A TRUST-BASED ROUTING PROTOCOL FOR AD HOC NETWORKS

Xiaoyun Xue and Jean Leneutre
INFRES department, ENST Paris - CNRS LTCI-UMR 5141, 46 rue Barrault 75634 Paris Cedex 13, France Tel: +33 1 45 81 71 98 Fax: +33 1 45 81 71 58
Xiaoyun.Xue, Jean.Leneutre@enst.fr

Jalel Ben-Othman
Prism laboratory, UVSQ - CNRS UMR 8144, 45 avenue des Etats-Unis 78035 Versailles Cedex, France
Jalel.Benothman@prism.uvsq.fr

Abstract Ad hoc networks are particularly vulnerable as compare to traditional networks mainly due to their lack of infrastructure. A malicious node can easily disrupt both the routing discovery phase and the data forwarding phase of a routing protocol if it is not secured enough. This paper proposes a new secure reactive routing protocol named TRP (Trust-based Routing Protocol) that relies on a distributed trust model managing trust levels. The model provides an estimation of trust level to each route to help a source node to chose the most secure one. Our security mechanism is protected and does not affect significantly the network performance.

Keywords: Ad hoc networks, trust model, secure routing, blackmail attack

1. Introduction

MANET (Mobile Ad hoc NETworks) are mobile wireless networks without fixed infrastructure. In an ad hoc network, each node is at the same time a router and a terminal, and is free to change its position with any speed and at any time. Many applications are possible: battlefields, conferences, urgency services...

Although the security requirements are different from one application to another, they are not negligible in most cases. Unfortunately, ad hoc networks are particularly vulnerable due mainly to their lack of infrastructure. Other reasons could be: high mobility, wireless links, limited bandwidths, lack of boundaries, short lifetime batteries and weak capacity of equipments.

Current ad hoc network security research works can be classified into two principal categories according to their main problematics: the distribution and management of keys and the authentication scheme, and the security of various routing protocols.

Traditional authentication and key management schemes are not applicable to ad hoc networks because they usually depend on some central server to establish trust relationships among nodes. In the contrary, an ad hoc environment requires a more distributed and robust solution. There are mainly three types of solutions proposed: emulation of a distributed certificate authority [Zhou and Haas, 1999] based on threshold cryptography; use of "trust chains" as done in PGP [J.P.Hubaux et al., 2001]; generation of one or more symmetric keys shared by the whole or a subset of the network [Asokan and Ginzboorg, 2000].

Most ad hoc routing protocols have been initially designed to deal with frequently changing topology, none of them have considered security issues in their design, suppose that it will be addressed later or by another layer's security mechanisms (for example 802.11's security mechanism or SSL/TLS). However a security mechanism at another layer is not sufficient because a mistake in routing choice can already welcome attacks, and security considerations must be integrated into routing protocols at the very first time.

A large part of newly proposed secure ad hoc routing protocols are applied to two reactive protocols: DSR and AODV, especially DSR. The original DSR is not secured at all, but since DSR controls every hop on a route and can work in a multi-routes mode, it is often used as a base of a secure routing protocol, just as what we have done in this work.

An ad hoc routing protocol has to be secured in both the routing discovery (and maintenance) phase and the data forwarding phase. In the first one, incorrect topology information should be forbidden and in the second, nodes who do not correctly forward data, intentionally or not, should be identified and excluded from the network.

This paper proposes a new secure reactive routing protocol named TRP based on DSR and a distributed trust model.

The rest of the paper is organized as follows: the section 2 discusses related work, the section 3 describes the proposition, the section 4 is dedicated to performance evaluation, residual vulnerabilities are discussed in the section 5 and finally our conclusion and perspectives are presented in the section 6.

2. Related work

Recent works trying to mitigate problems in the routing discovery phase usually integrate cryptographic mechanisms into routing protocols to guarantee the authentication and the integrity of routing control messages. The SRP (Secure Routing Protocol) protocol proposed in [Papadimitratos and J.Haas,

2002] uses only light-weight cryptographic operations limited to MAC calculations and is able to avoid a large variety of attacks.

For the security in the data forwarding phase, the protocol SMT (Secure Message Transmission) [P.Papadimitratos and Z.J.Haas, 2003] takes advantage of the presence of multiple routes through the distribution of data on several routes and the addition of a feedback mechanism to control further retransmissions. SMT is designed as a complement of SRP and it can effectively shield attacks even with up to 50% attackers. However, it induces overloads, and it is not able to isolate attackers.

Otherwise, by introducing a distributed supervision system using promiscuous mode to each node, it is possible to detect undesired behaviors made by malicious or failing nodes. To achieve this, every node memorizes its own observations [Marti et al., 2000], or as well as negative recommendations in form of alarm messages [Buchegger and Boudec, 2002, Lakshmi, 2001] or positive confirmations made by traffics [Michiardi and Molva, 2002]. The choice of routes is then carried out from trust values. However, the approach in [Marti et al., 2000] do not accept second-hand reputations, therefore its training time will be longer. And since [Buchegger and Boudec, 2002, Lakshmi, 2001] accept negative reputations, they are vulnerable to blackmail attacks (attacks where a malicious node send false accusations to tarnish reputations of honest nodes). Furthermore, the problem of authentication in supervision systems, an essential issue, is only partially addressed in [Buchegger and Boudec, 2002] and [Lakshmi, 2001].

Most of current works encounter the same difficulty which is to propose a robust but light-weight security mechanism. Our proposition TRP adopts a mechanism close to SRP to provide security to the routing discovery phase, and a light-weight supervision mechanism together with a trust model are integrated to secure the data forwarding phase. The exchanges of recommendations are protected by the same MAC used in SRP. In addition, a particular care has been taken in the definition of the reputation model used: it is based on the work presented in [Beth et al., 1994].

3. TRP protocol

The following assumptions are applicable to the rest of the paper: each node has at least a sufficient storage capacity for the supervision of a restricted part of the traffics forwarded by itself; each node has a minimum calculation capacity to carry out simple arithmetic calculations; transmission ranges of nodes are identical; each node has a unique identifier (ID), and it is possible to authenticate nodes; and lastly, there is at most one attacker on a route, and in the case there are more than one attacker on a route, they are not neighbors.

Routing discovery phase

TRP is largely inspired by SRP with regard to the security in the routing discovery phase. We suppose also that a sender node S trusts its destination node D (a Security Association with mainly a shared secret key $K_{S,D}$ exists between S and D).

Like SRP, the proposed security mechanism adds an additional header into routing control messages. The sender S initiates a routing discovery process by adding two integers, a sequence number Q_{seq} and a random number Q_{id}, to a normal DSR RREQ. A MAC using key $K_{S,D}$ is also added at the end. During the broadcast of a RREQ, intermediate nodes add their identities into the request, and continue to relay the request until it reaches its destination. The receiver D verifies the MAC, and sends back a RREP including the found route, Q_{seq}, Q_{id} and a new MAC calculated over the RREP. S verifies the MAC and then the route included can be written into S's cache.

This mechanism is able to resist to a great part of attacks (see [Papadimitratos and J.Haas, 2002] for a detailed description), anyway, it remains vulnerable:

- since RERRs are not authenticated, malicious nodes can invalidate correct routes by sending invented RERRs with spoofing; a fast moving attacker could thus invalidate a lot of routes;

- by using the promiscuous mode, a malicious node can refuse to add its own identity into a RREQ, so that the route will seem shorter; otherwise, if an attacker is a neighbor of a destination node, wrong routes can be created by spoofing IP addresses and finally, a loop can be inserted into a RREQ;

- wormhole attacks are not treated, selfish nodes and cooperating attackers neither.

The first attack can be avoided by adding an authentication mechanism to RERRs. The attacks in the second item could be detected by using the supervision mechanism presented later in the paper. The attacks in the last item are more sophisticated and are not addressed in TRP.

TRP uses above mechanisms, together with an additional header added to RREQs and RREPs to exchange trust informations between nodes, as described below.

Data forwarding phase

A mini supervision mechanism is integrated into each node. According to events that a node observed, trust values for each neighbors could be evaluated

dynamically. The goal of the mechanism is to let the source node concludes a trust degree for each route so that the most secure one will be chosen.

The trust values are calculated using the following reputation (trust) model:

Trust evaluation. Our reputation model is a variant of the trust model initially introduced in [B.K.R.Yahalom and T.Beth, 1994] and developed in [Beth et al., 1994] for its valuation part (it is modified to adapt it to the ad hoc context). The original model gives us the possibility to take into account various classes of trust relations: an entity is not absolutely trusted, but with regard to one or more specified tasks (for example, key generation, nondisclosure of secrets, or in our case, the routing function). Moreover, it allows valuation of trust relationships: from the numbers of positive and negative experiences an entity has assigned to another entity (with regard to a giving function), the former computes a trust level associated to the latter. Furthurmore, it is also a distributed trust model and it is possible to derive trust relationships from recommendations using transitivity.

Thus, three types of trust relationships are considered in our model:

- *direct trust relationship between neighbors*: that is valuated by positive and negative direct observation experiences;
- *indirect trust relationship between two nodes*: that is derived from direct trust relationships using transitivity;
- *trust relationship between a node and a route*: that is computed using direct and indirect trust values.

All these trust values are taken in $\{-1\} \cup [0, 1[$, and when the value -1 is associated to a node, it means this node is considered as malicious (or failing).

An initiator of a RREQ will obtain in each returned RREP a series of direct trust values given by the nodes on the route. With these direct values, the sender of the RREQ is then able to evaluate the indirect trusts for the nodes on the route for which it has no direct trust values. Finally, a trust value of the route will be computed in order to avoid encountering malicious nodes during the data forwarding phase.

Direct trusts. All direct trust values are initialized to 0 by default. But since the model is totally local, nodes are free to initiate some trust values to some wanted values when there exists some pre-established trust relationships.

The evolution of the trust value of the node n_i on the node n_j is given by the following formula:

$$C^D_{n_i,n_j}(t) = \begin{cases} -1 & \text{if } p_{n_i,n_j}(t) < 0 \\ 1 - \alpha^{p_{n_i,n_j}(t)} & \text{otherwise} \end{cases}$$

here $\alpha \in (0,1)$, (higher is α, slower a direct trust goes from 0 to 1 with time, vice versa) and $p_{n_i,n_j}(t)$ depends on the number of positive experiences $p^{+}_{n_i,n_j}(t)$ (the number of good behaviors) and the number of negative experiences $p^{-}_{n_i,n_j}(t)$ (the number of bad behaviors) of n_j observed by n_i until time t. The value of $p_{n_i,n_j}(t)$ is defined as:

$$p_{n_i,n_j}(t) = p^{+}_{n_i,n_j}(t) - \beta * p^{-}_{n_i,n_j}(t)$$

where β ($\beta > 1$) is a parameter which allows the modulation of the importance of negative experiences (greater is β, larger is the influence of negative experiments). β is introduced so that a certain number of faults may be tolerated. Both α and β should be relatively high to keep the efficiency of the model. According to the form of the formula, we can see that:

- if a node always behaves well, its trust value will rapidly increase to 1;
- if a node is moderately malicious or failing, its trust value will be stable;
- if the node is malicious or quite failing, then it will immediately become untrusted.

Indirect trust. Since a source node has not inevitably a neighborhood relationship with all nodes on a route, it is sometimes needed to derive indirect trust values using recommendations, when a RREP is returned to a source. For example, we consider a route made up of $k+1$ nodes, where n_i is the ith node and n_0 and n_k are respectively the source node and the destination node, the indirect trust values are defined as:

$$C^{I}_{n_0,n_{k-1}} = \begin{cases} -1 & \text{if } C^{D}_{n_0,n_k} or C^{D}_{n_k,n_{k-1}} = -1 \\ 1 - (1 - C^{D}_{n_k,n_{k-1}})^{C^{D}_{n_0,n_k}} & \text{otherwise} \end{cases}$$

and for $1 \leq i \leq k-2$

$$C^{I}_{n_0,n_i} = \begin{cases} -1 & \text{if } C^{I}_{n_0,n_{i+1}} or C^{D}_{n_{i+1},n_i} = -1 \\ 1 - (1 - C^{D}_{n_{i+1},n_i})^{C^{I}_{n_0,n_{i+1}}} & \text{otherwise} \end{cases}$$

For our propose, we only need to consider the indirect trust relations starting from n_0 but our definition could easily be extended.

The indirect trust values are defined so that:

- if one of the direct trust values used in the recommendation chain indicates that one of the nodes on the route is a potential attacker (the corresponding direct trust value is -1), then the afterward derived indirect trust values are no more relevant (will equal to -1);

- if $C^D_{n_{i+1},n_i}$ is based on a experience level p, then $C^I_{n_0,n_i}$ is based on the experience level $p * C^I_{n_0,n_{i+1}}$, because $C^I_{n_0,n_i} = 1 - (1 - (1 - \alpha^p))^{C^I_{n_0,n_{i+1}}} = 1 - \alpha^{p*C^I_{n_0,n_{i+1}}}$.

An optimization could be to check beforehand if there is a node n_l between n_i and n_k with which n_0 has a direct trust value. If so, the $C^I_{n_0,n_i}$ $(1 \leq i < l)$ can be calculated starting from $C^D_{n_0,n_l}$ instead of $C^D_{n_0,n_k}$.

All indirect trust values are deleted after the derivation of the trust value of the route so that influences of malicious recommendations are limited.

Route trust. According to the principle that the security level of a whole system corresponds to the security level of its weakest component, our trust level of a route corresponds to the source's lowest trust level on its intermediate nodes. For example, the trust value of a route n_0, ..., n_k equals:

$$C^R_{n_0,\ldots,n_k} = min^k_{i=1}(C_{n_0,n_i})$$

where

$$C_{n_0,n_i} = \begin{cases} C^D_{n_0,n_i} & \text{if } p^+_{n_0,n_i} + p^-_{n_0,n_i} \geq 1 \\ C^I_{n_0,n_i} & \text{otherwise} \end{cases}$$

The main difference between [Beth et al., 1994] and our model is that we tolerate some faults (we added the -1 value to keep a rating for untrustful nodes), and for our particular objective, we introduced trust values for routes.

TRP implementation details.

Supervision system. In order not to overload nodes, we chose a restricted supervision mode: a node does not supervise all traffics in its neighborhood, but only supervises traffics passed by itself. We called this mode "supervision on routes" comparing to the mode "supervision in the neighborhood" in all other works.

Each node maintains a trust information table memorizing the number of good behaviors p^+, the number of bad behaviors p^- and the direct trust value ("rating") of each neighbor observed. In the data forwarding phase, for each data packet sent or forwarded, the sender or the forwarder n_i save a copy of the packet in its buffer, and then supervises the action of the next node n_{i+1} (when n_{i+1} is not the destination). Depending on whether or not n_{i+1} correctly forwards the data within a limited time period, n_i increments the value of $p^+_{n_i,n_{i+1}}$ or $p^-_{n_i,n_{i+1}}$. The rating will also be updated if necessary.

We suppose of course that the authentication of neighbors is performed.

Modifications to RREQs and RREPs. The header of SRP is extended to include a new table memorizing the trust values reported by intermediate nodes. Every intermediate node adds its direct trust value on the former node into the request.

All trust values collected in a RREQ should be send back to the source node within a RREP, and the MAC in the RREP covers equally their integrity. Furthermore, when a RREP is received by an intermediate node n_i, $C^D_{n_i,n_{i-1}}$ is verified by n_i to ensure that $C^D_{n_i,n_{i-1}}$ has not been modified by one or more nodes among $n_{i+1},...,n_k$. A packet with a modified trust value should be dropped, so that a attacker has no possibility to modify any value not reported by itself.

As an option, every intermediate node n_i checks also if $C^D_{n_i,n_{i+1}}$ equals to -1, and if it is the case, it stops to forward the RREP. This option has been implemented in an optimized version of TRP called TRP$^+$.

When a RREP is received by its source, the latter checks the MAC. If the MAC examination succeeds, the direct trust values included in the RREP will be used to calculate the necessary indirect trust values and a route trust value will be computed.

The only way for a malicious node n_i to introduce a wrong trust value is to give a wrong $C^D_{n_i,n_{i-1}}$ to a RREQ. If $C^D_{n_i,n_{i-1}}$ is too low, the route will not be chosen so that the attacker will not be on an active route; if $C^D_{n_i,n_{i-1}}$ is too high, because there is a great probability that trust levels $C^D_{n_0,n_i}$ and $C^D_{n_{i+1},n_i}$ are low or negative and also because the trust value of the route only depends on the minimum value of all the trust values, the probability that the trust level of the route increases is low. Even in an extreme situation, where all intermediate nodes are malicious, the destination which is trusted by the source can still give a bad reputation to n_{k-1}, so that the route will not be chosen.

Route management. Since each route has a specific trust value, routes accepted by a node should be stored separately into its route cache, and all routes will expire after a timeout (which is important for refreshing trust levels of routes). Routes with a trust value equal to -1 are not stored into caches. Moreover, if all nodes on a given route are completely included in another route resulting from a same RREQ, the second route will be discarded.

Route choice. Each data packet must obtain a route before its sending. Two strategies are possible: always choose the route with the highest trust value; or set a threshold as the lowest acceptable trust level, and chose the shortest route among all routes having a trust level greater than the threshold. The first strategy emphasizes the security requirement, while the second is a compromise between the security and the network performance.

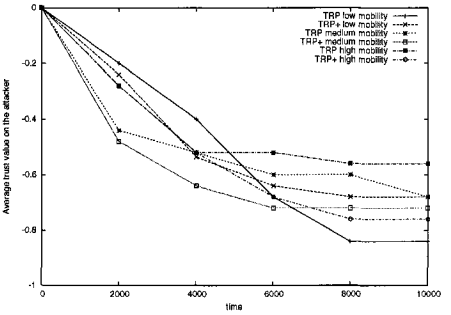

Figure 1. Average trust value on the attacker

4. Performance evaluation

Some simulations have been carried out under NS-2 using OpenSSL's library for cryptographic functions. TRP is implemented on the DSR module.

The simulation network contains 24 normal nodes and a attacker. The attacker modifies a packet when forwarding it, and it will never send back (initiate or forward) a RERR whether there is a broken link or not.

Three mobility scenarios have been tested:

- low mobility - 100s as pause time and 2m/s as maximum speed;
- medium mobility - 20s as pause time and 5m/s as maximum speed;
- high mobility - 5s as pause time and 20m/s as maximum speed.

We use FTP as application, with 22 random CBR sources and with a packet rate of 2 packets/s. The simulation time is 10,000s and the simulation range is 700m*700m. $\alpha = 0.75$, $\beta = 10$ and the promiscuous buffer size is 30.

The simulations are realized on TRP as well as on TRP$^+$.

At first, the figure 1 shows that the average direct trust value on the attacker decreases with time, whatever is the mobility scenario. TRP$^+$ is generally better than TRP, because in TRP$^+$ more nodes could have a forwarding relationship with the attacker.

The figures 2, 3 and 4 show us that the number of successful attacks could be stabilized after a time with TRP, and this time will still be shorter with TRP$^+$. Furthermore, as we could foresee, stronger is the mobility, better are the results: a frequently changed network topology can help nodes to detect the attacker.

We also observed that the average route length increased not more than 3% in all simulations.

In term of communication overhead, no new message is added but only the size of RREQs and RREPs are slightly increased due to the addition of

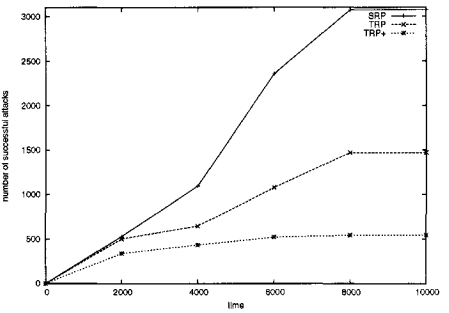

Figure 2. Losses comparison: low mobility

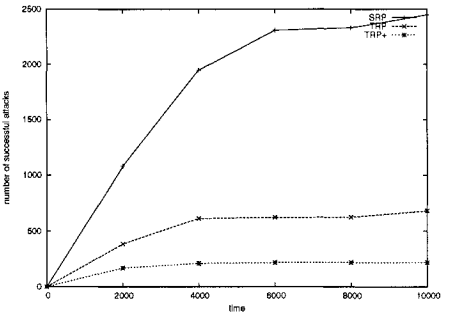

Figure 3. Losses comparison: medium mobility

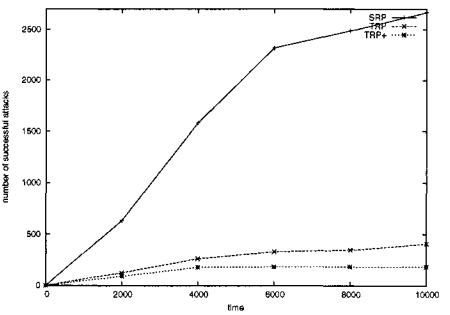

Figure 4. Losses comparison: high mobility

the new trust header. However, the routing overload increases considerably comparing to DSR because of the deletion of many DSR optimizations and the refreshments of caches: since routes expire after a lifetime, we have to more often initiate RREQs to refresh routes.

5. Residual vulnerability

Firstly, we need a training phase before TRP can really be efficient; secondly, we have not considered selfish nodes: in our solution, a node can simply give a "-1" as a trust value to save its energy; thirdly, cooperate attackers are not treated; and lastly, spoofing of IP addresses should be stopped by using some authentication schemes.

6. Conclusion and future work

We presented in this article a new solution relying on a trust model which allows us to secure a reactive ad hoc routing protocol. The proposition can counter most of attacks during both the routing discovery phase and the data forwarding phase and it seems more suited to ad hoc networks with a long lifetime and a frequently changing topology.

Contrary to other approaches adopting a similar reputation system [Marti et al., 2000, Buchegger and Boudec, 2002, Michiardi and Molva, 2002], our solution protects its reputation exchanges to be not vulnerable to blackmail attacks. Compare with a solution which transfers data on multiple routes [P.Papadimitratos and Z.J.Haas, 2003], our solution has disadvantages such as needing a training phase before becoming operational and needing authentications of intermediate nodes, but it has the advantages of not requiring Ack messages and isolating attackers.

As an immediate work, we plan to:

- optimize the trust calculation in particular by calibrating parameters α and β;
- carry out more intensive simulations, especially by considering more attackers, and by adopting less systematic attackers' behaviors;
- extend the supervision to the neighborhood in order to reduce the training phase.

In the future, we will study if our solution can be adapted to other reactive protocols, or even to proactive protocols. And lastly, we hope to tackle the complex problem of key management and authentication scheme.

References

[Asokan and Ginzboorg, 2000] Asokan, N. and Ginzboorg, Philip (2000). Key-agreement in ad-hoc networks. *Computer Communications*, 23(17):1627–1637.

[Beth et al., 1994] Beth, T., Borcherding, M., and Klein, B. (1994). Valuation of trust in open networks. In *Proc. 3rd European Symposium on Research in Computer Security – ESORICS '94*, pages 3–18.

[B.K.R.Yahalom and T.Beth, 1994] B.K.R.Yahalom and T.Beth (1994). Trust relationships in secure systems - a distributed authentication perspective. *Computer Systems*, 7(1):45–73.

[Buchegger and Boudec, 2002] Buchegger, Sonja and Boudec, Jean-Yves Le (2002). Cooperative Routing in Mobile Ad-hoc Networks: Current Efforts Against Malice and Selfishness. In *Lecture Notes on Informatics, Mobile Internet Workshop, Informatik 2002*, Dortmund, Germany. Springer.

[J.P.Hubaux et al., 2001] J.P.Hubaux, L.Buttyan, and S.Capkun (2001). The quest for security in mobile ad hoc networks. In *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*, Long Beach, CA, USA.

[Lakshmi, 2001] Lakshmi, Venkatraman (2001). Secured routing protocol for ad hoc networks. Master thesis, University of Cincinnati, Computer Science.

[Marti et al., 2000] Marti, S, Giuli, T, Lai, K, and Baker, M (2000). Mitigating routing misbehavior in mobile ad hoc networks. *Proceedings of MOBICOM*.

[Michiardi and Molva, 2002] Michiardi, Pietro and Molva, Refik (2002). Core: A COllaborative REputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Communication and Multimedia Security 2002 Conference*.

[Papadimitratos and J.Haas, 2002] Papadimitratos, Panagiotis and J.Haas, Zygmunt (2002). Secure routing for mobile ad hoc networks. *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*.

[P.Papadimitratos and Z.J.Haas, 2003] P.Papadimitratos and Z.J.Haas (2003). Secure data transmission in mobile ad hoc networks. In *Proceedings of the 2003 ACM workshop on Wireless security table of contents*, pages 41–50.

[Zhou and Haas, 1999] Zhou, Lidong and Haas, Zygmunt J. (1999). Securing ad hoc networks. *IEEE Network*, 13(6):24–30.