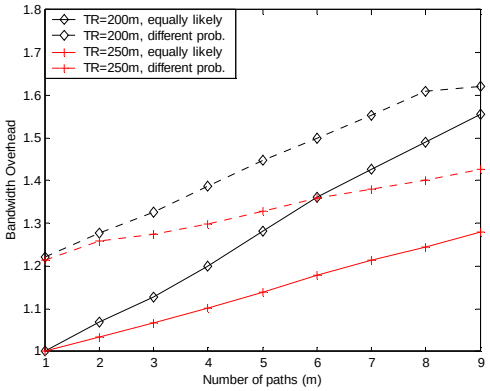


Figure 7 Bandwidth overhead

range of the source node would be able to overhear all the shares. Of course, this probability is the one that an adversary might overhear a message, it does not mean that the message can be compromised because the message shares are encrypted as well. Again, this verifies that the SPREAD idea makes it harder for an enemy to collect enough data to break the secret.

Figure 7 shows the bandwidth overhead calculated on a per-hop basis when multiple paths are used compared with the single minimum-hop path case. We can see that using multiple paths does consume more network bandwidth because longer paths are used. However, this is the tradeoff. For security critical applications, the network efficiency might not be a major concern.

## 5 Conclusions and Discussions

The basic idea of SPREAD is to distribute the secrecy, first by secret sharing algorithm at the source node and then by multipath routing while shares are transmitted across the network, so that in the event that a small number of shares are compromised, the secret itself will not be compromised. A few remarks are in order. First, the SPREAD scheme considers the security when messages are transmitted across the network, assuming the source and destination are trusted. Secondly, the SPREAD scheme cannot address the confidentiality alone, it only statistically enhances such service. For example, it is still possible for adversaries to compromise all the shares, e.g. by collusion. Finally, the SPREAD can be made adaptive in the sense that the source node could make final decision whether a message is delivered at certain time instant according to the security level and the availability of multiple paths. Moreover, the chosen set of multiple paths may be changed from time to time to avoid any potential capture of those multiple shares by adversaries.

## Reference


[1] W. Lou, Y. Fang, “A survey on wireless security in mobile ad hoc networks: challenges and available solutions”, book chapter in *Ad Hoc Wireless Networking*, to be published by Kluwer in May 2003

[2] L. Zhou and Z. J. Haas, ``Securing ad hoc networks,'' *IEEE Network Magazine*, vol. 13, no. 6, November/December 1999

[3] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, ``Providing robust and ubiquitous security support for manet,'' *Proceedings of the 9th IEEE International Conference on Network Protocols(ICNP)*, 2001

[4] J-P. Hubaux, L. Buttyan and S. Capkun, ``The quest for security in mobile ad hoc networks, *MobiHOC'01*, 2001.

[5] Y.-C. Hu, D. B. Johnson and A. Perrig, ``SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks,'' *WMCSA'02*, June 2002.

[6] Y.-C. Hu, A. Perrig and D. B. Johnson, ``Ariadne : a secure on-demand routing protocol for ad hoc networks,'' *MobiCom 2002*, September 2002.

[7] P. Papadimitratos and Z. J. Haas, ``Secure routing for mobile ad hoc networks,'' *CNDS 2002*, San Antonio, TX, January 2002

[8] H. Yang, X. Meng and S. Lu, ``Self-organized network-layer security in mobile ad hoc networks,'' *ACM WiSe'02*, September 2002.

[9] S. Marti, T. Giuli, K. Lai and M. Baker, ``Mitigating routing misbehavior in mobile ad hoc networks,'' *MobiCom'00*, Boston, MA, USA, August 2000.

[10] L. Buttyan and J.-P. Hubaux, ``Enforcing service availability in mobile ad hoc networks,'' MobiHOC'00, 2000

[11] S. Buchegger and J.-Y. Le Boudec, ``Performance analysis of the CONFIDENT protocol,'' *MobiHOC'02*, June 2002.

[12] Y. Zhang, W. Lee and Y. Huang, ``Intrusion detection techniques for mobile wireless networks,'' *ACM/Kluwer Mobile Networks and Applications (MONET)*, to appear.

[13] “Security of the WEP algorithm”, http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

[14] W. Lou, Y. Fang, “A multipath routing approach for secure data delivery”, *IEEE Milcom’01*, Oct 2001

[15] W. Lou, W. Liu, Y. Fang, “SPREAD: Improving network security by multipath routing”, *IEEE Milcom’03*, Boston, MA, Oct 2003

[16] A. Tsirigos, Z.J. Haas, “Multipath routing in the presence of frequent topological changes”, *IEEE Communication Magazine*, Nov 2001

[17] M.R. Pearlman, Z.J. Haas, P. Sholander, S. S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks”, *MobiHOC*, 2000

[18] K. Wu, J. Harms, “Performance study of a multipath routing method for wireless mobile ad hoc networks”, *9th international symposium on modeling, analysis and simulation of computer and telecommunication system*, 2001

[19] W. Lou, Y. Fang, “Predictive caching strategy for on-demand routing protocols in ad hoc networks”, *Wireless Networks*, vol.8, issue 6, Nov 2002

[20] R. Bhandari, Survivable Networks – Algorithms for diverse routing, Kluwer Academic Publisher, 1999