Adaptation and Performance of Covert Channels in Dynamic Source Routing

Michael Marone
Computer Science Department
Yale University
michael.marone@yale.edu

Abstract—Advancements in ad-hoc wireless network routing protocols have raced to keep up with the demands of a less structured, mobile, and dynamic routing environment. Along with the rapid development of these protocols, new forms of attacks and exploitations have begun to raise security issues in data transmission. Current ad-hoc mobile network protocols allow for the presence of covert channels, which create secret information transfer across an otherwise secure network policy. In this paper, we present several methods of data encoding and undetectable communication transfer between nodes in the Dynamic Source Routing (DSR) protocol. The DSR protocol uses source routing rather than the hop-by-hop routing used by the majority of other protocols, which eliminates the need for frequent route advertisement and neighbor detection packets. Unlike covert channels seen in other protocols such as the Ad-hoc On-demand Distance Vector (AODV) protocol, due to promiscuous listening and salvaging by intermediate node route caches, a destination node in a DSR network does not always receive routing packets sent from an originating sender node. We therefore present several methods of covert communication which guarantee covert information delivery. We also explore methods of detection and prevention from this manipulation of network resources, and provide a frame of comparison for the protocol's capacity for covert channels versus other network routing protocols. The comparison demonstrates that covert channels in DSR are sometimes impossible to detect; however, measures such as aggressive cache cleanup, neighborhood watches, and digital signatures on forwarded request packets can provide a backdrop of security against malicious users.

I. Introduction

As people become increasingly dependent on cell phones and other wireless communication links, the development of efficient and economic routing protocols has gained greater focus. An important factor in weighing the relative capabilities of these protocols is their ability to withstand attacks and exploitation from malicious and selfish users. Protection against covert security breaches across mobile ad hoc networks is hard to achieve due to the dynamic and decentralized topology of these networks. The difficulty of this task is compounded by the inability of an ad-hoc architecture to implement a predefined security architecture or authentication system for the expansion and shrinking of the network. When ad hoc mobile routing protocols face subtle and cleverly coordinated security threats, the lack of built-in defense mechanisms raise multiple vulnerability concerns.

A. Covert Channels and Information Hiding

Though research in ad-hoc mobile network security protocols have uncovered ways to defend against many complex security hazards, protecting against the surreptitious transfer of data through covert channels has been scarcely considered. A covert channel simply put is the transfer of data between two processes that are not permitted or not known to be in touch with each other. To set up the channel, the sender and receiver communicate covertly over normal ports, exchanging secret messages using the standard procedure without tipping off neighbor nodes or security monitors as to their behavior. The notion of covert communication was first discussed by

Lampson [12] and was later refined through papers by Huskamp [6], Schaefer [21], and Kemmerer [11]. These early sources describe covert channels as processes acting on behalf of another user in an operating system. They are mainly divided into two categories: storage and timing channels. Storage channels allow processes to write covert information to a shared storage medium, while timing channels signal information to another process by altering digits in the CPU system clock. Lampson was the first to point out that nondiscretionary security policies should address such problems of unauthorized information release over covert channels. He and other researchers responded to this threat by providing security models in operating systems that prevented the two stealth processes from communicating with each other.

The notion of covert channels in networks was later described as "leakage channels" between nodes after the development of packet forwarding in internet routing [13]. A comprehensive paper on covert channels [17] describes a covert communication between a sender and receiver as being illegal over a nondiscretionary security policy model and its interpretation. Tsai [22] further notes that the communication is covert "if and only if" the communication of the two processes are strictly prohibited over the security model. To further clarify the notion of a covert channel in network transmission, Millen discusses briefly the distinction between covert channels and information hiding. On the one hand, covert channels take place between two parties that are not supposed to communicate with each other or are not known to be communicating, whereas information hiding is the transfer of encrypted data between two parties that are allowed to send messages between each other. The most basic method of information hiding is described by Katzenbeisser and Petitcolas [9] and by Kurak [10] in the image downgrading problem. In this method of information hiding, the secret data is encoded in the LSB (least significant bits) of regular images. Since it is very difficult to notice the slight pixel differences, the transfer of secret data between two sources occurs undetected. The transfer of this message between parties is clearly allowed by the security protocol and is an extremely difficult method to detect against. In this paper, however, we limit our discussion to covert channels and their modern implementation in an ad hoc network. In order to establish a covert channel, the two parties involved must synchronize the sending and receiving of the encrypted data in a manner that does not disrupt the security protocol.

Though covert channels are difficult to implement and utilize, when correctly executed they directly violate a node's privacy and create integrity vulnerabilities on the network. As society depends more and more on transferring sensitive information such as ATM or bank account information over wireless or digital networks, network security developers must pay more attention to protecting against the malevolent harnessing of covert channels. The subtle communication of malicious users over a covert channel allows for illicit and even unlawful leakages of confidential or personal information. In this paper, we describe a situation in which a user implants a covert process such as a Trojan horse program [23] to locate sensitive data (passwords, personal information, credit card numbers, etc), encode it, and gradually send it back to a destination node on the network over a period of time. We concentrate on discussing the threat of covert channels in the Dynamic Source Routing (DSR) protocol, an increasingly popular method of routing for mobile ad-hoc networks.

B. Description and Relevance of DSR protocol

The Dynamic Source Routing Protocol [7, 16] adopts an on-demand approach of source routing to allow for the efficient forwarding of packets. This adaptable and economical method of data transfer over large areas was created and developed for mobile ad hoc networks (MANET) at Carnegie Mellon University [8]. Each packet sent over DSR carries a header with an ordered source list of the nodes to traverse to reach its destination. It was specifically designed for use in mobile ad hoc wireless networks where nodes are constantly leaving and joining the network. The data structure DSR uses to keep track of the routing information is a route cache, which stores the route replies it receives from route requests. Since packets already contain the routing path, intermediate nodes do not need to maintain a current map of the network infrastructure. This also allows a packet to be loop-free, and eliminates the necessity for the frequent neighbor updates or route advertisement packets that are essential for the operation of other routing protocols. The two most prominent features of DSR are route discovery and route maintenance. Route discovery is initiated through route request packets from a sender to a destination. The request packets are broadcast in a controlled way to all nodes in the network and are answered by either the destination node or an intercepting node that knows a route to the destination [17]. This on-demand feature allows DSR to be completely self-organizing and self-configuring, making changes only to the routes requested by sending nodes. It has proven useful in eliminating the flood of update messages due to the frequent addition or departing of nodes as in mobile wireless networks.

DSR has developed into a prominent protocol in large scale ad-hoc mobile routing (along with AODV—Ad-hoc On-demand Distance Vector) and has become a focal point for several routing research and development projects. A notable venture in the use of DSR protocol is in an undersea systems network in Australia. In the project, researchers manipulate dynamic source routing to search for a robust and efficient set of routing methods for AUVs (autonomous undersea vehicles) [15]. Another example of the capacity of DSR could be an ad-hoc network wireless cellular system in a developing country. Such environments with no supporting infrastructure, an extremely large number of users and a constantly changing topology could become potential targets for DSR implementation. The DSR protocol guarantees that neighbor nodes can safely transfer packets without excessive storage or forwarding overhead, but it does not provide complete informational security. For the DSR protocol to continue grow in usage over a large scale network, it must assure the complete integrity and security of data transfer.

II. Ad hoc Network Security and Related Work

A. Security and Related Work

Johnson et al. [8] designed the DSR protocol so that individual nodes could efficiently and securely send data through neighbor nodes to a desired location in any given network topology. Due to the infrastructureless and dynamic nature of this mobile ad hoc network, malicious nodes have ample opportunity to attack network security. Several coordinated research efforts have been aimed at achieving greater security in ad-hoc routing protocols. Most notably are Zhou and Haas [24], who propose a key management service with no central authentication server. This takes advantage of the inherent redundancies given by multiple routes to provide each node with a distinct security key. Other methods of securing data links include hash-coding [19, 20] and password key exchanges that allow for source verification [1, 5]. Furthermore, research performed by Marti et al. [14] suggests implementing a neighborhood watchdog system to detect malicious behavior and keep track of malicious to avoid. In depth implementations of the neighborhood watch system have been conducted [5] and demonstrate its efficiency in identifying malicious nodes characterized by their desire to gain advantages through selfish or reckless network behavior. Such behavior could include exploiting bandwidth or disrupting the functionality of the network to gain better traffic routes for their own data packets. This paper narrows in on the concealment and transfer of data across these routing links without the blatancy that would alert neighbor nodes.

Recent work at the University of Maryland [4] focuses on covert transmissions in the Ad-hoc On-demand Distance Vector Routing (AODV) protocol and tests for vulnerabilities. This work shows how the on-demand nature of AODV path discovery allows for manipulation of the routing control packets to convey covert transmission. It also shows several ways for communication between a sender and receiver, and focuses more specifically on the request RREQ broadcasts by a sender. The covert information is decoded into the destination ID of the route requests, and is sent one bit at a time to a knowing receiver. The order of reception is assured through the sequence number in the RREQs. Though technically considered information hiding because of the unconcealed nature of the channel between the sender and the receiver, the simulation shows that in a lossy, noisy environment with limited bandwidth, covert information transfer is possible. The affirmation of the possibility of covert data transfer in the results of this simulation and the sheer difficulty of detection have been the driving force behind the exploration of covert information transfer in the DSR protocol.

B. Contribution

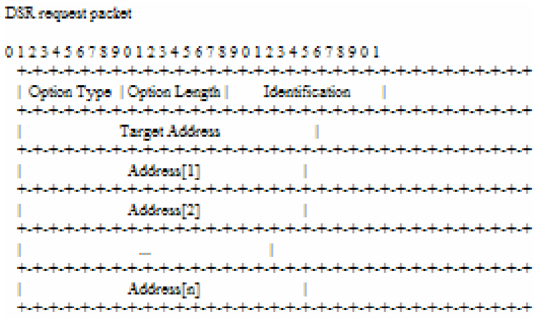
The results of the AODV experiment demonstrated one method of covert information in a routing protocol and showed that this data transfer may sometimes be impossible to detect or prevent. In this paper, we explore methods of data encoding and covert transfer in the Dynamic Source Routing protocol and offer a comprehensive response to prevention of these methods of transfer. To demonstrate covert data transfer, we implement two situations of data transfer over covert channels and show how difficult they are to difficult to detect. The first deals with two nodes in collusion that seek to transfer information through other nodes without them knowing. The second method creates a situation where the node itself does not know it is in a covert channel transfer with another node. We then explore both successful and ineffective methods of prevention that could be used to combat certain security issues. The concepts discussed in this paper could be used to increase awareness of the threats of covert channels and strengthen the base of protection in the future for routing protocols and topology management.

III. Covert Channels in the DSR Protocol

A. Encoding data in DSR routing

We will first examine the types and structure of message transfer in the DSR protocol, then look at several methods of encoding encrypted data, and finally consider some methods of opening covert channels to transfer this data.

There are three main data packets originated by the DSR protocol: requests (RREQ), replies (RREP) and error messages (ERR). The main method of covert data transfer we focus on in this paper is the request packet. As illustrated in the diagram below, the request packet stores three main variables: the option type/length, the identification id, and the list of paths to the target address.



```
DSR request packet

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Option Type  | Option Length |       Identification       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Target Address                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Address[1]                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Address[2]                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      ...                  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Address[n]                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The request packet also includes room for other variables such as hop counts, time-to-live counts, and times stamps. All of these variables can be utilized for covert data transfer.

The overall format used for the DSR protocol allows for a flood of methods to encrypt data through the seemingly normal route requests and replies during the discovery process. For instance, during route discovery, the most basic method of covert encryption of the compromised data is in the route **request identification number**. Each route request message identifies the initiator and target of the Route Discovery, and also contains a unique request identification number (sometimes two), determined by the initiator of the Request. This code number modulo an agreed upon key K could be a code word used to indicate encrypted data.

In addition, the number of **hop limits** could be altered or informative numbers can be added to the hop limit to denote encrypted data. A more traditional way of sending covert data is through altering the clock times of the data. For instance, a message might be sent with a **clock time** several milliseconds off what it should be, and the difference is significant for the data transfer.

A more dynamic method of transfer is by sending packets in the wrong order, and having a **reordering algorithm** that takes this into effect. To illustrate, if the packets p(1,2,3) are sent in the wrong order—such as p(2,1,3)—it is clear that there are 2^8=256 different combinations of bits that are sent through this reordering. With prior agreement on an algorithm scheme, this could be a practical and mischievous way of sending encrypted data.

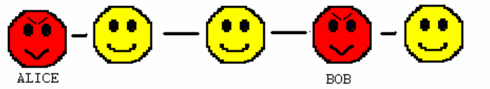
Another possible scheme is by encoding data into the routing location by putting **trivial destination addresses** into the packet address route. For instance, the address for a packet from sender A to receiver B could be something like: A, C, B, N, O, W. In this case, the receiver B would receive the message after the packet is routed to C, and discovers the extra nodes "N, O, W" as the secret message. B could even intercept the message, and reply with either an error or reply message.

Finally, all packets include the option to **piggyback data** onto the message through the options header—encoded data could be piggybacked onto a message and stripped off by the receiver message without any other nodes noticing. Also, error messages in routing requests are piggybacked, they can be altered in a way that sends back covert data.

B. Opening a covert channel and transferring the data

When transferring data between two knowledgeable agents (between Alice and Bob) in a covert channel DSR allows for several ways of exchanging information between two covert agents. For instance, if Alice wanted to tell Bob to "Attack at Midnight" without the other nodes to know about it, she might use these methods for her covert transfer:

Case 1: Alice knowingly opens channel with Bob.



1) Alice sends a request for *a phantom node*, which is broadcast throughout the network and allows Bob to intercept. Segments from the encrypted data could be sent through this initial route request packet to Bob. Bob could then either send an error message back to Alice, which would be the most reasonable, and Alice could keep sending requests for phantom nodes to transmit encoded data through a packet identification number or option type number to Bob. Bob could even send a route reply message to Alice, encrypting data in the reply to inform Alice of his response to a question, for instance.

2) Alice could continuously *originate route error messages* with the encoded data. Error messages are broadcast to nodes in a network and do not have to be instigated. Data could be encrypted by a method described above.

3) Alice sends a route request message or data packet to Bob, but *Bob does not respond*. The node connected to Bob will then sense that Bob is disconnected and send an error message to Alice. Alice can then initiate route discovery and send request messages to Bob, which Bob in turn ignores, and the process continues.

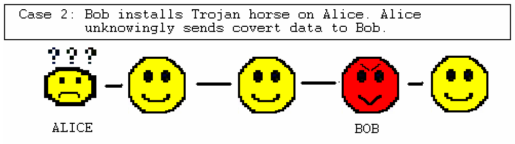
4) An infinite cycle of *gratuitous route discovery messages* could be sent back and forth between two nodes Bob and Alice in which Alice sends a route request for Bob, and when Bob receives the request, he initiates route discovery for Alice, and piggybacks his response onto this message. The communication can continue back and forth.

5) An alternative method presupposes that Alice and Bob have a *prearranged cache list* and use certain routes to indicate codes For instance, A and B have three paths to node E:

1. A, B, C, E
2. A, D, B, C, E
3. A, B, C, F, E

Note all paths include A and B. If A sends a message to E using a combination of these routes, then a secret code can be formed (i.e. 212 213 21…) which indicates the code word. A number of methods can keep track of the sequencing of the code, in case a packet is lost or dropped, and Alice and Bob can send messages in any of the encoding techniques to agree on the different numbering for each path in case a node in one of the paths leaves during transmission.

A more vulnerable security and privacy threat occurs if one party (Bob) installs a process on another node (Alice) such as a Trojan horse that causes unacknowledged transfer of data between the violated computer (Alice) and the violating computer (Bob). Such a covert channel could steal information from Alice's computer such as credit card information and leak it to Bob over a period of time without Alice or anyone else knowing about the information leak.



Case 2: Bob installs Trojan horse on Alice. Alice unknowingly sends covert data to Bob.

1) When Alice wants to send a message to a node C in the network, the Trojan horse program on her computer could *change the forwarding information to pass through Bob* on purpose even if it was not the most efficient path. Alice would thus be unaware any encoded data passing through Bob on its way to C.

2) The stealth Trojan horse program on Alice could *add routes for phantom nodes into Alice's cache*. Bob could then query for these phantom nodes, and Alice could respond to the request and forward the packets to the phantom nodes in its cache. These messages in turn could carry encoded data and be routed back to Bob. For instance, if the Trojan program writes a route path to the phantom destination node S into Alice's cache list, and Bob searches for S, he will always come into contact with Alice, since no other node has node S in their list. This technique can be maneuvered by Bob so that Alice sends information on the phantom node to Bob without her ever knowing of the information transfer.

3) A combination of the methods described in 4 or 5, where Bob initiates *continuous route discoveries* into Alice's cache and the Trojan horse program sends route reply paths based on a *particular order* that converts to the transmitted data.. The relative success of these transmissions depends on the security protocol implemented by Alice.

4) When routing requests send from Bob pass through Alice, her Trojan horse program *alters the route paths* on request packets to denote the covert data stolen by the program.

IV. Protection and Detection

Some of the above mentioned events may be difficult to monitor and detect, which adds difficulty to the process of providing security to the network. It is important to develop security methods which prevent against these forms of data attack so that the confidentiality and privacy of member nodes is not compromised. The goal of protection is for member nodes to gain awareness of the status of fellow member nodes and to be protected if it is found out that a member node is malicious.

A. Aggressive caching

The current specifications of DSR lack a mechanism to determine the freshness of routes along a route path [16]. In addition, stale routes have no way of being purged from the cache memory. More importantly, an unprotected and un-updated cache allows room for malicious nodes to manipulate cache for covert data transfer as noted in the cases above. Stale cache links and incomplete error notification also contribute to greater inefficiency and harmful affects on DSR [15]. DSR already uses aggressive caching, but several modifications of the standard protocol if implemented can contribute to greater security. For instance, the physical breakage of a link would go undetected if a particular node were not to attempt to use this link. A timestamp method of pruning would prohibit this situation for the manipulation of cache data by a malicious user from occurring. In addition, downed links known from overhearing transmitted packets should be aggressively pruned from the cache. These methods provide for a greater house-cleaning of the cache and serve to protect a node as well as to make it more efficient.

B. Neighborhood Watch

In a networking environment, the most effective way to ensure security is to monitor and enforce the respect of security policy. In ad hoc mobile networks it is not feasible to have a central authority that takes main control in enforcing protocol, as it is not efficient nor reasonable, so the detection of protocol deviances must be handled by the neighbors of a node [14, 3]. Nodes may learn from observed behavior of their neighbors to watch for misbehavior such as the dropping of packets, unusual traffic attention or unnecessary route salvaging. This monitoring system could also learn from a neighbor's behavior, and report its experiences with other neighbors. Should other neighbors pick up on the mischievousness also, nodes would all know not to use the aforementioned node. This method allows for greater of dynamic adjustments and autonomous learning methods for the network as a whole, and it prevents against nodes flooding the network with endless streams of trivial data.

C. Digital Signatures

Another addition to prevent covert channels in the DSR protocol is the inclusion of a layer of asymmetric cryptography. Each node could be provided with a private key relative to its IP address [24] and a public key open to all nodes on the network to provide the tools necessary for a digital signature. Each time a node on the network forwards a packet, it stamps it with its signature. Each intermediate node therefore would be able to verify each of the signatures on the RREQ message they received within a certain probability [5]. This would protect against nodes altering the route paths when they forward messages. Though this provides necessary security to DSR, it may not be the most appropriate. The main draw is that authenticating a new node is difficult when malicious nodes in a network can learn information about the identity of the new through listening to communication with it. This requires the addition of a certificate authority, which may not be feasible when the network is constantly changing. However, if each node signed forwarded messages during the route request, and checked their signature on the route reply, they may each determine whether their particular forwarding route was altered.

V. Future Work and Conclusion

A. Future Work

The next steps to carry out will consist of implementing more approaches discussed so far and putting them into the *ns-2* simulator for testing. The focus for these tests will be similar to the simulations performed at The University of Maryland—the success rate of covert channels on the protocol. Whereas the results in the AODV experiment proved that the qualities of channels were poor because of the frequent dropping of packets, the DSR protocol allows for more assurance of packet delivery. This allows for greater success of packet transfer over network broadcasts. Analysis of the scalability, throughput ratios, and the relative ability of the covert channels to go undetected should be undertaken.

Furthermore, the protection methods against covert channels should be tested to evaluate the decrease in covert throughput, cost/benefit ratios and overhead. The overall cooperativeness of the network can also be tested by involving neighborhood nodes into the inference mechanism.

B. Conclusion

Due to the rise in popularity and growing cheapness of wireless and sensor networks, efficient and secure network routing and broadcasting will attract more research projects involving wireless communication. In order to meet the demand and ensure safeness, the foundations for secure broadcast and network maintenance must be solidified to pave the way for future wireless network expansion in cell phones, personal computers, and digital entertainment. The above mention attempts to bypass network security highlight the need for aggressive security measures in future protocols. Because DSR allows for compromised communication for these networks, further research in secure mobile network communication and broadcast will be necessary.

VI. References

[1] N. Asokan, P. Ginzboorg. Key Agreement in Ad-hoc Networks. Computer Communications, 23:1627-1637, 2000.

[2] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. Proceedings of MobiCom'98, Dallas, TX, October 1998.

[3] Sonja Buchegger, Jean-Yves Le Boudec. Nodes Bearing Grudges : Towards Routing Security, Fairness, and Robustness in Mobile Ad hoc Networks. In *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, 2002.

[4] Anthony Ephremides. Some Trade-Offs In Sensor Networks. 2003.

[5] Shayan Ghazizadeh, Okhtay Ilghami, Evren Sirin, Fusun Yaman. Security-Aware Adaptive Dynamic Source Routing Protocol. In *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks(LCN'02)*, 2002.

[6] C. Huskamp, Covert Communication Channels in Timesharing Systems, *Technical Report UCB-CS-78-02*, Ph.D. Thesis, University of California, Berkeley, California, 1978.

[7] David B. Johnson, David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks , *Mobile Computing*, 1996.

[8] D.B. Johnson and D. A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, Mobile Ad Hoc Network (MANET) Working Group, IETF, October 1999.

[9] Stefan Katzenbeisser, Fabien A. P. Petitcolas. Information hiding techniques for steganography and digital watermarking. Artech House Books, 1999

[10] C. Kurak and J. McHugh. A cautionary note on image downgrading. In *Computer Security Applications Conference.*

[11] R. A. Kemmerer. Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels. *ACM Transactions on Computer Systems*, 1:3, pp. 256-277, August 1983.

[12] B. W. Lampson. A Note on the Confinement Problem. *Communications of the ACM*, 16:10, pp. 613-615, October 1973

[13] Jonathan Millen. 20 Years of Covert Channel Modeling and Analysis. 1998.

[14] S. Marti, T. Giuli, K. Lai, and m. Baker. Mitigating routing misbehavior in Mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pg 255-265, 2000.

[15] D. Maltz, J. Borch, J. Jetcheva, and D. Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks. *IEEE Journal on Selected*

Areas of Communication, 17 (8), August 1999.

[16] M. K. Marina and S. R. Das. Performance of Route Caching Strategies in Dynamic Source Routing. In *Proceedings of the 2nd Wireless Networking and Mobile Computing (WNMC)*, Phoenix, April 2001.

[17] National Computer Security Center. A Guide to Understanding Covert Channel Analysis of Trusted Systems. NCSC-TG-030, November 1993.

[18] Robert Nitzel and Charles Benton. Exploiting Dynamic Source routing to Enable Undersea Networking over an Ad-hoc topology. AUSNET 2002.

[19] Krishna Paul, Dick Westhoff. Context Aware Detection of Selfish Nodes in DSR based Ad-hoc Networks. IEEE 2002.

[20] P. Papadimitrados, Z. Haas. Securing Routing for Mobile Ad hoc Networks. *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference* (CNDS 2002), 2002.

[21] M. Schaefer, B. Gold, R. Linde, and J. Scheid. Program Confinement in KVM/370. *Proceedings of the 1977 Annual ACM Conference*. Seattle, Washington, ACM, New York, pp. 404-410, October 1977.

[22] R. Tsai, V. D. Gligor, and C. S Chandersekaran. A Formal Method for the Identification of Covert Storage Channels in Source Code. *lEEE Transactions on Software Engineering*, 16:6, pp. 569-580, June 1990.

[23] John P. Wack, Lisa J. Carnahan. Computer Viruses and Related Threats: A Management Guide. 1989

[24] L. Zhou, Z.J. Haas, Securing Ad Hoc Networks, in *IEEE Network Magazine*, vol 3, Nov/Dec 1999.