Mitigating Byzantine Attacks in Ad Hoc Wireless Networks

Technical Report Version 1
March 2004

Baruch Awerbuch [*] Reza Curtmola [*] David Holmer [*] Cristina Nita-Rotaru [†]
Herbert Rubens [*]

Abstract

Attacks where adversaries have full control of a number of authenticated devices and behave arbitrarily to disrupt the network are referred to as Byzantine attacks. Traditional secure routing protocols are vulnerable to this class of attacks since they usually assume that once authenticated, a node can be trusted to execute the protocol correctly. We present a detailed description of several Byzantine attacks (black hole, flood rushing, wormhole and overlay network wormhole), analyze their mechanisms and describe the major mitigation techniques. Through simulation, we perform a quantitative evaluation of the impact of these attacks on an insecure on-demand routing protocol. The relative strength of the attacks is analyzed in terms of the magnitude of disruption caused per adversary. An implementation of the On-Demand Secure Byzantine Routing protocol (ODSBR) was created in order to quantify its ability to mitigate the considered attacks. ODSBR was chosen because its design addresses a wide range of Byzantine attacks.

1 Introduction

The wide-spread adoption of portable computing devices combined with the recent advances in wireless technology has lead to increases in productivity in the corporate and industrial sectors. While these recent advances have enhanced existing business processes, they have also introduced new security vulnerabilities.

In the past, networks have strongly relied on physical security. The concept of a network firewall is a perfect example in this direction. A network firewall is intended to provide an access control division between the insecure public network (the Internet) and the seemingly secure private internal corporate network.

However, the rapid adoption of wireless networking technology, makes the assumption about the physical security of the network infrastructure unrealistic. This is because the wireless shared medium is completely exposed to outsiders and susceptible to attacks that could potentially target any of the OSI/ISO layers in the network stack. Examples of such attacks include jamming of the physical layer, disruption of the medium access control layer coordination packets, attacks against the routing infrastructure, targeted attacks on the transport protocols (such as an attack against packets addressed to a specific port), or even attacks intended to disrupt specific applications.

In addition to the vulnerabilities of the wireless communication infrastructure, the ultra portability of modern devices provides an increased susceptibility to theft. Over the past year, 59% of companies surveyed in the CSI/FBI Computer Crime and Security Survey [1] reported that laptops had been stolen. The cost of these stolen devices is minimal in comparison to the information they contain and the resources they provide access to. If an attacker was able to gain access to the corporate network of a financial services company and disrupt the trading floor network, the monetary consequences could be catastrophic.

The military has served as both the initial investigator and the earliest adopter of wireless ad hoc networking technologies [2]. The security of military networks is critical since a disruption could lead to the loss of life. The likelihood of authenticated devices being captured by the enemy in a chaotic battlefield environment is extremely high.

In this work we consider the case where a device or a set of devices could be compromised and be under the control of an adversary or set of adversaries that can collude. Once an adversary has control of an authenticated device, protocols which rely on authentication to provide security ser-

[*] Department of Computer Science, Johns Hopkins University, 3400 N. Charles St. Baltimore, MD 21218 USA. E-mail: {baruch, crix, dholmer, herb}@cs.jhu.edu .

[†] Department of Computer Science, Purdue University, 250 N. University Street, West Lafayette, IN 47907. E-mail: crisn@cs.purdue.edu .

vices become of little use. Authentication and data integrity mechanisms, although needed in order to prevent injection, fabrication and impersonation attacks, do not provide protection against insider attacks since they cannot force a node to behave according to the protocol. We call such attacks, where the adversary has full control of an authenticated device and can perform arbitrary behavior to disrupt the system, Byzantine attacks. From a more general perspective, a Byzantine attack is any attack that involves the leaking of authentication secrets so that an adversarial device is indistinguishable from a legitimate one. This model requires the use of protocols that are designed to withstand disruptions caused by authenticated nodes in addition to the more traditional protection against external attacks.

1.1 Byzantine Attacks

Many vulnerabilities in network protocols (including wireless ad hoc routing protocols) are caused by the lack of message integrity and authentication mechanisms, which allows an attacker to alter or fabricate packets. Significant research in securing ad hoc wireless routing protocols [3, 4, 5, 6] and wired routing protocols [7, 8, 9] focused on this aspect. Authentication and integrity are required to protect a network protocol, since they ensure that a packet was generated by an authenticated node and has not been tampered with. However, they do not provide any guarantee about the legitimacy of actions taken by authenticated nodes.

Attacks where the adversary has full control of an authenticated device and can perform arbitrary behavior to disrupt the system are referred to as Byzantine[1] attacks. Research addressing this category of attacks is quite scarce. Below, we outline several Byzantine attacks that are considered in this work. All of them can be mounted against ad hoc wireless routing protocols.

Although many Byzantine attacks share certain features with the "selfish" node problem [11] (e.g. not forwarding the data packets of others), the intentions of nodes under these two models are different. The goal of a selfish node is to reap the benefits of participating in the ad hoc network without having to expend its own resources in exchange. In contrast, the goal of a Byzantine node is to disrupt the communication of other nodes in the network, without regard to its own resource consumption.

Black Hole Attack A basic Byzantine attack is a black hole attack where the adversary stops forwarding data packets, but still participates in the routing protocol correctly. As a result, whenever the adversarial node is selected as part of a path by the routing protocol, it prevents communication on that path from taking place. Most existing secure and insecure routing protocols are disrupted by black hole attacks because they render the normal methods of route maintenance useless.

Flood Rushing Attack A flood rushing attack exploits the flood duplicate suppression technique used by many routing protocols. This attack takes place during the propagation of a legitimate flood and can be seen as a "race" between the legitimate flood and the adversarial variant of it. If an adversary successfully reaches some of its neighbors with its own version of the flood packet before they receive a version through a legitimate route, then those nodes will ignore the legitimate version and will propagate the adversarial version. This may result in the continual inability to establish an adversarial-free route, even when authentication techniques are used.

Byzantine Wormhole Attack If more than one node is compromised, it is reasonable to assume that these nodes may interact in order to gain an additional advantage. This allows the adversary to perform a more effective attack. Indeed, one such attack is a Byzantine wormhole, where two adversaries collude by tunnelling packets between each other in order to create a shortcut (or wormhole) in the network. This tunnel can be created either using a private communication channel, such as a pair of radios and directional antennas, or by using the existing ad hoc network infrastructure. The adversaries can send a route request and discover a route across the ad hoc network, then tunnel packets through the non-adversarial nodes to execute the attack. The adversaries can use the low cost appearance of the wormhole links in order to increase the probability of being selected as part of the route, and then attempt to disrupt the network by dropping all of the data packets. The Byzantine wormhole attack is an extremely strong attack that can be performed even if only two nodes have been compromised.

Byzantine Overlay Network Wormhole Attack A more general variant of the previous attack occurs when several nodes are compromised and form an overlay network. By tunnelling packets through the overlay network, the adversaries make it appear to the routing protocol that they are all neighbors, which considerably increases their chances of being selected on routes. This is the strongest attack considered in this work.

1.2 Our contributions

Many of the above attacks were studied individually in prior work, but under a weaker adversarial model. [12] studied a black hole attack, under a model where attackers cannot collude and the only malicious action is refus-

[1] The Byzantine term was introduced in [10] which addressed the problem of trying to reach agreement between Byzantine generals in the presence of traitors. More generally, the term is now used to denote participants whose actions cannot be trusted, or whose action do not conform with protocol specifications.

ing to forward data packets. The wormhole [13] and flood rushing [14] attacks were discussed together with some potential solutions. However, to our knowledge, there is no work attempting to quantify the damage caused by a large class of Byzantine attacks, particularly when combinations of attacks are considered.

In this work we also evaluate the effectiveness of the On-Demand Secure Byzantine Routing (ODSBR) protocol [15], which was specifically designed to mitigate a wide range of Byzantine attacks in ad hoc wireless networks. More precisely:

- We present a detailed description of several Byzantine attacks (black hole, flood rushing, wormhole and overlay network wormhole), analyze their mechanisms and describe mitigation techniques.
- We developed a protocol independent Byzantine attack module for NS2 in order to simulate these attacks. This module is a helpful tool for the secure routing community.
- We demonstrate the effects of the considered attacks on the NS2 implementation of the AODV[16] routing protocol. Our results quantify the damage caused by various Byzantine attacks.
- We implemented the ODSBR protocol in order to quantify its ability to mitigate the considered Byzantine attacks.
- We use the simulation results to compare the attacks and identify those which result in the greatest network disruption while requiring the least number of adversarial participants.

The rest of the paper is organized as follows. We present an overview of the ODSBR protocol in Section 2, and analyze different types of Byzantine attacks as well as demonstrate their impact on AODV and how ODSBR mitigates the damage in Section 3. Section 4 presents an analysis of the simulation results. Section 5 overviews related work. We conclude and suggest future work directions in Section 6.

2 ODSBR

The ODSBR protocol was introduced in [15], but was never implemented prior to this work. Below we present a brief overview of the protocol (Section 2.1), which may be skipped by readers already familiar with the original work. We also discuss implementation details, as well as changes to the original protocol motivated by practical considerations (Section 2.2).

2.1 Overview

The ODSBR protocol is an on-demand routing protocol for wireless ad hoc networks that detects Byzantine behavior and avoids it. The protocol is designed to locate a fault free path in an ad hoc network (if such a path exists), even when a majority of the nodes in the network have been compromised and are exhibiting Byzantine behavior, alone or colluding. The protocol assumes that only the source and the destination are trusted. Nodes that cannot be authenticated do not participate in the protocol, and are not trusted. Intermediate nodes on the path between the source and the destination can be authenticated and can participate in the protocol, but may exhibit Byzantine behavior.

A fault is defined as any disruption that causes significant loss or delay in the network. It can be caused by Byzantine behavior, external adversaries, lower layer influences, and certain types of normal network behavior such as bursting traffic. An adversary or group of adversaries can intercept, modify, or fabricate packets, create routing loops, drop packets selectively, artificially delay packets, route packets along non-optimal paths, or make a path look either longer or shorter than it is. All the above attacks result in disruption or degradation of the routing service. In addition, they can induce excess resource consumption which is particularly problematic in wireless networks.

The ODSBR protocol establishes a reliability metric based on past history and uses it to select the best path. The metric is represented by a list of link weights where high weights correspond to low reliability. Each node in the network maintains its own list, referred to as a weight list, and dynamically updates that list when it detects faults. Faulty links are identified using a secure adaptive probing technique that is embedded in the regular packet stream, to protect it from adversary detection. These links are then avoided using a secure route discovery protocol that incorporates the reliability metric. More specifically, the protocol can be separated into three successive phases, each phase using as input the output from the previous:

- *Route discovery with fault avoidance.* Using flooding, cryptographic primitives and, as input, a list with the weights of faulty links, this phase outputs the full least weight path from the source to the destination.
- *Byzantine fault detection.* The goal of this phase is to discover faulty links on the path from the source to the destination. This phase takes as input the full path and outputs a faulty link, using an adaptive probing technique. Cryptographic primitives and sequence numbers are used to protect the detection mechanism from adversaries.
- *Link weight management.* This phase maintains a weight list for links discovered by the fault detection algorithm. A multiplicative increase scheme is used to

penalize links which are then rehabilitated as packets are successfully delivered. The weight list is used by the route discovery phase to avoid faulty paths.

2.2 Implementation Details

This section describes key details of the protocol implementation. We present changes to the original protocol motivated by practical considerations and discuss other enhancements that improved the performance of the protocol. To be able to explore the performance of ODSBR under a variety of network environments and attack scenarios, we implemented the protocol using the NS2 [17] network simulator (version 2.27). We assumed the protocol uses RSA [18] with 1024-bit keys for public key operations (128 bytes), AES with 128-bit keys for symmetric encryptions and HMAC [19] with SHA1 as the message authentication code (20 bytes). The actual cryptographic operations performed by the protocol are not executed in the simulation, as this would drastically increase the simulation runtime, reducing its efficiency as an experimental tool. Instead, the impact of these cryptographic operations is represented by adjusting the simulated packet sizes and by introducing packet delay accordingly, as if the packet actually contained authenticating data (e.g. digital signatures or MACs), and as if CPU time was spent performing cryptographic operations[2]. Also, meta data that represents the integrity of any cryptographic content is associated with each packet. This meta data allows us to simulate the effect of adversaries that “tamper” with packets.

In the original protocol, the fault detection phase detects a faulty link by inserting probes gradually, according to a binary search algorithm. For practical reasons we decided to simplify this scheme by having only two states. In the “non-probing” state only the destination returns ACKs, while in the “probing” state all intermediate nodes also return ACKs. The protocol operates in the non-probing state until a loss threshold violation occurs and a fault is detected. Where the original algorithm would divide the path in two at this point (thus creating two intervals), the implemented version switches to the probing state, effectively probing all nodes along the path (in which case there is an interval for every link). While the original strategy avoids having to exchange keys with all intermediate nodes on the path, it may take several faults before an individual link is identified. When the total number of hops is relatively small, the cost of enabling all the probes at once is low, and the two-state technique both reduces the amount of time necessary to identify a link (down to exactly two faults), and considerably simplifies the protocol implementation. If, in probing state, the source node successfully delivers enough packets and the loss rate goes below a specified threshold, then the source node returns to the non-probing state.

The performance of the implementation is influenced by the values of several parameters: the loss threshold rate, the timeout allowed for a packet to traverse a link and the size of the sliding window necessary to keep track of the packet loss history. After conducting a series of experiments with different sets of parameters, the values in Table 1 were chosen. We tuned these parameters conservatively in order to ensure that the protocol will operate in a wide range of environments. Although the simulations in this work were conducted with 50 nodes, these values were tuned for efficient operation with up to 100.

Parameter	Value
loss threshold rate	10%
link timeout	250 milliseconds
sliding window size	100 packets

Table 1. ODSBR implementation parameters

3 Analysis and Experimental Results

In this section we consider several Byzantine attacks that can be performed by an adversary or group of colluding adversaries. We describe the attack mechanisms, focusing on the ratio between the amount of effort needed to perform an attack and the disruption caused by the attack. Intuitively, it is the simple yet strong attacks that are most likely to occur, and these are the most important to be mitigated. We then discuss approaches that can mitigate these attacks. We simulate a number of different attacks against the insecure AODV routing protocol, showing the impact these attacks can have.

We conduct additional simulations in order to investigate the effectiveness of ODSBR in mitigating these attacks. Although a number of secure ad hoc routing protocols exist which provide authentication to AODV or to similar on-demand protocols, we did not simulate them because these protocols cannot protect against attacks coming from adversarial nodes that behave in an arbitrary manner. Under the set of Byzantine attacks simulated in this paper, authentication-based secure routing protocols, such as [6], [4], [3], [5], do not provide additional resilience over the insecure AODV protocol.

[2] We have adjusted the time delays to approximate the performance of a 1.5 GHz Intel Pentium M processor. Further exploration of protocol performance on CPU-constrained devices, such as PDAs, should be evaluated in future work.

3.1 Simulation Setup

Simulations were conducted using the NS2[17] network simulator. Nodes in the network were configured to use 802.11 radios with a bandwidth of 2 Mbps and a nominal range or 250 m. All the simulated routing protocols were configured with their default parameters. The simulations were conducted by randomly placing 50 nodes within a 1000 by 1000 meter square area. In addition to these 50 nodes, 0 to 10 adversarial nodes were added to the simulations, depending on the considered attack configuration. A traffic load of 10 constant bit rate (CBR) flows was used to simulate data communication through the ad hoc network. An aggregate load of 0.1 Mbps was offered to the network by having each flow send 256 byte packets at approximately 4.9 packets per second. The simulation time was 300 seconds for each simulation and the results were averaged over 30 random seeds. We used a slightly modified random waypoint mobility model to address the concerns raised in [20].

3.2 Byzantine Attack Simulation Module

In order to simulate most of the proposed Byzantine attacks in NS2, a protocol independent Byzantine attack simulation module was developed. This module provides the capability to simulate the black hole, Byzantine wormhole, and Byzantine overlay network wormhole attacks without modifying the routing protocol. It was not possible to simulate the flood rushing attack using this technique because it requires timing changes in the routing protocol code. This attack simulation module is potentially useful to the secure routing community, and will be made publicly available. The remainder of this section describes the module functionality. Readers that not interested in NS2 implementation details are advised to skip ahead to the next section.

The module is implemented as part of the NS2 Link Layer (LL) object which lies directly below the Routing Agent and directly above the MAC layer. The modified LL has several commands that allow it to be configured from the simulation TCL setup script. The first command enables the black hole attack, which is executed by checking the packet type of any packet sent down by the routing agent, and silently dropping any packet which has an application data type (as opposed to a routing protocol type). The second command is used to setup the various wormhole configurations, and creates a back channel connection from one node to another *wormhole peer* node. The attack module manages any number of these wormhole peer connections thus allowing the setup script to create either a simple point to point wormhole or the more complicated overlay network wormhole. As a packet is sent down from the routing protocol, its next hop address is used to determine the correct action. In addition to being sent down to the interface queue for transmission by the MAC, copies of any broadcast packets are sent to every configured wormhole peer. If the next hop address of a unicast packet matches a wormhole peer address, the packet is sent directly to that peer. Otherwise, it is sent down the stack normally.

3.3 The Black Hole Attack

A basic Byzantine attack that an adversary can execute is to stop forwarding data packets. As a result, whenever the adversarial node is selected as part of a path by the routing protocol, it prevents communication on that path from taking place. The majority of existing secure and insecure routing protocols are disrupted by black hole attacks because they can render the normal methods of route maintenance useless. More specifically, if the adversary selectively drops only data packets, while still participating in the routing protocol correctly, the normal methods of route maintenance will indicate that the route is fully operational, misleading the other nodes about the success of the data delivery.

The total network damage caused by a black hole attack is directly related to the likelihood of an adversary being selected as part of the routing paths in the network. In a dense network, there will be a large number of available paths, so the probability of selecting one containing an adversary may be small unless there is a large number of attackers. In low density networks, the number of available paths is lower, so the probability of selecting an adversarial path is higher. In addition, if adversaries have some knowledge of the network topology and/or traffic patterns, they may be able to select strategic locations which increase the effectiveness of the attack. For example, an adversary may locate itself in the vicinity of a specific target, or position itself between two nodes that communicate frequently. The effectiveness of the basic black hole attack can also be increased by combining it with the more advanced Byzantine attacks covered in later sections.

3.3.1 Attack Mitigation

Several techniques exist which attempt to mitigate the effect of black hole attacks on network performance. In this section we review the major approaches, showing their advantages and limitations.

Watchdog and Pathrater The technique presented in [12] takes advantage of the wireless shared medium by exploiting the fact that a node can overhear its neighboring nodes forwarding packets to other destinations. If a node does not overhear a neighbor forwarding more than a threshold number of packets, it concludes that the neighbor is adversarial. The approach has two components, *watchdog*, a service that is run by each node and monitors the node's neighbors, and *pathrater*, a service that ensures that adversarial nodes are avoided when selecting future routes. The scheme does not require any explicit network overhead or cryptography while being effective against the basic black hole attack in single rate fixed transmission power networks.

However, the approach is prone to many false positives and does not perform well when either power control or multi-rate (i.e. 802.11abg [21, 22]) are used, since their use will violate the assumption that the forwarding transmission is successfully overheard. In addition, the method is vulnerable to attacks from two consecutive and colluding adversaries where the first adversarial node does not report that the second did not forward the data.

Secure Data Transmission (SDT) An alternate technique for avoiding black hole attacks is the SDT protocol [23]. SDT uses authenticated end-to-end acknowledgments from the final destination, providing proof that the packets reached their destination. While this scheme always detects the presence of a black hole attack, it is unable to identify a specific adversarial node along the path. SDT sidesteps this limitation by disseminating a packet across several node-disjoint paths. The intuition is that since a path experiencing a black hole attack is known to contain an adversary, then a node-disjoint path will not contain that same adversary. The method has relatively low overhead, and works effectively in a well connected ad hoc wireless network since the number of disjoint paths can be large. The disadvantage of this technique is that in a sparsely connected network, where the number of available disjoint paths is small, all of the discovered paths may contain an attacker.

It should be noted that when using this node-disjoint path technique, it is critical to protect the method of discovering the node-disjoint paths. In the absence of such protection, both false topology and path discovery denial of service can compromise the operation of SDT (as it will be either inundated with false paths or will have no paths to choose from). In [23], the authors suggest using SDT with their Secure Routing Protocol (SRP) [6], but the modifications required to allow this protocol to find multiple node-disjoint paths are not specified. While the original SRP is fairly resilient to falsified topology when attackers act individually, it cannot fully protect against colluding attackers. Also, SRP is vulnerable to flood rushing attacks (discussed in Section 3.4), which may prevent successful route discovery.

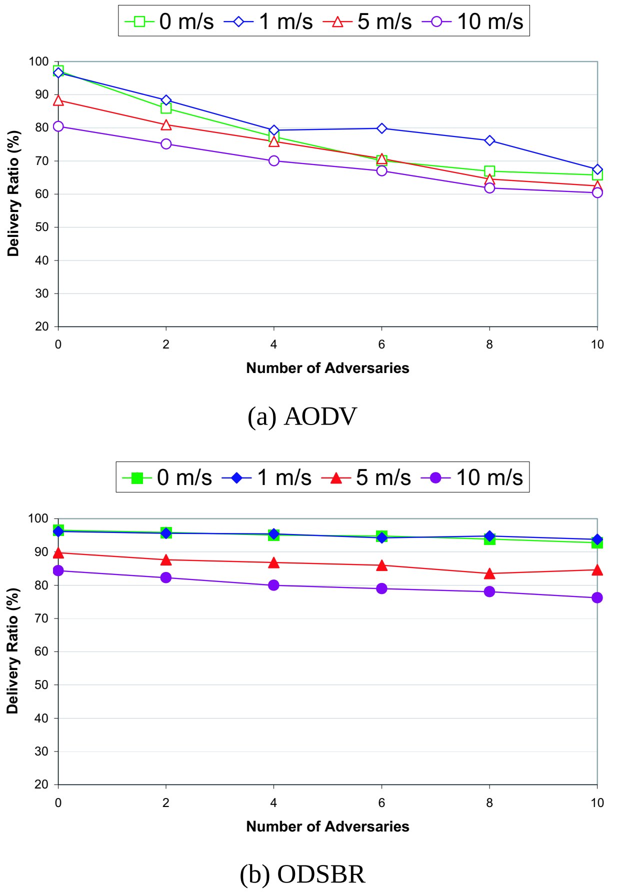
ODSBR The ODSBR [15] protocol also uses end-to-end acknowledgments from the destination to detect the presence of a black hole attack. However, unlike SDT, upon detection of the attack, ODSBR enters a probing mode with the goal of discovering the attack location. The result of this probing procedure is that the location of the adversary can be narrowed down to a link (the guilt is assigned to a link, since it is theoretically impossible to indicate a node). When a link is blamed, its weight is doubled, which ensures that the protocol will avoid selecting paths containing that link during future route discoveries. This fault locating technique shares an advantage with the watchdog approach in that the locations of the attackers are learned, thus enabling adversary avoidance in arbitrary network configurations (a large number of node-disjoint paths are not needed). Also, as in SDT, ODSBR cannot be "tricked" by an intermediate adversary into thinking that packets are being successfully delivered to the destination. As a result, if there exists an adversarial-free path to the destination, ODSBR is *guaranteed* [15] to eventually find it within a bounded amount of packet loss.

One disadvantage of the ODSBR probing technique is that it may converge slower than SDT, particularly when the number of black holes is small. This is because SDT has the ability to try many paths in parallel for each round of route discovery. Like the watchdog technique ODSBR only tries one path per round. As long as the number of adversaries is relatively small and the number of disjoint paths is large, SDT may be able to find a working path in one or two rounds, where as the other two techniques may take several rounds.

3.3.2 Simulation Results

We simulate a black hole attack by dropping any data packet sent down the stack by the routing agent. Routing protocol packets are unaffected. On a real device, depending on the routing protocol implementation, performing a black hole attack may be as simple as deactivating IP forwarding.

We evaluate the delivery ratio by using as a baseline the case where no black holes exist in the network. We then increase the number of adversarial nodes in the network and evaluate the effect this has on the delivery ratio. The adversarial nodes are placed randomly within the simulation area. Figure 1 shows the delivery ratio of the AODV and ODSBR protocols as a function of the number of adversarial nodes, for different levels of mobility. While the delivery ratio of AODV does decrease as the number of adversaries increases, a large number of adversarial nodes is required in order for this attack to cause a significant network disruption. For example, approximately 10 adversarial nodes are required to drop the delivery ratio of AODV below 70%. At low mobility, ODSBR manages to maintain a delivery ratio of about 95%, even in the presence of 10 adversarial nodes.



(a) AODV

(b) ODSBR

Figure 1. Black Hole Attack

3.4 Flood Rushing Attack

Although the basic black hole attack has a negative effect on the network performance, an adversarial node can only create disruptions if it is selected on a routing path. A stronger attack occurs when the adversary takes an active approach in disrupting not only data forwarding, but also the path discovery mechanisms. Since flooding is the main mechanism used by on-demand routing protocols to establish paths, disrupting flooding is an effective attack against these types of protocols. Attacks on flood propagation stem from the property that typically, protocols process only the first copy of a flooded packet, and discard any additionally received copies. This mechanism, also known as *flood suppression*, is required to prevent a single flooded packet from creating a never-ending series of broadcasts that would quickly consume all available medium time.

An adversarial node can exploit the flood suppression mechanism either to increase its chance of being selected as part of the path, or to prevent a valid path from being established when end-to-end authentication protocols are used. The attack requires an adversary to be able to "rush" a packet through the network, propagating the flood faster than the normal flood. This can be achieved in several ways, but one of the primary mechanisms is to ignore the flood rebroadcast delays required by the routing protocol. Another method to achieve fast propagation is the use of wormholes, which is described in Section 3.5.

During the propagation of a valid flood packet the adversary's goal is to propagate its modified flood packet to intermediate nodes *before* a flood packet reaches them through a set of valid nodes. Note that in this case source authentication will not help, because the adversary rushes authenticated data through the network. If an adversary successfully reaches some of its neighbors with its own version of the flood packet before they receive a version through a non-adversarial route, then those nodes will propagate the adversary's version of the flood and ignore any valid version of it. The result is a chain reaction where the adversarial version of the flood packet can propagate to a large fraction of the network. The chance of the adversarial node being selected on a route is considerably increased, even if the node does not lie on the shortest path, since on-demand protocols such as AODV silently discard duplicate floods [24].

3.4.1 Attack Mitigation

Most existing on-demand insecure or secure protocols are vulnerable to the flood rushing attack. Previous work in addressing the rushing attack is scarce, we are only aware of Rushing Attack Prevention [14] and ODSBR [15].

Rushing Attack Prevention (RAP) The intuition in this work is that the rushing attack can be prevented by waiting (up to a time limit w) to receive up to k requests (flood re-broadcasts) and then randomly selecting one to forward rather than always forwarding only the first one. The advantage of this technique is that the random selection probabilistically reduces the advantage gained by reaching a node first. To prevent a single attacker from bypassing the scheme by simply sending k requests, the RAP protocol incorporates secure neighbor discovery and secure route delegation schemes. However, these schemes result in a great deal of network overhead because multiple rounds of communication are required for every hop the route request propagates. In addition, RAP will continue to be ineffective if the adversary has compromised k or more nodes.

ODSBR The route discovery phase of the ODSBR protocol has several features which help mitigate the effects of flood rushing. The integrity of all the information contained in a route discovery flood packet is verifiable by every node in the network. This protects against an attack possible when using only end-to-end authentication (source and destination only), where an invalid variant of the flood can propagate through the network and block the valid flood. Also, the flood suppression mechanism in the ODSBR protocol reduces the effect of small timing differences; ODSBR processes all duplicate flood packets and if a valid flood packet with a lower metric is received, an additional re-broadcast is scheduled.

The advantage of this technique is that even if an adversary performs a successful rush in an attempt to be selected on the path, the adversarial variant of the flood will be shortly overridden by the legitimate flood if there is a lower cost legitimate path. One disadvantage of this technique is that it may cause more protocol overhead because the set of nodes affected by the rushing adversary needs to re-broadcast the flood packet more than once. Also, this technique still allows an adversary that does lie on the shortest path to gain an advantage in being selected (although this is significantly weaker than the original rushing attack). This remaining rushing advantage is negated when ODSBR identifies the fault location and increases the weight (as the rushing adversary will no longer lie on the shortest path).

3.4.2 Simulation Results

Simulations were conducted to evaluate the impact of flood rushing on the effectiveness of a black hole attack. During the propagation of a normal flood packet, each node waits a small randomized delay before re-transmitting the flood. These randomized delays are designed to reduce the number of collisions and in some protocols to help ensure that the shortest paths are selected. Eliminating the extra delay is the simplest mechanism available to provide an adversary a time advantage over the normal flood. This technique was used to simulate the flood rushing attack.

Figure 2 shows the delivery ratio of the AODV and ODSBR protocols as a function of the number of adversarial nodes, for different mobility values. Observe that compared with the results in Figure 1, for AODV, flood rushing increases the effectiveness of the black hole attack by approximately 20%. On the other hand, the impact of flood rushing on ODSBR is almost unnoticeable. Also note that, for low mobility, ODSBR delivers over 90% of the packets, even in the presence of 10 adversaries. The attack is relatively strong and lowers AODV's delivery ratio below 50% when 10 adversaries are present.

3.5 Byzantine Wormhole Attacks

The black hole attack results indicate that a large number of attackers would be required to disrupt the network using strictly black holes. Intuition would lead us to believe that if the adversary was capable of compromising some set of nodes, there would exist a more effective attack which would involve the cooperation of the adversarial nodes. One such attack is a Byzantine wormhole, which we refer to as a *wormhole*.

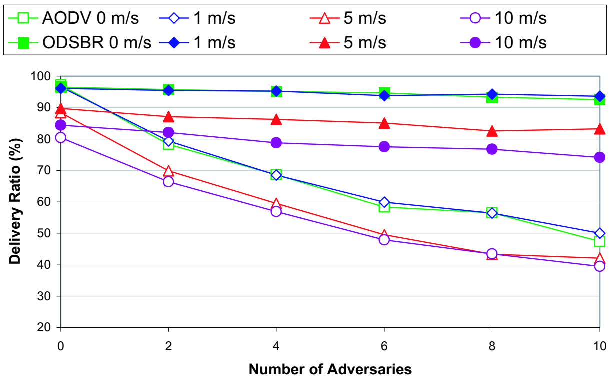


Figure 2. Flood Rushing combined with Black Hole Attack

This attack occurs when two adversaries cooperate to tunnel packets between each other in order to create a shortcut (or wormhole) in the network. Such a tunnel can be created by using a private communication channel (such as wired communication or a pair of radios and directional antennas), or by even using the existing ad hoc network infrastructure. Since the adversaries are using authenticated devices, they have complete access to use the ad hoc network. As a result, the adversaries can send a route request and discover a route across the ad hoc network. The adversaries can then tunnel packets through the non-adversarial nodes to execute the attack. This is in essence using the network against itself.

When the adversaries tunnel a route request between one another, they are able to make the route appear shorter than it actually is. By creating the appearance of a short path, the adversaries have an extremely high probability of being selected by the routing protocol. Once selected, the adversaries perform a black hole attack, by dropping the actual data packets. Also, as it allows an adversary to jump several hops ahead of the legitimate flood at once, a wormhole serves as an effective tool for conducting flood rushing attacks. Although implemented with only two adversarial nodes, this type of colluding attack is particularly strong.

It should be pointed out that the Byzantine wormhole attack considered in this work is different from the traditional wormhole attack. In the traditional wormhole attack, an adversary or multiple adversaries trick two honest nodes into believing that there exists a direct link between the honest nodes. The difference is that in the Byzantine case, the wormhole link exists between the adversarial nodes, not between the honest nodes.

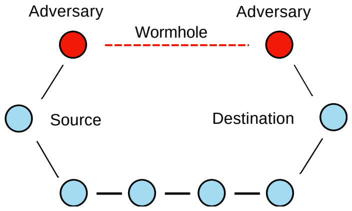


Figure 3. Simple Wormhole Configuration

3.5.1 Attack Mitigation

Packet Leashes A mechanism for preventing wormholes by limiting the transmission distance of a link is proposed in [13]. The authors propose the use of either extremely tight time synchronization (a temporal leash) or location information (a geographic leash) to restrict the maximum transmission distance, and present the TIK protocol which implements temporal leashes using efficient hash trees. Although the TIK protocol may require additional hardware (e.g. accurate clocks or GPS receivers), it is effective at preventing the traditional wormhole attack. However, the TIK protocol is ineffective against the Byzantine wormhole attack because "preventing" the wormhole link is the responsibility of its end points. In this case the end points are adversarial and cannot be trusted to follow the protocol.

Directional Antenna A more recent method for preventing wormholes uses the angle of arrival information available when using directional antennas [25]. This approach takes advantage of the topology distortion that occurs when nodes communicate through a wormhole in order to prevent wormhole links from being used. Due to the wormhole detection geometry, a third node in a particular region is required to completely verify the link between two nodes. If no node is available in the verifier region, then the link cannot be used even if it is indeed valid. As a result, this strategy will reduce the number of available links in the network, particularly in low density networks. In addition, while this scheme is effective against traditional wormholes, it does not prevent Byzantine wormhole attacks because the adversarial end points will use the wormhole link without verification. In addition, if Byzantine adversaries are present near either end of a traditional wormhole, they can falsely "verify" wormhole links between good nodes.

ODSBR Unlike the previous schemes that focus only on wormhole prevention, ODSBR takes a completely different approach. The authors of ODSBR observed that the primary attack is the dropping of data packets that attempt to travel through the wormhole, rather than the actual wormhole formation itself. As a result, the authors claim that preventing the wormhole is not strictly necessary. A wormhole attack will appear to ODSBR as a faulty link existing between two nodes. ODSBR mitigates the attack not by preventing the formation of this link, but by increasing its weight if it lies on the path and is discovered to be faulty. Once the wormhole's link weight has been increased sufficiently, ODSBR will avoid it and select the next best alternate path to the destination.

The advantage of this strategy is that it does not require any additional hardware or capabilities to function, and it works equally well for Byzantine and traditional wormholes. The main disadvantage of the ODSBR strategy is that because it uses fault locating techniques, avoiding wormhole links requires the protocol to lose a number of packets. In addition, the number of packets lost and the amount of time taken while "finding its way," will be proportional to the number of wormhole links that create paths shorter than the legitimate route. As a result, ODSBR's ability to mitigate the wormhole attack will be reduced if many wormhole links are present.

In the remainder of this section, we provide a simple step by step example of how ODSBR operates to detect and avoid the effects caused by a wormhole attack. Consider the network topology presented in Figure 3. The network consists of a single source and destination and a valid non-adversarial path between them. In addition, there exists a single wormhole in the network. The wormhole in the network makes it appear to the routing protocol that the shortest path is only three hops, when in reality the only fault free path in the network is five hops. The AODV protocol will continue to select the three hop path since it appears shorter and will never discover a working route. The ODSBR protocol will also initially select the shorter three hop path. When ODSBR attempts to route packets across the adversarial controlled path, it will detect that the path behaves maliciously and drops packets. ODSBR will then enable probing on the path to detect the fault location. Once the fault location is detected, the weight of the faulty link is doubled. ODSBR will now initiate a second route request. After doubling the weight of the link, ODSBR will still select the adversarial path that has a cost of four. At the next request, the faulty link will be incriminated again and its weight doubled. As a result, ODSBR will discover the fault free path, since it has a lower cost than the adversarial path.

3.5.2 Simulation Results

Through simulation we evaluate the impact of the Byzantine wormhole attack on the AODV protocol and demonstrate the effectiveness of the ODSBR protocol in mitigating this attack. We simulated the most effective wormhole attack, by assuming communication through the wormhole tunnel to have no latency and unlimited bandwidth. The simulations for attacks involving wormholes represent an upper bound on the damage adversaries can cause, because the adversarial communication channel would be more limited in reality.

After simulating the configuration presented in Figure 3, we observed that the AODV protocol achieved a delivery ratio of 0%. Any protocol that relies only on authentication will have a delivery ratio of 0%, since the wormhole is also an authenticated path. In contrast, ODSBR delivered 94.7% of the packets.

The above example demonstrates the power of the wormhole attack in a small static network configuration. In order to estimate its effectiveness in a large mobile ad hoc network, additional simulations are required. In the following set of simulations a static wormhole configuration is placed within the network. The non-adversarial nodes will remain mobile and the disruption caused by the wormholes will be evaluated. We investigated three configurations which we refer to as *central wormhole*, *cross of death* and *random placement*. In all cases, we evaluated both the effect of the wormhole attack by itself, and when combined with flood rushing. As in the case of the black hole attack, flood rushing increased the effectiveness of the wormhole attack against AODV.

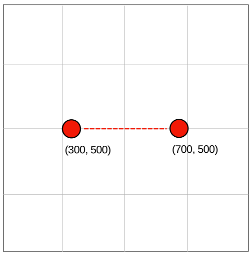


Figure 4. Central Wormhole Configuration

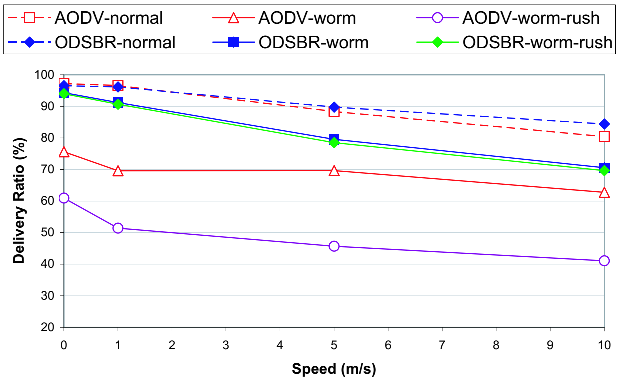


Figure 5. Wormhole Attack: Central Wormhole Configuration

Central Wormhole. As seen in Figure 4, this configuration contains only two adversaries placed at coordinates (300,500) and (700,500) in the 1000 x 1000 area considered for our simulations. The results presented in Figure 5 show the delivery ratio as a function of the mobility of the nodes, for AODV and ODSBR. In addition, the normal delivery ratios in the case of no adversaries are shown for reference. Although only one wormhole is present, this attack causes a fairly large amount of disruption to AODV, especially in the presence of flood rushing. When compared with results for the black hole attack (Figure 1 and Figure 2), two strategically placed adversaries that are able to cooperate can considerably increase the effectiveness of the attack. For example, when flood rushing is enabled and two attackers coordinate to form a *central wormhole*, AODV's delivery ratio can drop as low as 41%, which is similar in strength to ten randomly placed adversaries performing the black hole attack, where AODV delivers 39% of the packets. The explanation is simple: the adversaries are strategically placed towards the center of the simulation area and since many of the routes will pass through their range, the adversaries can effectively be selected on many routes.

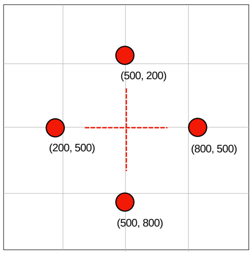


Figure 6. Cross of Death Configuration

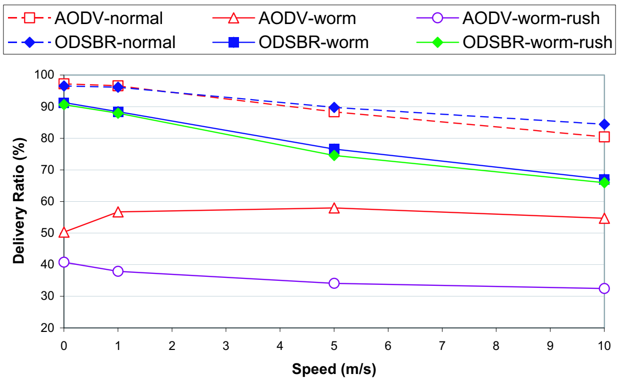


Figure 7. Wormhole Attack: Cross of Death Configuration

Cross of Death. As seen in Figure 6, this configuration contains four adversaries placed at coordinates (200,500), (800,500), (500,200), (500,800). They form two wormholes, in the shape of a cross. The results presented in Figure 7 show the delivery ratio as a function of the mobility of the nodes, for AODV and ODSBR. As we expected, this is a more effective attack against AODV than the *central wormhole* attack, since the adversarial nodes are covering a larger area and are able to draw in (and drop) more traffic; however, ODSBR is barely affected by the increase in the number of adversaries from two to four.

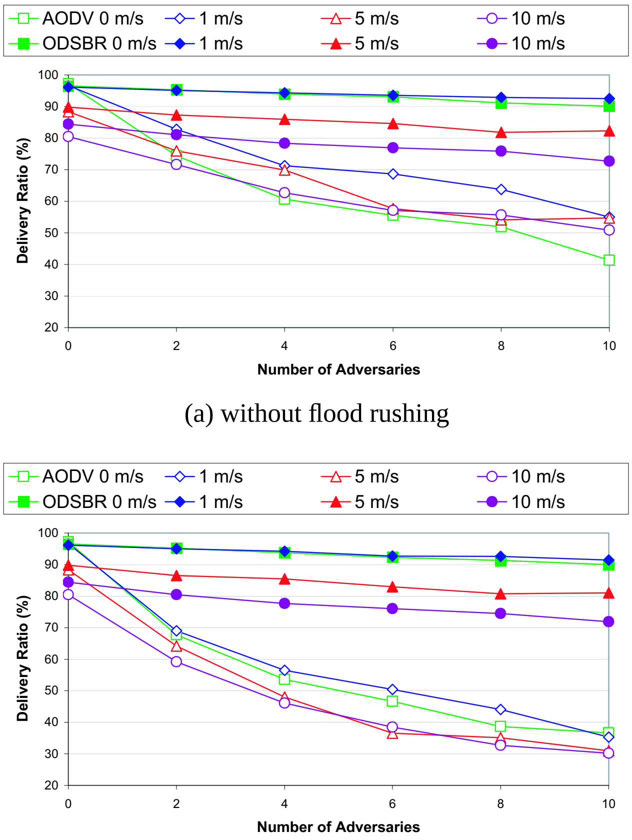
Random Placement. The last configuration we consider is where a set of wormholes is randomly placed in the network. We perform simulations to investigate how many randomly placed wormholes are required to provide the same amount of damage as a strategically placed attack. Figure 8(a) presents results for AODV and ODSBR in the presence of the wormhole attack, while Figure 8(b) presents results for the wormhole attack combined with flood rushing.

When compared to the black hole attack with randomly placed adversaries (Figure 1 and Figure 2), the same number of adversaries placed randomly, but now forming wormholes, can mount a more effective attack against AODV. This confirms our expectations that by using wormhole tunnelling, the adversaries are selected as part of more routes and are able to drop more traffic. As opposed to AODV, observe that ODSBR is much more resilient to the change from black holes to wormholes, and is practically unaffected by the addition of flood rushing.

By analyzing Figures 5, 7 and 8, we can determine the number of randomly placed adversaries required to inflict the same amount of damage as a strategically placed attack. We conclude that for AODV, with mobility $>$ 0 m/s, the *central wormhole* configuration inflicts slightly more damage than 4 randomly placed adversaries (2 random wormholes) and the *cross of death* inflicts slightly more damage than 8 such adversaries (4 random wormholes). In the case of ODSBR, both the *central wormhole* and the *cross of death* cause more damage than 10 randomly placed adversarial nodes (5 wormholes). This indicates that the wormhole attack is more effective if the adversaries are strategically placed, rather than randomly placed.

3.6 Byzantine Overlay Network Wormhole Attacks

In Section 3.5 we analyzed the case where the wormholes were just point-to-point tunnels between two adversaries. While this attack is strong and effective, an even stronger variant exists, where a set of nodes organized in an overlay network are under the control of an adversary or a set of colluding adversaries. We refer to this attack as a Byzantine overlay network wormhole, or a *super-wormhole*.



(a) without flood rushing

(b) with flood rushing

Figure 8. Wormhole Attack: Random Placement

In a super-wormhole attack, n adversaries use the existing ad hoc network infrastructure to create an overlay network between all of them. There exist essentially n^2 point-to-point tunnels between the adversaries. When an adversary receives a route request packet, it sends it out all of its tunnels to the other adversaries in the network. When they receive the packet they rebroadcast it as if they had just received a route request. By using an overlay network strategy, the adversaries are able to perform a much stronger attack and greatly increase their chances of being selected by the routing protocol.

By using the super-wormhole attack, the adversary can draw a massive amount of the routing protocols traffic into the wormholes and cause a significant disruption in the network. One can object that this attack is not really feasible in practice, given the large number of point-to-point tunnels required to be established between the adversaries. However, as shown in the simulations, even a small number of adversaries can cause a major disruption in the network, making this attack a lot more practical and easier to mount.

3.6.1 Attack Mitigation

To our knowledge, ODSBR is the only protocol designed to mitigate the super-wormhole attack. More precisely, ODSBR provides an theoretical upper bound on the number of lost packets as a function of the number of links that are controlled by an adversary. More details about the analysis can be found in [15].

3.6.2 Simulation Results

Through simulation we evaluate the damage caused to AODV by a set of attackers performing a coordinated super-wormhole attack, and demonstrate the effectiveness of the ODSBR protocol in mitigating this attack. Similar to the wormhole attack, communication through the super-wormhole tunnels is instantaneous, so the simulations should be seen as an upper bound on the amount of damage a super-wormhole can cause.

In the following set of simulations a static wormhole configuration is placed within the network. We investigated three configurations which we refer to as *cross of death*, *random placement* and *complete coverage*. In all cases, we have first evaluated the effect of the super-wormhole attack on the delivery ratio. We then combined the super-wormhole with flood rushing and examined the impact of the combined attack.

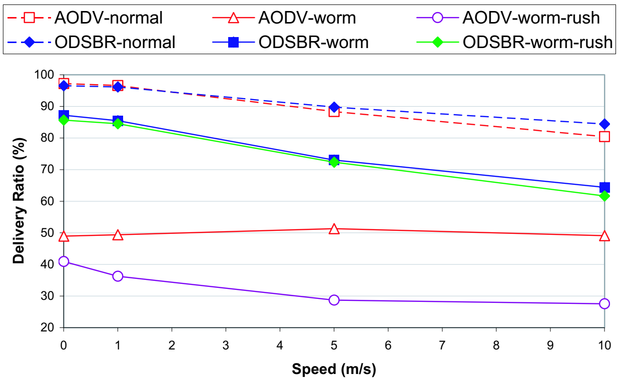


Figure 9. Super-Wormhole Attack: Cross of Death Configuration

Cross of Death. The same configuration as the *cross of death* in Section 3.5.2 was used, but with all four adversarial nodes connected in a super-wormhole configuration. The results presented in Figure 9 show the delivery ratio as a function of the mobility of the nodes, for AODV and ODSBR, both with and without flood rushing. In addition, the normal delivery ratios in the case of no adversaries are shown for reference. Observe that the attack is slightly more effective than the *cross of death* with regular wormholes (Figure 7). This indicates that the additional tunnels created in the super-wormhole scenario are of limited strategic value in comparison to the primary tunnels.

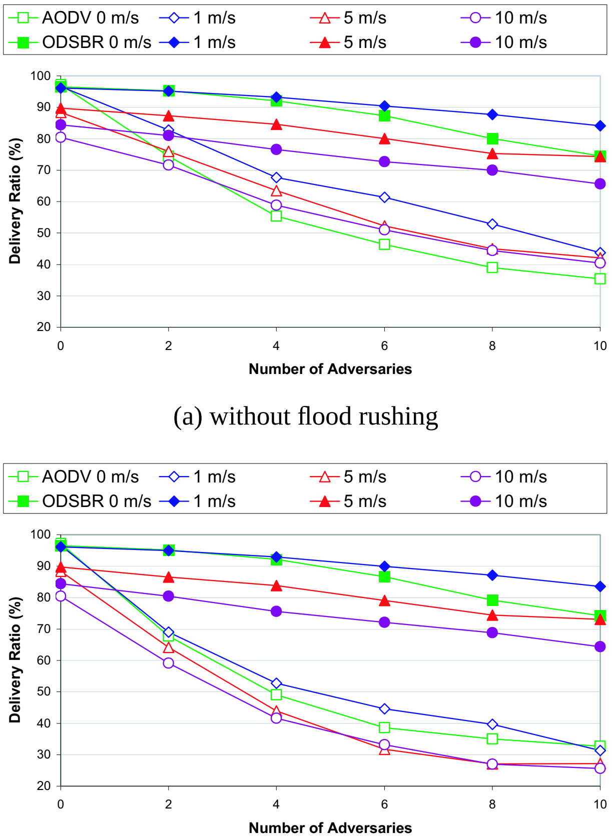
Random Placement. The next configuration we consider is where a set of adversarial nodes are randomly placed in the network and form a super-wormhole. We perform simulations to investigate how many randomly placed adversaries are required to provide the same amount of damage as a strategically placed attack. Figure 10(a) presents results for AODV and ODSBR in the presence of the super-wormhole attack, while Figure 10(b) presents results for the super-wormhole attack combined with flood rushing.

In this case, both for AODV and ODSBR, the super-wormhole attack is more effective than the regular wormhole attack, though not by much. This leads us to believe that a super-wormhole created by randomly placed adversaries will give little advantage over the case when the same number of adversaries create regular 1-to-1 wormholes.

By analyzing Figures 9 and 10, we can determine the number of randomly placed adversaries required to inflict the same amount of damage as a strategically placed attack. We conclude that for AODV, with mobility $>$ 0 m/s, the *cross of death* configuration inflicts slightly less damage than a super-wormhole created by 8 randomly placed adversaries. In the case of ODSBR, if mobility $>$ 0 m/s, the *cross of death* causes approximately the same damage as a super-wormhole created by 9 randomly placed adversaries if flood rushing is not used, or 10 adversaries if flood rushing is enabled.

Complete Coverage. The strength of the super-wormhole attack can be increased significantly if the adversaries are able to properly position themselves throughout the network. For this particular attack a dominating set adversarial configuration would provide the strongest attack. In a dominating set configuration, the adversaries attempt to arrange themselves so that their combined communication areas completely cover the full ad hoc network. This means that if any transmission takes place in the network, an adversary will hear it. The dominating set configuration does not have to be a connected dominating set, as long as the adversaries remain connected through other nodes in the network. As a result of this configuration, the adversaries can make any path longer then three hops appear to be exactly three hops. We simulated the configuration shown in Figure 11 , with five adversarial nodes placed at coordinates (250,250), (250,750), (500,500), (750,250), (750,750).

Observe the devastating effect of this attack in Figure 12. When combined with flood rushing, the delivery ratio of AODV drops as low as 20% in the presence of five adversaries, while ODSBR still delivers 60% of the packets. Since the five adversarial nodes almost completely cover the entire ad hoc network, adding more adversaries



(a) without flood rushing

(b) with flood rushing

Figure 10. Super-Wormhole Attack: Random Placement

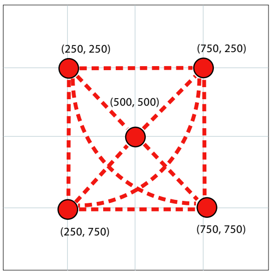


Figure 11. Complete Coverage Configuration

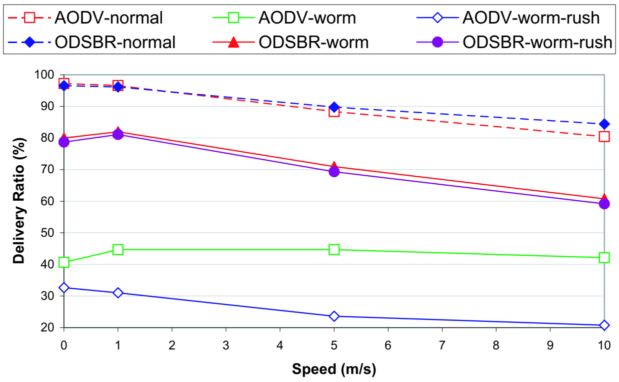


Figure 12. Super-Wormhole Attack: Complete Coverage Configuration

will not significantly increase the effectiveness of the attack. It is worth reiterating that the super-wormhole attack is extremely powerful: a set of only five colluding adversaries can practically paralyze the considered ad hoc network when an insecure routing protocol is used, and can cause serious problems even when a secure Byzantine routing protocol is used.

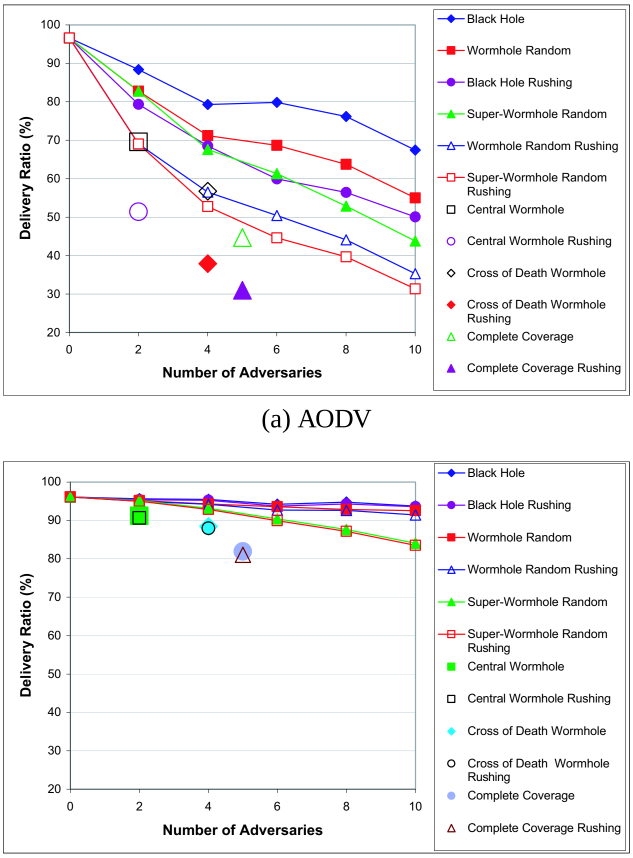
4 Discussion

4.1 Attack Strength Evaluation

In this section we provide a comparison of the simulation results from Section 3, in order to determine the relative strength of the Byzantine attacks (see Figure 13). To evaluate the effects of these attacks in a mobile ad hoc network, we selected scenarios where the mobility of the nodes was 1 m/s. This value was chosen rather than higher mobility values in order to better isolate the damage caused specifically by the Byzantine attacks as opposed to losses due to node mobility.

Analysis of these results indicates that two main factors contribute to the effectiveness of the attacks at disrupting the AODV routing protocol: flood rushing and strategic adversarial positioning. We will discuss the effects of these two techniques on the packet delivery ratios, and then explore the damage resulting from their combination.

Flood Rushing. In Figure 13, the line labelled "Black Hole Rushing" shows the results of a random placement black hole attack where flood rushing was enabled. Observe that by enabling flood rushing, this attack resulted in a much greater reduction in the delivery ratio as compared to the same attack without flood rushing. In addition, the flood rushing made this attack strong enough that it caused more damage then the random wormhole attack and comparable damage to the random super-wormhole attack. The fact that the black hole attack (a non-colluding attack and simpler to execute), combined with flood rushing can create more damage than the wormhole attack (a colluding attack and harder to mount) is an important observation. This motivates the need to design routing protocols which are able to mitigate the flood rushing attack.



(a) AODV

(b) ODSBR

Figure 13. Attacks Comparison

Strategic Positioning. The results indicate that the strength of the attacks can be significantly increased if the adversaries are strategically positioned. The point labelled “Complete Coverage” in Figure 13 is an illustrative example of the effectiveness of strategic positioning. These are the results of a super-wormhole with adversaries arranged in a dominating set configuration. By being strategically placed, five adversaries are able to reduce the delivery ratio of AODV to just 45%, without using flood rushing. In comparison, six randomly placed adversaries executing a super-wormhole attack, are only capable of reducing the delivery ratio of AODV to 61%. This demonstrates the power of strategic positioning in crippling the performance of the AODV routing protocol.

Flood Rushing + Strategic Positioning. While each of these two techniques can cause substantial damage to the routing protocol, their combination is even more destructive. We define the relative strength of a particular attack configuration σ as:

$$\sigma = \frac{DR_{norm} - DR_{adv}}{DR_{norm} \cdot Num_{adv}} \qquad (1)$$

where DR_{norm} and DR_{adv} are the delivery ratios in the absence or in the presence of adversaries respectively, and Num_{adv} is the number of adversaries. Intuitively, σ represents the amount of damage an attack can cause per adversary. The higher σ is, the greater the relative strength of the considered attack, since this indicates that a larger amount of damage can be inflicted by a smaller number of adversaries.

Observe that in the “Complete Coverage Rushing” case we see the delivery ratio drops to 30%. Although this point corresponds to an attack that results in the greatest reduction of AODV’s delivery ratio, this does not necessarily mean that the relative strength of the attack is the highest. In this case $\sigma = 13.6$. Alternatively, we can consider the point referred to as “Central Wormhole Rushing” in Figure 13. This attack is able to lower AODV’s delivery ratio by from 96.6% to 51.4%, while requiring only two colluding adversaries, thus $\sigma = 23.4$. In fact, this is the highest σ observed out of all considered attacks. This colluding attack executed by only two adversaries combines both flood rushing and strategic positioning, and inflicts the highest amount of damage with the least number of adversaries.

4.2 ODSBR Mitigation and Vulnerabilities

In this section we present a summary of the 1 m/s simulation results for the ODSBR protocol (see Figure 13) in order to analyze its ability to mitigate the attacks simulated in Section 3. The first observation is that at this level of mobility, the ODSBR protocol was able to successfully deliver over 80% of the packets under all simulated attack scenarios. This validates the protocol’s overall strategy for operation in a Byzantine environment. In particular, the results show that ODSBR is resilient against flood rushing attacks which we have shown are devastating to other existing on-demand protocols.

Although ODSBR performed well overall, the results show that the strategically placed wormhole configurations (and to a lesser extent the random super-wormhole configuration) result in significantly lower delivery ratios than the other attack scenarios. The common element in these attacks is that they are particulary effective in creating adversarial controlled paths that appear shorter than legitimate network paths. ODSBR is initially lured into using these paths, and must incriminate them at the cost of losing packets. Although ODSBR will always eventually find a fault-free path if one exists, the more adversarial links the protocol has to detect, the greater the number of packets lost.

5 Related Work

In this section we provide an overview of additional related work conducted in the area of securing ad hoc wireless routing protocols not already discussed in the body of the paper.

Source authentication is more of a concern in routing than confidentiality. Papadimitratos and Haas showed in [6] how impersonation and replay attacks can be prevented for on-demand routing by disabling route caching and providing end-to-end authentication using an HMAC [19] primitive which relies on the existence of security associations between sources and destinations. Other significant works in this aspect include SEAD [4] and Ariadne [3] who provide efficient secure solutions for the DSDV [16] and DSR [26] routing protocols, respectively. While SEAD uses one-way hash chains to provide authentication, Ariadne uses a variant of the Tesla [27] source authentication technique to achieve similar security goals. In [5] the authors focus on an analogous problem, providing end-to-end authentication for two well-known on-demand protocols: AODV [28] and DSR [26]. The difference is that they are using a strong, but expensive, authentication means: digital signatures. They also provide an expensive protocol that guarantees minimum path selection using an onion [29] like technique, where digital signatures and public cryptography encryption/decryption are performed and accumulated at each hop. The destination node strips off the encryption/signed layers to determine the path.

6 Conclusions

Through simulation, we performed a quantitative evaluation of the impact of a wide range of Byzantine attacks on the insecure AODV routing protocol. We analyzed the relative strength of these attacks in terms of the magnitude of disruption caused per adversary. We conclude that flood rushing and strategic positioning of adversaries are the two most important factors for an effective attack against insecure on-demand protocols. Our experiments showed that only two adversaries forming a central wormhole combined with flood rushing can mount an attack that has the highest relative strength. This attack is relatively easy to execute since it requires only two colluding adversaries, and is able to reduce the delivery ratio to 51%. We also showed that ODSBR was able to mitigate a wide range of Byzantine attacks; in particular, it was not significantly affected by flood rushing. Its performance only decreased when it needed to detect and avoid a large number of adversarial links.

References

[1] "CSI/FBI computer crime and security survey," *CSI Computer Security Institute*, vol. 8, 2003.

[2] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," in *Proceedings of the IEEE*, vol. 75, January 1987.

[3] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *The 8th ACM International Conference on Mobile Computing and Networking*, September 2002.

[4] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," in *The 4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[5] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," in *10th IEEE International Conference on Network Protocols (ICNP'02)*, November 2002.

[6] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, pp. 27–31, January 2002.

[7] R. Hauser, T. Przygienda, , and G. Tsudik, "Reducing the cost of security in link-state routing," in *Symposium of Network and Distributed Systems Security*, 1997.

[8] B. R. Smith, S. Murthy, and J. Garcia-Luna-Aceves, "Securing distance-vector routing protocols," in *Symposium on Networks and Distributed Systems Security*, 1997.

[9] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)," *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 4, 2000.

[10] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," in *Advances in Ultra-Dependable Distributed Systems, N. Suri, C. J. Walter, and M. M. Hugue (Eds.), IEEE Computer Society Press*, 1995.

[11] P. Kyasanur and N. Vaidya, "Detection and handling of MAC layer misbehavior in wireless networks," in *International Conference on Dependable Systems and Networks (DSN'03)*, 2003.

[12] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *The 6th ACM International Conference on Mobile Computing and Networking*, August 2000.

[13] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *Proceedings of the 22^{nd} Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, April 2003.

[14] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *ACM Workshop on Wireless Security (WiSe)*, 2003.

[15] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *ACM Workshop on Wireless Security (WiSe)*, September 2002.

[16] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994.

[17] “The network simulator - ns2.” http://www.isi.edu/nsnam/ns/.

[18] R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[19] *The Keyed-Hash Message Authentication Code (HMAC)*. No. FIPS 198, National Institute for Standards and Technology (NIST), 2002. http://csrc.nist.gov/publications/fips/index.html.

[20] J. Yoon, M. Liu, and B. D. Noble, “Random waypoint considered harmful,” in *INFOCOM ’03*, (San Francisco, CA), April 2003.

[21] *IEEE Std 802.11a-1999*. http://standards.ieee.org/.

[22] *IEEE Std 802.11b-1999*. http://standards.ieee.org/.

[23] P. Papadimitratos and Z. Haas, “Secure data transmission in mobile ad hoc networks,” in 2^{nd} *ACM Workshop on Wireless Security (WiSe)*, 2003.

[24] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*. IETF - Network Working Group, The Internet Society, July 2003. RFC3561.

[25] L. Hu and D. Evans, “Using directional antennas to prevent wormhole attacks,” in *NDSS 2004*, 2004.

[26] D. B. Johnson, D. A. Maltz, and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. in Ad Hoc Networking*, ch. 5, pp. 139–172. Addison-Wesley, 2001.

[27] A. Perrig, R. Canetti, D. Song, and D. Tygar, “Efficient and secure source authentication for multicast,” in *Network and Distributed System Security Symposium*, February 2001.

[28] C. E. Perkins and E. M. Royer, *Ad hoc Networking*, ch. Ad hoc On-Demand Distance Vector Routing. Addison-Wesley, 2000.

[29] P. F. Syverson, D. M. Goldschlag, and M. G. Reed, “Anonymous connections and onion routing,” in *IEEE Symposium on Security and Privacy*, 1997.