2HARP: A SECURE ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS

Po-Wah Yau and Chris J. Mitchell
Mobile VCE Research Group
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
P.Yau@rhul.ac.uk, C.Mitchell@rhul.ac.uk

I. INTRODUCTION

Mobile ad hoc networks are a class of networks based on wireless technologies. An ad hoc network is a permanent or temporary collection of nodes that can communicate with each other. The distinguishing properties are that there is no pre-existing infrastructure, that there is no central entity to provide network administration services, and that end-to-end communication may require routing information via several intermediate nodes. This is why ad hoc networks are sometimes referred to as multi-hop networks, where a hop is a direct link between two nodes. If wireless communication is being used, then two nodes are within one hop of each other if they lie in each other's transmission range.

Mobility introduces a major design constraint not present in wired networks, namely the need for energy efficiency. The consequence of this is that network services must be efficient, and must also take account of nodes which do not have enough energy to participate. An example of where this can give rise to a security threat is provided by the routing service in the network layer. Routing is a distributed operation in ad hoc networks, where every node can act as a router. In a previous paper on threat models [19], failed nodes are defined as those nodes which do not have enough resources to generate or forward data packets, and such nodes may often occur through battery exhaustion.

However, there is a related class of threats to routing arising from selfish nodes, who try to exploit the routing protocol to their own advantage. The primary motivation for their unhelpful behaviour is to enhance their own performance and to save their own energy resources. In ad hoc networks, the main threat from such nodes comes from the selfish dropping of packets, which can severely affect the performance of the network [15]. Selfish nodes may also attempt to gain a better quality of service by reserving routes and bandwidth by not responding to routing messages. The key difference between failed and selfish nodes is that selfish nodes have the ability and resources to forward packets, whereas failed nodes do not.

The motivation for selfish behaviour in a commons such as an ad hoc network is discussed in [8]. If a node feels that it can gain more benefit by behaving selfishly and refusing to perform services then it will do so. Current network technology typically assumes, quite correctly, that most devices cooperate in routing information. However in the future this may not be the case. This is due, in part, to network devices becoming smaller, more personal, and customisable through software downloads.

Hence, one requirement for the network layer would be to prevent selfish behaviour, or to at least detect selfish nodes so that the network can react appropriately. Prevention, by forcing nodes to forward data packets, is extremely difficult. Thus detection of selfish behaviour is a problem which has been the subject of much recent research interest [3], [4], [16].

A second requirement is to detect failed nodes and to exclude them, so that they are not used whilst they are failed. Failed nodes are of no use for routing, but the protocol should support their ability to recharge and come back online.

This paper[1] presents a hybrid ad hoc routing protocol which uses a 2-hop acknowledgement mechanism to detect and react to both failed and selfish nodes. The protocol then uses a route selection method to increase the network's tolerance of selfish and failed nodes. The protocol is hybrid as it uses both proactive and reactive mechanisms [20]. In particular, each node proactively maintains a topological view of all nodes which are a maximum of two hops away. When a node requires a route to a destination node not within two hops, a reactive route discovery mechanism is used.

The following terms are used in this document, but may be used differently elsewhere. A *node* is a device with a network interface that is participating in routing in a mobile ad hoc network. It may or may not be mobile, and may also be part of another network. It is important to realise that a node can actually be a large network, or it could be just a single mobile device such as a mobile phone. An *originator node* is a node which originates a data packet, intended for

[1]The work reported in this paper has formed part of the Networks & Services area of the Core 2 Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com, whose funding support, including that of EPSRC, is gratefully acknowledged. Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE.

a certain *destination node*. A node is a *neighbour node* of another node if it is only one hop away, i.e. within direct transmission range. Likewise, a *2-hop neighbour node* is a node which is two hops away. If the destination node is not a neighbour node of the originator node, the data packet will have to traverse a multi-hop route consisting of *intermediate nodes*. In a specific scenario, the *sending node* is the last node to send the data packet. Data packets are unicast in this protocol except where indicated otherwise. A *routing message* is any packet used by the routing protocol to affect routing information.

II. The 2-Hop Acknowledged Routing Protocol

The 2-Hop Acknowledged Routing Protocol (2HARP) is based on the Zone Routing Protocol (ZRP) [7]. Proactive routing is performed locally, and reactive routing is used to discover routes outside of the proactive routing zone. The rest of this section gives an overview of 2HARP, after outlining some important issues and assumptions. Full details of the final scheme will depend on the results of ongoing simulation studies.

A. Important Issues and Assumptions

An adversarial environment is assumed, where every neighbour node is potentially failed or selfish, e.g. a multi-domain ad hoc network where nodes do not trust each other.

The threat that the 2HARP protocol addresses is posed by those selfish nodes which refuse to forward data for other nodes, but continue to send their own data into the ad hoc network for routing. The other type of selfish behaviour, not dealt with by 2HARP, is where a node does not participate in any operation in the ad hoc network. This threat cannot be tackled solely within the network layer, where the end user's view is of paramount importance. Possible solutions are a policy whereby a user will not receive the full range of services if they do not fully involve their ad hoc device in the network, or the user will receive billing discounts if they do play a full part in supporting network operation.

Identity and address resolution is an important issue which is outside the scope of this paper. In order to describe the routing protocol more easily, it will be assumed that each node has only one network interface operating using 2HARP, and therefore has only one identifiable address.

The use of a single identifier implicitly reveals another important assumption. We suppose that each node cannot connect to the network using a false identifier, or masquerade as another node, i.e. the ownership of an address is trusted. This may require peer-to-peer entity authentication in the underlying data-link layer. How this might be efficiently achieved in a multi-domain ad hoc environment is a major issue in itself, and while some suggestions are given, this is not the main purpose of this paper.

The paper also assumes that each node has the capability to digitally sign each packet it sends and also verify the signatures of each other node in the network. This might, for example, be achieved by equipping each node with a public key certificate signed by a common certification authority, and exchanging certificates in the messages used by the Neighbour Sensing Protocol (see below). Digital signatures provide origin authentication, so that a node can always be sure that it has received a data packet from the claimed source. This is a fundamental service which is needed to enhance the security techniques in the 2HARP protocol. Digital signatures can also provide data integrity. The specific details of the mechanisms used are outside of the scope of this paper.

One further assumption related to digital signatures is that each node will have a private/public key pair. For the moment, the paper also assumes that a public key is bound to only one node. Key distribution and management are not the main subject of this paper.

Attacks are possible on the OSI/ISO physical and data-link layers [14]. This paper assumes that either security mechanisms exist in the lower layers to make such attacks too expensive to perform, or that such attacks are unlikely. If this were not the case, then the availability of any routing protocol would suffer severely from denial of service attacks.

B. Fundamentals of the Protocol

Each node following the protocol maintains a Routing Table and an Acknowledgement Table. The Routing Table contains information about the node's neighbours, including which neighbour to use to route information to a certain destination node. The Acknowledgement Table is used to store information about packets which are awaiting an acknowledgement. The two tables are populated during the operation of the routing protocol, which is divided into three phases — neighbour sensing, route discovery and data packet sending.

C. The Neighbour Sensing Protocol

The proactive Neighbour Sensing Protocol is a modification of the neighbour sensing function of the Optimised Link State Routing (OLSR) protocol [5], which allows nodes to discover who is in their 2-hop neighbourhood. This is achieved by all nodes advertising their 1-hop neighbourhoods to each other. Associativity is important so new nodes are only accepted as neighbours if they can demonstrate that they can maintain a permanent link for a certain period of time. This part of the protocol is proactive, as each node has to periodically broadcast updates indicating any changes to their 1-hop neighbourhood.

D. The Route Discovery Cycle

The second phase is the reactive Route Discovery Cycle, used to discover new routes when they are needed. If a node requires a route to a destination which is not within the node's 2-hop neighbourhood, it broadcasts a Route Request message, which contains the addresses of both the originating node and the destination node. The 1-hop neighbours will reply to the node with a Request Acknowledgement message stating which of their neighbour nodes (i.e. the originator

node's 2-hop neigbours) will be able to propagate the Route Request further. Those 1-hop neighbours who replied also forward the Route Request to the 2-hop neighbours they advertised. These 2-hop neighbours will perform the same steps and send a Request Acknowledgement to the 1-hop node from which they received the Route Request. The 1-hop node will then forward on the acknowledgements to the originator node. These acknowledgements form the basis of deciding which 1-hop neighbour nodes to use where multiple routes exist. Route Requests are only processed if they are received from, or are for, responsive neighbours. See section II-F for more information on how this is used.

Every node along the route uses the 2-hop acknowledgement to determine if its neighbour is selfish or failed, until the Route Request reaches the destination node. See figure 1 for an illustration of a general example. Here a Route Request, Route Reply or data packet has been sent by node A, destined for node F, where all nodes know the route to send packets for node F. All nodes must respond with a signed acknowledgement (Ack^x) where x is the node which signed the message. Each message is prefixed with a number to demonstrate the order in which the messages will be sent. Only the destination node itself can generate a Route Reply, which is propagated back towards the originator node using the same acknowledgement mechanism as for Route Requests.

If a node does not receive a Request Acknowledgement message within time t, it may resend the Route Request up to a maximum of s times. It is important that the node performs a exponential backoff by doubling the value of t every time. The receipt of a duplicate Route Request is an indication that the neighbour node did not receive the Request Acknowledgement, so a 1-hop node will know it will have to rebroadcast the acknowledgement.

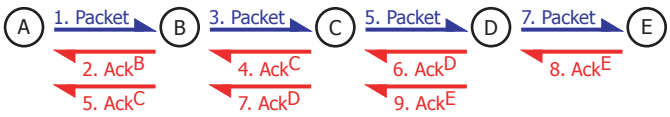

Fig. 1. A generalisation of the 2-hop acknowledgement mechanism.

As Route Requests are received, nodes can extract the orginator node's address to record a reverse path, i.e. the node from which the Route Request was received is recorded as the next hop towards the originator node. The converse is true when receiving Route Replies, where the node from which the Route Reply was received is recorded as the next hop towards the destination node. This mechanism is also used in the Ad Hoc On-demand Distance Vector (AODV) [17] routing protocol. Another feature of AODV which is employed in 2HARP is the use of unique sequence numbers to discard duplicate or old Route Requests so that routing loops do not occur.

E. Sending Data Packets

Finally, the third phase involves sending data packets using the discovered routes. This uses the same acknowledgement mechanism as in Route discovery, where a node that is 2 hops down the route must produce an acknowledgement that the data packet was received, and send it 2 hops upstream. Data packets are only forwarded if they have been received from responsive nodes.

Once a node has received a data packet with a specific (originator node, destination node) pair, it should always use the same route for all traffic between these two nodes. If a node receives a packet and discovers that the default route has broken (i.e. its link with the 1-hop neighbour node no longer exists), the node can use an alternative route if permitted in the data packet header and the packet has not been rerouted before. If there is no alternative route or the data packet is not to be retransmitted, then the node must perform route maintenance. Here, the node upstream of the break will have to generate a Route Error message and route it back towards the originator of the data packet. Upon verification of the Route Error, nodes can decide whether or not to discover a new route.

F. Coping with Selfishness

The following route selection mechanism is proposed for use with 2HARP when multiple routes exist to one target destination. A node should try each alternative route when the default route fails, using the reputations of each neighbour node as an indicator of which 1-hop node to choose.

A node that requests its neighbour to send a Route Request message or data packet will give the 1-hop neighbour node $retry$ opportunities to do so before it will decrease that neighbour's reputation. While the reputation remains positive, the node may try to use the neighbour node, but it will give it less $retry$ attempts, according to the neighbour's reputation. If the reputation of the 1-hop neighbour node becomes less than zero, then that neighbour will be marked as 'unresponsive'.

When no acknowledgements have been missed by a 1-hop neighbour for the past y consecutive messages sent, the node should increase the 1-hop node's reputation. After a certain number (i) of messages, the reputation of the neighbour should be reset to the original value[2]. Negative behaviour has a greater effect than positive behaviour, to allow 2HARP to detect negative behaviour more quickly.

In order to take into account the possibility that an unresponsive node may be a failed rather than a selfish node, the node will wait an amount of time k before giving the excluded neighbour another $retry$ = 1 opportunities, if it has packets to route through the unresponsive node. The node should exponentially backoff after every unsuccessful

[2]This makes it more difficult for a malicious node to build up a good reputation to enable it to behave selfishly without being excluded for a sustained period.

attempt at trying to get the neighbour to forward a packet. Thus failed nodes, and even 'repentant' selfish nodes, are permitted to become involved in routing again should they overcome their previous problem with forwarding packets. The period of time k is doubled so that resources are not repeatedly wasted testing a node which is still failed or selfish. Thus it is important that the Routing Table entry for an unresponsive 1-hop node is not immediately deleted.

III. Protocol Design Issues

This section discusses the mechanisms used to achieve the main aim of the protocol, namely detecting and reacting to failed and selfish nodes. Future work will also address performance issues including network flooding of Route Requests, intermediate nodes replying to Route Requests, and the choice of a hybrid scheme.

A. Detecting Failed and Selfish Nodes

The main design criterion for 2HARP was that it should detect and react to selfish behaviour in a reliable way. Without a proactive scheme the 2-hop acknowledgement mechanism would fail. A node will need to know who its 2-hop neighbours are, in order to know from whom to expect 2-hop acknowledgements. However, the proactive Neighbour Sensing Protocol by itself does not completely solve this problem, so additional mechanisms are needed.

The digital signature mechanism has been included to provide origin authenticity, where the process has been made more robust through the use of unique sequence numbers. The primary use of unique sequence numbers is to provide protection against replay attacks. They can also be used to further decrease the likelihood of a successful masquerade attack, analogously to the RAND value in GSM authentication [6]. Thus good management of sequence numbers in 2HARP will ensure that a node's sequence numbers monotonically and randomly increase. This additional complexity will help to discourage potential masquerade attacks.

Another threat arises where the 1-hop node is responsive, but the 2-hop node is unresponsive when forwarding data packets. Thus, the 0-hop node will receive an acknowledgement from the 1-hop node, but not the 2-hop node. This should not be possible in 2HARP, as the 1-hop node should detect that the 2-hop node is unresponsive from its own communications with the 2-hop node, which would result in it being removed from the list of responsive nodes it periodically distributes to its 1-hop neighbours. However, there is a small possibility that a 1-hop node does not know that a 2-hop node is unresponsive if, for example, it has never asked the 2-hop node to forward any packets. In the case where a 0-hop node sends a packet for forwarding, and the 1-hop node only has the one 2-hop neighbour to whom to forward the packet, the 2-hop will be marked as unresponsive by the 1-hop node, and the 1-hop node will be labelled as unresponsive by the 0-hop node. This has different consequences for each of the parties involved. The 0-hop node will gain from the situation as it knows it cannot use that 1-hop node to send packets to the destination. However, the 1-hop and 2-hop nodes' communication with the 0-hop node will be severely restricted while they are marked as unresponsive. Therefore, it is the responsibility of each node to make sure it knows which of its neighbours are being unresponsive, and to advertise these within the Neighbour Sensing Protocol. Further research will be needed to see if this can be achieved by other means.

A selfish node can also decide not to send messages to indicate that it has discovered new neighbour nodes. Currently, there is no explicit mechanism to prevent this attack succeeding. The impact of this threat will depend on the density of nodes as, the more nodes there are, the less effect the selfish node's actions will have. Again, where failed nodes are unable to send Neighbourhood Update messages, neighbour nodes will in any case be unable to use failed nodes as a next hop to new 2-hop nodes.

It would be disadvantageous for a selfish node to not advertise unresponsive or expired neighbours, as another neighbour could route through the selfish node and still expect acknowledgements from the non-cooperating or non-existent 2-hop neighbours.

B. Reacting to Failed and Selfish Nodes

The ideal situation would be a protocol where ad hoc nodes could trade energy for every data packet forwarded. There would then be no motivation for selfish behaviour, and failed nodes may also be of little consequence. However, the technology to achieve this does not exist.

The problem with untrusted, multi-domain ad hoc networks is that one node cannot control how a message is sent by another node; it is only possible to gain assurance that it has been sent. In 2HARP, this assurance is gained through the receipt of 2-hop acknowledgement messages. The reputation mechanism used is a simple technique utilised by many protocols, where it is rarely described as a reputation scheme, as the term itself was introduced after these mechanisms were devised.

Preventing the participation of selfish nodes is a local access control mechanism adapted for use in 2HARP, where failed and selfish nodes are marked as unresponsive and are therefore unauthorised to participate. By preventing selfish nodes from sending data packets, the protocol tries to change their objective from saving power to exchanging information, the essence of networking. A node marked as unresponsive cannot send Route Requests, Replies, or data packets, and thus cannot be the target of any Route Requests. However, the protocol allows data packets to be sent to a selfish node. The reasoning behind this is to cater for the possibility of failed nodes, which may not have enough power to send packets, but may still receive packets.

There is a delicate trade-off between the existence of selfish and failed nodes. Ideally, the protocol would not even allow information to reach selfish nodes but, to achieve this, the protocol would need a mechanism which could discover a neighbour node's energy levels in a secure way. This would

prevent a selfish node pretending that it does not have enough energy to forward packets.

Failed nodes which cannot forward packets are of no use in ad hoc networks. Hence, 2HARP detects them and stops other nodes from using them, to allow failed nodes to recover without the additional burden of having to deal with Route Requests, etc.

However, there are still unaddressed issues in 2HARP. A node can be selectively selfish towards some of its neighbours, as reputation is a local calculation. Also, if a selfish node does not object to having its communication links severed, there is nothing that 2HARP currently does to force a selfish node to forward packets. These issues will be discussed further in section VI.

IV. Security Analysis

This section presents a threat analysis summary of 2HARP, in the context of the other two threat classes in the threat model of [19], namely badly failed and malicious nodes. Badly failed nodes can introduce false routing messages, which are still correctly formatted, but contain false information. The threat of false routing messages can also come from malicious nodes, who aim to deliberately disrupt the correct operation of the routing protocol, denying network services if possible. There are no explicit mechanisms within 2HARP to protect against these threats; further work, discussed below, will be needed to design mechanisms to do so.

Any false routing messages sent in the Neighbour Sensing Protocol will cause nodes to waste resources, and misdirect traffic by setting up false routes. However, 2HARP differs from other routing protocols in that false information will be confined to a 2-hop neighbourhood, preventing its spread throughout the whole network. As the Routing Tables are periodically updated using fresh information, any false route information only remains during the period in which a node is sending false routing messages.

The use of digital signatures makes it difficult for malicious nodes to introduce false 2-hop nodes, as in order to maintain the pretence of the non-existent 2-hop nodes, the malicious 1-hop neighbour would have to generate private/public key pairs for each of those 2-hop neighbour nodes and respond to any messages using the private key associated with a non-existent 2-hop node to create a signature. This will be computationally expensive and will therefore use up a significant amount of resource, maybe more than will be wasted by the 1-hop nodes that the malicious node is trying to attack.

The effect of false messages is mitigated by the use of explicit Neighbour Sensing messages, the fact that they are only trusted if they have been received several times within a certain number of $update$ periods, and the requirement for digital signatures. These mechanisms also help to prevent the realisation of other threats such as wormhole attacks [10, p2]. However, a badly failed or malicious node will be able to send routing messages to cause other attacks. For example, Route Requests for nodes which do not exist, or for nodes known to be unresponsive, could be generated and sent. This is an example of a denial of service attack, also known as the sleep deprivation torture attack [18, p4]. This is an important threat as the false Route Request will propagate and cause nodes to waste resources trying to process it. There is currently no mechanism in 2HARP to prevent this from happening, since it has been designed to detect non-forwarding behaviour; this is a topic for further research.

A malicious node could send a Route Error message in response to a data packet it receives. This will force the originator of the corresponding data packet to believe that delivery to the destination has not been possible. 2HARP does not prevent this but, again, a node may not wish to send packets along a route involving a malicious node. 2HARP limits the influence of this attack because a malicious node needs a current data packet in order to generate a valid Route Error message. Thus the malicious node has to be involved in the route of the data packet.

Finally, messages could be sent by a malicious node using a false source address, to masquerade as another node. This could be addressed if the public key of each node is securely bound to their address, and this will also be the subject of future work.

V. Implementation Issues

The 2HARP protocol could be implemented either in dedicated hardware or software. However, there are potential concerns arising from the implementation of digital signatures in software in mobile devices, due to the amount of computational power needed.

The computational complexity of signature operations can be significantly reduced by using techniques such as elliptic curve signature schemes, for example EC-DSA as standardised in ISO/IEC 14888-3 [11], or the NTRU signature scheme [9]. It has been demonstrated that signing using such schemes can be implemented on very limited devices, e.g. smart cards without a dedicated cryptographic processor. The storage complexity (for public and private keys) can also be reduced by using elliptic curve signature schemes, where keys are quite compact.

Transmission complexity can be significantly reduced by using either elliptic curve schemes (for which signatures are inherently short) or any signature scheme giving (partial) message recovery, e.g. one of the schemes standardised in ISO/IEC 9796-2 or 9796-3 [12], [13], as used in the EMV industry standard for integrated circuit cards [1]. In the latter case, adding a signature to a message can add as little as 25 bytes to the message length. Similar techniques, again employed in EMV, can be used to make public key certificates only 40-50 bytes longer than the public key.

VI. Further Work

More research is needed to to see if the threats identified in section IV can be addressed in the network layer. If not, other mechanisms may be needed, where one possible solution could lie with mobile agents.

2HARP is currently being tested via simulation work to assess its performance, and to see whether it achieves its main goals in different scenarios. These include wireless sensor networks where the emphasis is on organisation of sensed information, and in energy constrained routing where metrics based on available energy are more important than other attributes such as shortest path. Simulations will also be used to test how the performance of the upper layers is affected, e.g. TCP connection-oriented communication, especially if the ad hoc network is going to operate with other IP-based networks. Finally, the possibility of adapting the mechanisms of 2HARP to operate with other protocols, such as the Intrazone Routing Protocol (IARP), part of the hybrid Zone Routing Protocol (ZRP) [7], will be considered. Future research will look at the feasibility of turning 2HARP from a protocol reactive to threats to a protocol which can prevent selfish behaviour.

In parallel with this, work will be undertaken on providing a security architecture for 2HARP and other security mechanisms in mobile ad hoc networks. Key management and distribution is a vast topic, which is critical to the successful operation of 2HARP and many other proposed security mechanisms. Arkko et al. [2] have proposed the use of Address Based Keys (ABK) in IPv6 Neighbour Discovery, where the node address is used to generate the public key. This may be an interesting starting point for an infrastructure which will be difficult to manage due to the lack of an online central entity. Other techniques will be investigated to improve the efficiency of 2HARP, including the use of location based services and the broadcasting of a mobility warning message when a node moves.

REFERENCES

[1] *EMV 2000: Integrated Circuit Card Specification for Payment Systems Book 2 - Security and Key Management*, 2000.

[2] J. Arkko, T. Aura, J. Kempf, V. Mäntylä, P. Nikander, and M. Roe. Securing IPv6 neighbor and router discovery. In D. Maughan and N. Vaidya, editors, *Proceedings of the ACM Workshop on Wireless Security, September 28, 2002, Atlanta, Georgia, USA*, pages 77–86. ACM Press, 2002.

[3] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In D. Maughan and N. Vaidya, editors, *Proceedings of the ACM Workshop on Wireless Security, September 28, 2002, Atlanta, Georgia, USA*, pages 21–30. ACM Press, 2002.

[4] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the CONFIDANT protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). In J. Hubaux, J. J. Garcia-Luna-Aceves, and D. Johnson, editors, *Proceedings of The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, 9-11 June, 2002, Lausanne, Switzerland*, pages 226–236. ACM Press, 2002.

[5] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann. The optimized link state routing protocol, evaluation through experiments and simulation. In *Proceedings 4th International Symposium on Wireless Personal Multimedia Communications, September 9-12, 2001, Aalborg, Denmark*, pages 841–846. IEEE Press, 2001.

[6] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2+); Security related network functions (GSM 03.20 version 8.1.0 Release 1999)*.

[7] Z. Haas and M. Pearlman. ZRP — A hybrid framework for routing in ad hoc networks. In C. Perkins, editor, *Ad Hoc Networking*, chapter 7, pages 221–253. Addison-Wesley, 2001.

[8] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1246, 1968.

[9] J. Hoffstein, J. Pipher, and J. Silverman. NSS: An NTRU lattice-based signature scheme. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, May 6-10, 2001, Innsbruck, Austria*, volume 2045 of *Lecture Notes in Computer Science*, pages 211–228. Springer, 2001.

[10] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, April 1-3, 2003, San Fransisco, CA, USA*. IEEE Press, 2003, to appear.

[11] International Organisation for Standardization. *ISO/IEC 14888-3: Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms*, 1998.

[12] International Organisation for Standardization. *ISO/IEC 9796-2: Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms*, 2nd edition, 2002.

[13] International Organisation for Standardization. *ISO/IEC 9796-3: Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms*, 2000.

[14] International Organization for Standardization. *ISO/IEC 7498-1: Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, 2nd edition, 1994.

[15] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In R. Pickholtz, S. Das, R. Caceres, and J. J. Garcia-Luna-Aceves, editors, *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, August 6-11, 2000, Boston, Massachusetts, USA*, pages 255–265. ACM Press, 2000.

[16] P. Michiardi and R. Molva. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In B. Jerman-Blazic and T. Klobucar, editors, *Communications and Multimedia Security, IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, September 26-27, 2002, Portoroz, Slovenia*, volume 228 of *IFIP Conference Proceedings*, pages 107–121. Kluwer Academic, 2002.

[17] C. Perkins and E. Royer. The ad hoc on-demand distance-vector protocol. In C. Perkins, editor, *Ad Hoc Networking*, chapter 6, pages 173–219. Addison-Wesley, 2001.

[18] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In B. Christianson, B. Crispo, and M. Roe, editors, *Security Protocols, 7th International Workshop, April 19-21, 1999, Cambridge, UK*, volume 1796 of *Lecture Notes in Computer Science*, pages 172–194. Springer, 2000.

[19] P. Yau and C. J. Mitchell. Security vulnerabilities in ad hoc networks. In *The Seventh International Symposium on Communication Theory and Applications, July 13–18, 2003, Ambleside, Lake District, UK*, pages 99–104. HW Communications Ltd., July 2003.

[20] P. Yau and V. Sdralia. Towards the security of routing in ad hoc networks. In C. J. Mitchell, editor, *Security for Mobility*, chapter 10, pages 231–268. IEE Press, 2004.