# Secure Multipath Communication in Mobile Ad hoc Networks*

Mike Burmester and Tri Van Le
Department of Computer Science
Florida State University
email: {burmester,levan}@cs.fsu.edu

## Abstract

*Mobile ad hoc networks (MANETs) are collections of autonomous mobile nodes with links that are made or broken in an arbitrary way. They have no fixed infrastructure and may have constrained resources. The next generation of IT applications is expected to rely heavily on such networks. However, before they can be successfully deployed several security threats must be addressed. These are mainly due to the ad hoc nature of these networks. Consequently it may be much harder (or even impossible) to establish routing communication paths that can tolerate malicious faults. In this paper we first propose a general Bayesian model that satisfies the basic mobility requirements of a MANET and define the requirements for secure communication in this model. We then consider several multipath routing schemes and propose a new adaptive multipath routing algorithm that satisfies our security requirements.*

## 1. Introduction

Mobile ad hoc networks (MANETs) are dynamic collections of autonomous mobile nodes with links that are changing in an unpredictable way. They are characterized by a dynamic topology and the lack of any fixed infrastructure. The communication medium is broadcast. The nodes can be regarded as wireless mobile hosts with limited power (operating off batteries), constrained bandwidth and transmission range (typically 250–1000 meters in an open field). The recent rise in popularity of mobile wireless devices and technological developments has made possible the deployment of such networks for several applications. Indeed, because ad hoc networks do not have any fixed infrastructure such as stations or routers, they are highly applicable for emergency deployments, disasters, search and rescue missions and military operations.

Finding and maintaining routes in a MANET is a major challenge. So far, most of the research has focused on functionality issues and efficiency (e.g., [3, 4, 9, 14, 18, 19, 20, 23]), with security being given a lower priority, and in many cases, regarded as an add-on afterthought technology rather than a design feature (e.g., [2, 16, 21]). Although such an approach may be suitable for networks with predictable faults, it not suitable for MANETs in which we have unpredictable or malicious (Byzantine [17]) faults [6, 7].

Of particular concern is the possibility that an established route is under the control of a malicious adversary, and will be disconnected at a critical time when damage is maximized, and when there is not sufficient time to fix the route or to find alternative routes. In such cases multipath routing is of benefit. Multipath routing involves the establishment of multiple paths between source and destination pairs. These paths are used for replicated (or redundant) communication to prevent Byzantine attacks. In particular, if $k$ is an upperbound on the number of malicious faults then secure communication is achieved by using $(2k+1)$ vertex-disjoint paths linking the source to the destination (see e.g., [17]). If cryptographic mechanisms (authentication and encryption) are used then only $k+1$ vertex-disjoint paths are needed. Multipath routing will also enhance bandwidth usage, load balancing and more generally efficiency (e.g., [24, 12]).

Our main goal in this paper is to propose a multipath source to destination routing algorithm that can be used for secure communication in the presence of Byzantine faults. We first describe a general model for MANETs that satisfies the basic mobility requirements of such networks, and then formally define our security requirements in this model for Byzantine adversaries. We use this model to prove that our multipath routing algorithm is robust against Byzantine attacks.

The rest of this paper is organized as follows. In Section 2 we propose a model for ad hoc networks, based on a Bayesian inference structure and give our definitions. In Section 3 we consider multipath routing algorithms. We motivate the requirement for such algorithms and discuss several multipath scenarios and then propose a multipath algorithm that is secure against Byzantine faults.

*This material is based on work supported in part by the U.S. Army Research Laboratory and the U.S. Research Office under grant number DAAD19-02-1-0235.

## 2. A model for ad hoc networks

There are several ways in which one can model the unpredictable nature of an ad hoc network. Whichever way is used, there are important mobility aspects that must be reflected in the model. Clearly the model has to be time dependent and stochastic. That is, there must be a probability distribution $p_{ij}^t = \Pr[(x_i, x_j) = 1]$ defined on all possible links $(x_i, x_j)$ of the network. Here $p_{ij}^t$ is the probability that pair of distinct nodes $(x_i, x_j)$ is linked at time $t$. Since the ad hoc-ness of the system is due to the mobility of the nodes, the model should reflect this. In particular, the link probability distribution should have *memory*. Consequently we model an ad hoc network by using a *Bayesian inference* structure for which established links have a high probability to remain so and a low probability to be disconnected. Similarly, disconnected links have a high probability to remain so and a low probability to become connected [6] –Figure 1.

$$
\begin{aligned}
\Pr[(x_i,x_j)^t = 0 \mid (x_i,x_j)^{t-1} = 0] &= a_{ij}^{0t} \quad (high)\\
\Pr[(x_i,x_j)^t = 1 \mid (x_i,x_j)^{t-1} = 0] &= 1 - a_{ij}^{0t} \ (low)\\
\Pr[(x_i,x_j)^t = 1 \mid (x_i,x_j)^{t-1} = 1] &= a_{ij}^{1t} \quad (high)\\
\Pr[(x_i,x_j)^t = 0 \mid (x_i,x_j)^{t-1} = 1] &= 1 - a_{ij}^{1t} \ (low)
\end{aligned}
$$

**Figure 1.** The Bayesian inference requirements

That is, the à posteriori probability that $x_i, x_j$ are linked given that $x_i, x_j$ where previously linked is high, and similarly for the non-linked case. On the other hand the conditional probabilities that links are broken (or made) are low. The link probabilities are determined jointly by the nodes $x_i$, Nature and possibly the Adversary. The contribution of the nodes comes from their mobility. Nature's contribution comes from the fact that communication is wireless. A wide range of environmental factors may affect communication, ranging from weather and radio interference to physical obstacles. Finally there are scenarios in which the Adversary influences the mobility of the system, as for example in "seek and destroy" missions. These probabilities may also be linked by Markov interdependencies, to reflect the particular nature of the node mobility. The Bayesian model supports the automatic derivation of probabilities for a set of possible causes and supports a stochastic infrastructure. However there are cases when the unpredictable nature of the mobility of the system is such that the uncertainty (entropy) of this infrastructure is maximal, resulting in networks for which there is no useful structure.

**Definition 1** Let $\mathcal{G} = \{(V, L^t)\}$ be a family of networks[1] with node set $V$ and link sets $L^t$, indexed by $t \in Z^+$, where $t$ is time, and let $\mathcal{N}$ be Nature. The nodes $x_i \in V$ and $\mathcal{N}$ are *Interactive Turing Machines*, that is Turing Mathines [1] with a random tape, a read tape and a write tape. These tapes are used for neighbor communication. The random tape is used to describe the mobility of the nodes. $\mathcal{G}$ is an *ad hoc* network, if the transitional link probabilities $\{a_{ij}^{bt}\}$, $b = 0, 1$, satisfy the Bayesian inferences in Figure 1. The transitional link probability distribution is determined *jointly* by the nodes of $V$, Nature $\mathcal{N}$ and possibly the adversary $\mathcal{A}$.
*Nature* $\mathcal{N}$: $\mathcal{N}$ has $|V|$ read and write tapes that are shared with the nodes of $V$ in such a way that $\mathcal{N}$ can control jointly their mobility. The randomness of $\mathcal{N}$ is independent of that of the nodes $x_i$ of the network and that of $\mathcal{A}$.
*The Adversary* $\mathcal{A}$: $\mathcal{A}$ is Byzantine and can corrupt up to $k$ nodes of $V$ throughout the lifetime of the system. The corrupted nodes are selected at random from the node set $V$. $\mathcal{A}$ may also control some *enemy* nodes $x_i{}' \in V'$, where $V \cap V' = \emptyset$. The nodes in $V'$ are also Interactive Turing Machines.

**Definition 2** The ad hoc network $\mathcal{G} = \{(V, L^t)\}$ is $\varepsilon$-*simulatable* if there is a probabilistic polynomial time algorithm which on input the inputs of the nodes $x_i \in V$ will output a link probability distribution $\{\hat{a}_{ij}^{bt}\}$, $b = 0, 1$, such that: $\sum_{ij} |\hat{a}_{ij}^{bt} - a_{ij}^{bt}| / \sum_{ij} 1 < \varepsilon$.

Ad hoc networks are subject to storms that are unpredictable changes in the transitional link probabilities $a_{ij}^{bt}$, $b = 0, 1$, caused by Nature. Such events cannot be simulated. The duration of a storm may be short, and the network may revert to a Bayesian simulatable state. However the unpredictability of the distribution of a storm may make it impossible to simulate such networks.

Recently [7] the authors proved that an ad hoc network $\mathcal{G} = \{(V, N^t)\}$ converges for large values of $t$, and therefore can be *simulated* if the transitional link probabilities in Figure 1 are fixed, that is $a_{ij}^{0t} = a_{ij}^0$ and $a_{ij}^{1t} = a_{ij}^1$ for $t = 0, 1, 2, \ldots$. This result however does not extend to the case when we have unpredictable storms.

**Definition 3** A communication algorithm over a routing path with source $s$ and destination $d$ in an ad hoc network $\mathcal{G} = \{(V, N^t)\}$ is *secure* if:[2]

- *Fault-tolerance*: For any adversary $\mathcal{A}$, if $d$ accepts a packet as sent by $s$, then $s$ sent this packet.
- *Privacy*: For any adversary $\mathcal{A}$, we have privacy [22] for all packets sent by $s$ to $d$.

Since privacy can easily be achieved by using cryptographic tools, we shall focus in this paper on fault-tolerant communication.

[1] For simplicity we assume that the transmission range of all mobile nodes is the same (omnidirectional). If this is not the case (sectored) then we have to use directed graphs.

[2] This is an informal definition. A more appropriate definition would allow for a small probability of error.

## 3. Multipath Routing

Multipath routing is needed for secure communication when route recovery cannot be guaranteed to be done fast enough because of the high mobility of the system. With standby paths, traffic can be redirected whenever we have route failure, thus reducing route recovery time. Multipath routing also offers other quality of service advantages (such as, load balancing, aggregation of network bandwidth, reducing traffic congestion etc).

Multipath routing in networks with no fixed infrastructure is a major challenge and in general requires a different approach from that used with fixed infrastructures. In this section we will first describe some general approaches that can be used to establish multipath routing by exploiting some particular features of ad hoc networks. Since we focus on security issues in this paper we are only concerned with vertex-disjoint paths. Our first example is of a location-based multipath algorithm.

### 3.1. Geodesic routing –location based routing

With location based routing, each node of the network $\mathcal{G}$ is assumed to know its approximate location (by either using a GPS device or some other means [8]). Vertex-disjointness is established by using spatially disjoint routes.

**Circle Based Routing.** With CBR, a family of circles incident with the source $s$ and destination $d$ is used for routing. The source selects a few circular paths $\pi_i$ with sufficient space disjointness –not too close (dealing with the areas close to $s$ and $d$ requires some attention). Packets are directed along these paths. The location coordinates of the center $C_i$ and the radius $R_i$ of the corresponding path $\pi_i$ are appended to each packet. To allow for the possibility that there may not be sufficient forwarding nodes on the selected paths, the paths may be broadened to narrow corridors by allowing nodes a few hops away to be used. The breadth is a variable selected by $s$ and is also appended to the packets sent. Two other variables are appended to the packets: $direction$ (clockwise/counterclockwise) and $ttl$ (time-to-live) in hop counts.

The location of $d$ is obtained by ordinary flooding. Public key encryption is used to protect the location of $d$. There are two communication modes that can be used with CBR: $(i)$ *multipath* routing and $(ii)$ *multibraid* routing. The first involves forwarding packets to specified neighbors along the paths $\pi_i$. The second is, essentially, directed flooding along given circular corridors (of specified hop diameter). With multibraid routing the *only* location information about $s, d$ that leaks to the intermediate nodes is the center and radius of the path used. This routing mode is appropriate for high mobility applications, and does not require any local neighborhood knowledge. Both modes scale well with intermediate node behavior being determined only at packet arrival time.

The paths are determined as follows. Let $S = (x_s, y_s)$ and $D = (x_d, y_d)$ be the coordinate positions in the Euclidean plane of the source $s$ and destination $d$. The source $s$ first computes the midpoint of $S, D$: $(x_{mid}, y_{mid})$, where $x_{mid} = \frac{x_s+x_d}{2}$, $y_{mid} = \frac{y_s+y_d}{2}$, and the slope $m$ of $(S, D)$: $m = \frac{y_d-y_s}{x_d-x_s}$. To select $k+1$ circular paths, $s$ chooses random numbers $t_i, i = 1, 2, \ldots, k+1$ and computes the coordinates $(x_i^*, y_i^*), i = 1, 2, \ldots, k+1$, of the centers $C_i$ and the radius $R_i$ as follows:

$$x_i^* = x_{mid} + t_i, \quad y_i^* = y_{mid} - t_i/m,$$
$$R_i = \sqrt{(\frac{x_d - x_s}{2} + t_i)^2 + (\frac{y_d - y_s}{2} - t_i/m)^2} \ ,$$

where $i = 1, 2, \ldots, k+1$. With multipath routing, each internal node $P_j$ selects as next node on its path a neighbor $Q$ (as specified by $direction$) for which $\mid distance(Q, C_i) - R_i \mid$ is minimized. With multibraid routing packets are locally flooded along the circular corridor selected by the source $s$.

**Generalizations.** Several other geometric families of curves can be used in a similar way, as for example families of ellipses.

**Sector Partitioning based Routing.** SPBR is an extension of CBR in which the space between the source node $s$ and the destination node $d$ is partitioned, and packets are sent through the partitions.

### 3.2. Color Graph Based Routing

With CGBR the nodes of the network $\mathcal{G}$ are colored, with no two nodes having the same color. This approach has been used with fixed infrastructure networks to deal with Byzantine faults [5]. One can also consider multipath routing for which the nodes on each path may have different colors, provided that these paths are color-disjoint.

## 4. Adaptive Multipath Routing

$AdaptivePath(s, d)$ is an adaptive multipath discovery algorithm that combines in parallel the Ford-Fulkerson Max Flow algorithm [11] with a network algorithm to find, incrementally, vertex-disjoint paths that link the source $s$ to the destination $d$.

Initially, $s$ broadcasts a query $req_s$ for neighbor list information. On receiving these lists, $s$ incrementally constructs paths linking $s$ to $d$. For this construction process, procedures $AddLink$ and $Update$ are used. The novelty of this construction is that its route discovery algorithm is resistant to malicious DoS attacks and its communication algorithm addresses adaptively Byzantine attacks. In particular, if there are no attacks then a single shortest path route is

used. With each Byzantine attack, the multipath route used is adaptively reconstructed to deal with this threat. The communication algorithm is activated as soon as a path is found so there is no unnecessary delays.

In the algorithm, $\mathcal{G}^* = \{(V^*, L^{*t})\}$ is a vertex-expanded version of the network graph $\mathcal{G}$. Each node $x_i$ of $\mathcal{G}$ corresponds to two nodes $x_i^+, x_i^-$ linked by $(x_i^+, x_i^-)$ in $\mathcal{G}^*$, and each link $(x_i, x_j)$ corresponds to a link $(x_i^-, x_j^+)$ in $\mathcal{G}^*$. Let $\tau$ be a time upper bound on single-hop round-trips and $[m]_x$ be the message $m$ together with a signature of node $x$ on it.

**AdaptiveMultiPath** $(s, d)$

Source $s$:

1. /* *initialize $\mathcal{G}^*$ with the neighborhood of $s$* */
   Let $\mathcal{G}^* = \emptyset$. For each $x_i$ in $neighbors(s)$ do $AddLink(s, x_i)$.
2. Let $flow = \emptyset$, $value(flow) = 0$, and $Update(flow)$.
3. Let $request = 1$, $ttl = initial\_ttl$.
4. While connection to $d$ has not been terminated do
   (a) /* *discover network by limited flooding* */
   While $value(flow) \leq request$ do
      i. Broadcast $req_s = ([id_s, current\_time_s, ttl]_s, 0)$ and set $timer_s = ttl * \tau$.
      ii. While $timer_s$ has not expired do
         A. If a valid reply $rep_j = ([id_j, id_s, time_j, hop_j; neighbors(x_j)]_j, count)$ is received then $AddLink(x_j, x_t)$ for all $x_t \in neighbors(x_j)$, and $Update(flow)$.
         B. If $value(flow) = request$ then use $flow$ for communications.
      iii. $ttl = 2 * ttl$.
   (b) Wait until $request$ increases.

**Procedure** $AddLink(x_i, x_j)$

1. /* *add nodes $x_i, x_j$* */
   Let $V^* = V^* \cup \{x_i^+, x_i^-, x_j^+, x_j^-\}$.
2. /* *add link $(x_i, x_j)$* */
   Let $L^{*t} = L^{*t} \cup \{(x_i^+, x_i^-), (x_j^+, x_j^-), (x_i^-, x_j^+)\}$.

**Procedure** $Update(flow)$

1. If $d^+ \notin V^*$ then return.
2. /* *an augmenting path* */
   For each sequence $(v_0, v_1, \ldots, v_n)$ such that:
   (a) $v_0 = s^-$, $v_n = d^+$, for all $0 < i < n$: $v_i \in V(\mathcal{G}^*)$, and
   (b) for all $0 \leq i < n$: $(v_i, v_{i+1}) \in \overline{flow}$ or $(v_{i+1}, v_i) \in flow$;
   do
   (a) /* *increase the flow along the path* */
   Let $p_1 = \{(v_i, v_{i+1}) \mid 0 \leq i < n\} \cap \overline{flow}$.
   Let $p_2 = \{(v_{i+1}, v_i) \mid 0 \leq i < n\} \cap flow$.
   Let $flow = flow + p_1 - p_2$.
   (b) Let $value(flow) = value(flow) + 1$.

Here $\overline{flow} = L^{*t} \backslash flow$. Since each edge in $\mathcal{G}^*$ has capacity 1, each flow in $\mathcal{G}^*$ is a set of edge-disjoint paths. Let $(s^-, x_1^+, x_1^-, \ldots, x_{n-1}^+, x_{n-1}^-, d^+)$ be a directed path in a flow in $\mathcal{G}^*$. The corresponding path in the network is $(s, x_1, \ldots, x_{n-1}, d)$. It is not hard to see that if $\{(s^-, x_1^+, x_1^-, \ldots, x_{n-1}^+, x_{n-1}^-, d^+)\}$ is a set of edge-disjoint paths then the corresponding paths $\{(s, x_1, \ldots, x_{n-1}, d)\}$ are vertex-disjoint, and vice-versa. We now discuss the protocol performed by each intermediate node.

Let $\Delta$ be a parameter that specifies how often an intermediate node will reply to repeated requests from the source node that have already been answered earlier. Also, let $\delta$ be an upper bound on time drift among hosts. Normally, we will choose the parameter $\Delta$ such that $\Delta \gg \delta$.

Intermediate node $x_i$ (including $d$):

1. If a valid $req_s = ([id_s, time_s, ttl_s]_s, hop)$ is received then
   (a) If $current\_time_i > lastsend_s + \Delta$ then broadcast $rep_i = ([id_i, id_s, current\_time_i, hop; neighbors(x_i)]_i, 0)$ and set $lastsend_s = current\_time_i$.
   (b) If $time_s > lastrecv_s$ and $ttl_s > hop$ then broadcast $req'_s = ([id_s, time_s, ttl_s]_s, hop+1)$ and set $lastrecv_s = time_s$.
2. If a valid $rep_j = ([id_j, id_s, time_j, hop_j; \ldots]_j, count)$ is received then
   (a) If $time_j > lastreply_j$ and $hop_j > count$ then broadcast $rep'_j = ([id_j, id_s, time_j, hop_j; \ldots]_j, count + 1)$ and set $lastreply_j = time_j$.
3. Loop.

**end of protocol**

1. *Network discovery.* In the protocol, replies from nodes are broadcast back to the source through multiple vertex-disjoint paths, which will include a good path (a path without malicious nodes) when the number of Byzantine faults are bounded.
2. *Packet verification.* Since the $hop$ counter is not signed by $s$, a request $req_s = ([id_s, time_s, ttl_s]_s, hop)$ is considered valid by a node $x_i$ if and only if the signature of $s$ is valid and $time_s + hop * \tau/2 > current\_time_i - \delta$. This prevents DoS replay attacks that use malicious $hop$ counts. Similarly, $x_i$ only considers $rep_j = ([id_j, time_j, hop_j; neighbors(x_j)]_j, count)$ valid if the signature of $j$ is valid and $time_j + count * \tau/2 > current\_time_i - \delta$. The use of $\Delta$ also reduces the effect of DoS attacks by limiting the rate that intermediate nodes reply to request for neighbor list. The bandwidth is minimized since nodes avoid duplicating broadcasts when the same information was recently broadcast.

3. *Path maintenance.* Assume that the source $s$ uses $t$ vertex-disjoint paths to communicate with the destination. When faults do occur beyond a certain threshold, $s$ will then switch to using $t + 1$ vertex-disjoint paths. Since this new set of paths is already constructed in the background, the delay caused by faults is minimized. Most of the time, there should be no delay at all. Further, in our algorithm, the set of vertex-disjoint paths are constructed incrementally, so even when delays are unavoidable, they are quite small.
4. *Neighbor discovery.* A cryptographic handshake protocol is used.
5. *Fault detection.* Byzantine faults are detected by exploiting the properties of broadcast channels as in [6].

## References


[1] A. Aho, J. Hopcroft and J. Ullman, *The Design and Analysis of Computer Algorithms*, Reading MA, Addison-Wesley, 1974.

[2] B. Awerbuch, D. Holmer, C. Nita-Rotaru and H. Rubens, "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures", *ACM Workshop on Wireless Security (WiSe'02)*, 2002.

[3] E.M. Belding-Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks", *IEEE Personal Communications Magazine*, pp. 46-55, 1999.

[4] J. Broch et al, "A performance comparison of multi-hop wireless ad hoc network routing protocols", *Proc. ACM MOBICOM*, pp. 85-97, 1998.

[5] Mike Burmester and Yvo G. Desmedt, "Secure Communication in an Unknown Network Using Certificates", *Advances in Cryptology - Asiacrypt '99, LNCS # 1716, Springer*, pp. 274–287, 1999.

[6] Mike Burmester and Tri van Le, "Tracing Byzantine faults in ad hoc networks", *Proc. IASTED, Computer, Network and Information Security 2003*, New York, pp. 43-46, 2003.

[7] Mike Burmester and Tri van Le, "Tracing faults in ad hoc networks", *3rd IFIP Networking conference, Networking 2004*, Athens, Greece, May 9-14, 2004.

[8] S. Capkun, M. Hambdi and J. Hubaux, "Gps-free positioning in mobile ad hoc networks", *Proc. Hawaii Int. Conf. on System Sciences*, 2001.

[9] C.C. Chiang et al, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", *Proc. IEEE SICON '97*, pp. 197–211, 1997.

[10] C. R. Davis, *IPSec: Securing VPNs*, McGraw-Hill, New York, 2000.

[11] L.R. Ford and D.R. Fulkerson, *Flows in Networks. Princeton University Press*, Princeton, NJ, 1962.

[12] P. Georgatsos and D. Griffin, "A management system for load balancing through adaptive routing in multiserve ATM networks", *Proc. IEEE INFOCOM*, 1996.

[13] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks", *Proc. ACM MOBICOM*, 2001.

[14] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks", *Mobile Computing*, ed. T. Imielinski and H. Korth, Kluwer Academic, pp. 152–181, 1996.

[15] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing in Mobile Ad Hoc Networks", *Proc. ACM/IEEE MOBICOM '98*, 1998.

[16] J. Kong et al., "Providing robust and ubiquitous security support for mobile ad-hoc networks", *Proc. IEEE ICNP*, pp. 251-260, 2001.

[17] L. Lamport, R. Shostac and M. Pease, "The Byzantine Generals problem". *ACM Transactions on programming languages and systems*, **4**(2), pp. 382-401, 1982.

[18] S. Murphy and J.J. Garcia-Lunca-Aceves, "An efficient routing protocol for wireless networks", *ACM Mobile Networks and Applications Journal*, pp. 182-197, 1996.

[19] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks", *Proc. INFOCOMM*, April 1997.

[20] C.E. Perkins and E.M. Royer, "Ad hoc on-demand distance vector routing", *IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.

[21] K. Sanzgiri et al, "A secure souting protocol for ad hoc networks", citeseer.nj.nec.com/sanzgiri02secure.html

[22] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, Inc, New York, 1996.

[23] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", *Proc. ACM/IEEE MOBICOM '98*, October 1998.

[24] H. Suzuki and F.A. Tobagi, "Fast bandwidth reservation scheme with multi-link and multi-path routing in ATM networks", *Proc. IEEE INFOCOM*, 1992.

[25] S. Yi, P. Naldurg, and R. Kravets, "Security-aware ad hoc routing for wireless networks", *Proc. ACM Mobihoc*, 2001.

[26] L. Zhou and Z. J. Haas, "Securing ad hoc networks", *IEEE Network*, 12(6):24-20, 1999.