Stealth Attacks on Ad-Hoc Wireless Networks

Markus Jakobsson
RSA Labs
174 Middlesex Turnpike
Bedford, MA 01730, USA
www.markus-jakobsson.com

Susanne Wetzel
Stevens Institute of Technology
Department of Computer Science
Castle Point on Hudson
Hoboken, NJ 07030, USA
swetzel@cs.stevens-tech.edu

Bülent Yener
Rensselaer Polytechnic Institute
Department of Computer Science
Lally 310, 110 Eighth Street
Troy, NY 12180-3590, USA
yener@cs.rpi.edu

***Abstract*— We study two classes of attacks that can be mounted by manipulation of routing information and exhaustive power consumption. Our attacks allow an attacker to partition a network, reduce its goodput, hi-jack and filter traffic from and to victim nodes, and thereby eavesdrop and perform traffic analysis. The methods described are "stealth attacks" in that they minimize the cost to and visibility of the attacker. We introduce the notion of *reputation based control*, and suggest that it can be used to augment existing routing protocols in order to immunize these against stealth attacks.**

I. Introduction

Most of today's communication infrastructure is based on altruistic collaboration among routers. Its robustness depends directly on the assumptions that errors are benevolent, and that there is no malicious entity wishing to disrupt communication or isolate chosen network nodes. However, given the increased economic reliance on a working communication infrastructure, this has become a potential future target for terrorists and other criminals. There are several ways in which an attacker could wreak havoc in general communication networks, and in mobile ad-hoc networks in particular. While a very powerful attacker could reduce the goodput of virtually any network simply by injecting trash messages or jamming the channels, these are attacks with a high cost for the attacker, and with a high visibility. Therefore, only powerful and dedicated attackers would have any hope of succeeding with such attacks for any extended period of time. However, as we will show, there are other attacks with lower cost and visibility, but which are at least as harmful as brute force attacks. These allow a skilled but not very powerful attacker to target communication networks in a way that makes it unlikely that he gets traced and caught. We call such attacks *stealth attacks*.

A. Stealth Attacks

We will study two principal types of attacks. In a *first* type of attack, the adversary wishes to disconnect the network, whether this means a general partition of the network or the isolation of particular nodes. (A related attack does not aim to partition the network, but to merely degrade the goodput of a network, whether globally or locally.) The well-known Denial of Service (DoS) attack is an attack with the same goal; in most such attacks, the adversary causes large amounts of traffic to be sent to a victim from some set of nodes he controls. We consider a version of this attack in which the adversary does *not* need to *control* nodes, but where he manipulates the routing information of honest nodes simply by communicating with these, thereby forcing *honest messages* (i.e., messages originating from these honest parties) to cause disruption. Thus, and similar in spirit to what was done in [8], the manipulated nodes are unaware of their involvement in the attack. We describe how the attacker can modify the behavior of such nodes by tricking them to incorrectly modify their routing tables. Given the low exposure of the attacker during this act, this is a stealth version of the common (distributed) denial of service attack.

In a *second* type of stealth attack, the adversary modifies routing information in order to hi-jack traffic from and to selected victim nodes. This, can be used to perform traffic-analysis, and may be combined with selective filtering of packets, which in turn can be used to make selected routers "disappear", as in the first type of stealth attack. The hi-jacking attack is perpetrated *remotely*, by abuse of routing protocols and detouring the messages. In other words, the type of eavesdropping we consider is *active* in that the attacker is outside the transmission range of the victim, from where he is performing the eavesdropping by detouring the traffic through corrupted nodes in the transmission range of the victim. (We note that passive eavesdropping, i.e., eavesdropping within the local transmission range of a victim, is straightforward in any broadcast protocol, but cannot easily be combined with filtering.)

In both of the above described attack types, the adversary's goal is not only to successfully perform the attack, but also to do so with a minimal effort, and in a way that hides his existence and whereabouts to the largest possible extent. From the attacker's point of view, a stealth attack is better than an attack that requires a larger amount of his energy and which leaves him more exposed to detection. In turn, this means that a routing protocol that is immune against stealth attacks is better than one that is not. Seeing this, we propose design techniques that can be used to strengthen protocols against stealth attacks. We introduce the notion of *reputation based control* and discuss how to apply such techniques to existing routing protocols. While it is easy to see that the use of our techniques strengthens the augmented protocols, there remain many open issues relating to how to best use such techniques.

B. Weapons

The attacker can employ three weapons to perform the above-mentioned attacks. A first weapon is *impersonation*, which is the introduction of packets with stated originators different from the real originators. (We do not consider it an impersonation attack if the stated originator is a cheater, but only if it is an honest party.) Practically speaking, impersonation can be performed by spoofing of IP addresses, or by using communication frequencies that have been assigned for others. If cryptographic techniques are used, impersonation requires the forging of authentication fields.

A second weapon is for the attacker to *lie*. With this, we mean to propagate incorrect information, such as incorrect routing tables. Note that an attacker may combine impersonation and lying by sending incorrect information that appears to originate from an entity other than himself. Both impersonation and lying are components of attacks of *Byzantine* nature. For purposes of clarity, and for technical reasons, we separate the two components as described above.

A third and final weapon is what we refer to as *overloading*. This is the technique that has been proposed for mounting Denial of Service (DoS) attacks. In an overloading attack, the attacker injects messages that he knows are invalid. Invalid messages can be due to (i) integrity violation (e.g., attacker flips some bits), (ii) message replay (i.e., attacker stores some valid messages and keeps resending them), or (iii) simply creating junk messages (e.g., spoofing some fake source IP addresses or adding incorrect authentication fields).

These will be detected and filtered as invalid, but filtering is computationally costly for the receiving router (e.g., it needs to buffer the incoming packet, check the header, and *verify* the checksum). By means of a repetitive overloading attack, an attacker can put a target router into a *busy-trashing* mode of operation in which no useful work gets done.

C. Tradeoffs

In general, the overloading weapon in itself does not correspond to a stealth attack, as it requires active involvement of the attacker for each offense. However, this weapon can be quite effective in the control plane operations such as route discovery or routing table update. Furthermore, in order to harden a network against attacks that employ the two first types of weapons, it is *necessary* to introduce elements whose very nature empower attacks using the third type of weapon. For example, in order for an attacker to avoid filtering based on his own source IP address, he can spoof the source IP addresses of honest routers. If the target router decides to filter out these source IP addresses, this would enable a very simple DoS attack (on the claimed sender) in which the attacker spoofs packets from nodes whose traffic he wishes to have ignored. Adding cryptographic defense techniques does not avoid DoS attacks. In fact, their inherent computational cost *enable* DoS attacks – on the server performing the verification.

Thus, in trying to immunize a system against stealth attacks, an appropriate balance has to be struck with the defense against other attacks. This insight, along with our proposed techniques, may constitute valuable guidance in designing routing protocols that are strengthened against abuse.

D. Security

From the argument above it should be clear that we do not believe that there is any way to fully secure existing open networks against attacks of the types we introduce. However, it also shows that it is meaningful to categorize various attacks with respect to their impact, and with respect to the effort they require from the adversary. It is possible to *strengthen* a network against attacks – but not *securing* it – by designing it to require a large effort for a small impact. In this paper, we elaborate on possible ways for an attacker to implement the above listed weapons, and ways in which he could use them to perform the attacks. We also discuss techniques for augmenting existing routing protocols and design new ones, such that the resulting routing protocols become less prone to stealth attacks.

E. Outline

This paper is organized as follows. In Section I we have introduced the notion of stealth attacks along with the weapons to perform the attacks. In Section II we give an overview of relevant previous work and detail our contributions. In Sections III and IV we discuss the network and security models and review details on routing algorithms. In Section V we introduce the building blocks of the attacks which are presented in Section VI. In Section VII we discuss countermeasures and introduce the notion of *reputation based control* to achieve improved routing security.

II. Related Work

A. Previous Efforts

Securing networks in general and routing in networks in particular has been studied widely. To date, however, most discussions focus on the traditional setting of static, wired network. As pointed out in [9], [11], [16], [19], mobile and in particular ad-hoc networking abilities introduce features that end up behooving attackers as well as honest users.

A first step in securing a network against attacks is to understand the nature of the attacks, and classify them with respect to how they are performed. A good example of an effort to describe attacks is a paper by Stajano and Anderson [16]. Therein, issues of confidentiality, authenticity, integrity and network availability are discussed on a high level, and various attack scenarios are described. Another example of this approach is the work by Lundberg [11], containing a brief description of some attacks on routing tables (such as black hole and overflow), and the introduction of power attacks, i.e., attacks exhausting the power resources of victims. Lundberg also describes how standard cryptographic techniques, such as digital signatures, can be employed to address problems. Thus, this is based on the unspoken assumption that routing information can be secured the same way as data traffic. Similarly, Zhou and Haas [19] describe threats and present a unified view of useful techniques from the field of cryptography. Thus, they

describe known techniques to deal with mobile adversaries, spoofing, etc.

A common drawback of the use of protection mechanisms such as those suggested in [11], [19] is that these mechanisms are not light-weight: they may not be applicable to small, mobile devices, such as those found in ad-hoc networks. Therefore, the employment of these techniques may cause more problems than they solve. In the extreme case, this holds plainly because the techniques are too expensive to use even when the system is not under attack. Moreover, and as we describe in our work, cryptographic immunization techniques can also be abused by attackers, most notably for DoS attacks. Thus, *too good* protection against one type of attack may not be desirable, as it may enable attacks on the protection mechanisms themselves. More particularly, if the effort of performing an operation – such as verifying a digital signature, performing key exchange, or updating secrets [12] – is an expensive operation, then an attacker can bring down a router by injecting bogus messages that the victim has to verify.

We are not the first to emphasize the importance of light-weight security primitives for use in wireless networks. In a recent contribution, Hu et al. [6] study another type of attack on routing schemes – the so-called replay attack – and propose an authentication technique suitable to protect against the attack. Our work takes a higher-level view of the problem of secure routing, considering the possible attacks in greater generality, resulting in a different type of proposed defense mechanism.

B. Our Contribution

The contribution of this paper is twofold. First, we aim to further the understanding of threats to routers by elaborating on attacks outlined in the above described work. Thus, we detail how attacks can be perpetrated, and categorize the techniques available to an attacker. The benefit of doing this is the improved understanding of the threat, and thereby, of the necessary techniques for securing against it.

A second contribution of our paper is a foundation for secure protocol design. To this extent, we introduce and describe general techniques useful for protection of routing information. An important insight enunciated in our paper is how the protection of one type of attacks weakens the network against a *second* type of attacks. Thus, network protection becomes a matter of delicate balance.

However, the need for balance creates a complexity that, in turn, makes it difficult to select optimal parameters and to prove that these are well chosen. We present heuristic approaches and argue their security. Our novel strategy makes use of (1) *reputation* based control, and techniques for evaluating good vs. bad behavior of fellow routers, along with (2) authentication mechanisms for use within such a system. In particular, we propose the use of message authentication codes or related lightweight tools. Thus, a particular aspect of our authentication mechanisms is that they do not require the use of a central Public Key Infrastructure. Rather, and in line with the idea of reputation based control, keys are created when first needed; the trust associated with individual keys (and their owners) develops over time.

We only lightly touch on different ways of evaluating whether a router lives up to its reputation. To improve the evaluation, more such techniques could be added, whether before a system is built, or after new attacks have been discovered. While this leaves a lot of room for flexible modification and optimization with respect to new situations, it also creates a dilemma: The evaluation techniques are based on heuristics, making rigid analysis near-impossible. This is not uncommon in systems dealing with ”fuzzy” and evolving threats.

Moreover, we do not focus on how to build a reputation table once behavior has been observed and evaluated. It is known that making decisions from a variety of vague and possibly contradictory information is difficult, and it is a field of ongoing research how to best approach such a situation. For now, we consider a simple Bayesian approach to decision making. It is important to clarify that the evaluation and decision making is not the focus of our paper. In contrast, our contribution is the description of the attacks, their classification, and the introduction of a general design technique: reputation based control. We suggest that this technique is useful in suppressing the most aggressive attacks, and without opening up to other strong attacks.

III. Model

A. Network Graph

We represent an ad-hoc network by an undirected graph. Vertices of the graph correspond to ad-hoc nodes with Radio Frequency (RF) transceivers. There is an edge (link) between two nodes if they are within each other’s transmission range. Thus, we make a simplifying assumption regarding the symmetry of the ability to receive a signal, but note that our results hold regardless of whether this assumption is made or not.

Our results hold both for static and dynamic network topologies, but we concentrate on networks with mobility (as indicated by the title) due to that the severity of our attacks increases for these. Namely, in situations when the network topology changes dynamically, this provides an advantage to the attacker for several reasons.

First, as the mobility increases, the distinction between locally and remotely mounted attacks disappears, allowing an attacker to use some attacks that require some degree of presence, but without the drawbacks (i.e., traceability) that normally come with these. More in particular, mobility allows a modification of the routing table of a selected node, simply by moving into the transmission range of the victim. The attacker can move away once it succeeds, and without the threat of being traced *(increasing the stealth property)*.

Second, the mobility of honest nodes can help to disperse the information that the attacker aims to advertise (*epidemic property*). Both of these problematic aspects can be modeled as propagation of outdated routing information (as pointed out in [19]) and therefore be countered in settings with enough redundancy. We consider possible techniques for doing this later on.

Third, the set of nodes within the transmission range of a node keeps changing dynamically in mobile networks. As the mobility increases, it becomes harder (and potentially more expensive) to successfully employ cryptographic techniques (such as authentication).

Finally, typical mobile nodes are less powerful – both in terms of computational resources and in terms of battery resources – than the typical stationary node, and typically do not enjoy the benefits of easy access to a trusted third party such as a Certification Authority.

Participants

We consider two types of participants; *honest* participants and *cheaters*. For simplicity, we assume that all nodes that do not behave correctly are cheaters, including nodes suffering benevolent failures. For simplicity, we also make the pessimistic assumption that all cheaters are controlled by an adversary, who coordinates their actions[1]. These may diverge from the prescribed protocol in an arbitrary and unpredictable manner. We may assume that the adversary is able to coordinate the actions of all cheaters by means of out-of-band communication. Again, this is a pessimistic assumption.

We let participants belong to one out of two power classes; those with "inexhaustible" power (e.g., not battery powered) and those with a limited power budget. Among the latter, we distinguish between three different modes of operation, corresponding to three different levels of power. These are *charged*, *reduced* and *dead*. It is possible to assume more such categories, and tailor the behavior of participants in these accordingly, but, for simplicity, we only use three. In the charged mode, a participant is assumed to act in an altruistic manner; in reduced mode in an egotistic manner, and when in "dead" mode, not at all. With altruistic, we mean that the router is in *promiscuous mode* and performs any properly performed requests relating to routing, while egotistic is used to refer to a mode in which a router only performs those actions that directly benefits its transmission of packets.

In terms of computational power, we assume participants either to be *normal* or *weak*, where "normal" corresponds to the computational abilities of typical desktop computers, etc., and "weak" relates to typical wireless devices. Typically, devices with inexhaustible power would have normal computational abilities, while devices with limited power budget are computationally weak.

IV. Operation

In the following, we describe the link layer and routing protocols. In Section VI, we will describe the attacks with respect to what selections are made for these protocols, and exemplify for common combinations of these.

[1]This is in accordance with standard cryptographic modeling techniques, and does not directly correspond to the use of any of the proposed weapons. Practically speaking, corruption of a set of participants may be achieved by means of viruses, insecure software, physically compromising of honest routers, other infiltration, or plainly by agreement between entities belonging to an adversarial organization. Therefore, this also covers the creation of nodes by the attacker, as this is equivalent to the corruption of newly created honest nodes.

A. Link Layer

For concreteness, we assume that the link layer protocol follows the IEEE 802.11 standard (see, e.g., [10]). Two modes of operation are considered: (i) priority based, contention free Point Coordination Protocol (PCP), and (ii) Distributed Coordination Protocol (DCP) which is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In CSMA/CA a node listens to the medium until the medium is idle; then it transmits. If there is a collision, the node will hear a different signal in the medium than what it was transmitting, and concludes that the transmission is in collision. Collided stations backup exponentially on the number of unsuccessful attempts to capture the channel. Communication between two stations is based on a 2-way handshake: after authentication, the sender first transmits a Request to Send (RTS) message, and the receiving station replies with a Clear to Send (CTS) message. The sender then transmits data and awaits an Acknowledge (ACK). It is worth noticing that all the management (control) messages are transmitted in the clear in the current specifications of IEEE 802.11. In the following, we limit the focus on security vulnerabilities relating to routing issues, and refer the reader to [7] for a discussion of other security concerns relating to this standard.

B. Network Layer

In the network layer, we assume that one of several available ad-hoc routing algorithms is deployed. We will consider both *proactive* routing and *reactive* routing protocols. In the former, nodes maintain a connectivity graph by exchanging routing tables regardless of whether there is demand for routing to every entity in the table. In the latter, routing information is obtained when there is a demand to send traffic to a particular destination. A node updates its routing table only after performing a route request (RREQ) and obtaining a response.

In particular, we consider the employment of Dynamic Source Routing (DSR) [1], Ad-Hoc On-Demand Distance Vector Routing (AODV) [14], Zone Routing Protocols (ZRP) [4], and TORA [13]. DSR and AODV are reactive protocols. The former uses route caches while the latter maintains routing tables and uses Distance Vector Routing algorithm to compute the routes. ZRP is a hybrid routing protocol that uses a hierarchical structure for routing. TORA is also a reactive protocol, and is based on techniques in [3]. The attacks considered in this work are relevant to all these protocols.

C. Proactive Routing

Proactive routing protocols maintain routing tables. When a message is sent using proactive routing, the packet carries only information relating to its origin and desired destination. Each node has a routing table to indicate what the next hop is for that particular destination. Nodes in proactive routing exchange routing tables periodically – either with neighbors only, or by flooding the entire network. This way, each node can infer the network graph and compute the routes. There are two types of protocols suggested for proactive routing. In

link-state protocols and its variants, each node floods its local connectivity (i.e., list of its neighbors and distances to them) to the entire network. (Thus, each node knows the (claimed) topology of the entire network and uses Dijkstra's shortest path algorithm to compute the routes.) In contrast, distance-vector protocols and its variants exchange the global topology information that is maintained only with immediate neighbors. Such algorithms are known to be prone to loops and slow convergence. If the topology of the graph changes during the transmission of a packet (e.g., a link or node goes down), the transient packet will be dropped. Control messages are propagated periodically, or whenever there is a link failure. Like all network operations, these are asynchronous.

A link failure is recorded locally to the routing table of the node that detects it. The link failure information is propagated to the network by routing table updates using link-state or distance vector protocols to prevent routing errors. However, an attacker can frequently report link failures to mount additional overflow attacks. Furthermore, such links may not even be real links.

D. Reactive Routing

While proactive routing protocols maintain routing or connectivity information to a node regardless of whether any packet will ever be sent to that node, in reactive protocols, a route is determined only if there is a packet to be sent. Route discovery information is then stored locally, but may not be communicated to others unless requested.

In order to limit the flooding of the network with route requests, and to speed up the route discovery process, some reactive protocols construct and maintain *route caches* or *route tables*. (For example, AODV uses local routing tables, while DSR with improvements applies routing caches.)

In contrast to routing tables, which only store the next hop (and distance metric) information, a route cache stores the *entire* route from a source to destination. There is no periodic exchange of route caches: each node "learns" the routing information from the route discovery process.

When a message is sent using a source routing protocol (e.g., DSR), the packet carries full routing information, i.e., the sequence of all nodes the packet will traverse. In contrast, in distance vector routing protocols (e.g., AODV), the packets carry only information about their origin and destination. If the graph topology changes during the transmission of a packet, the route will become invalid and transient packets will be lost.

Upon receiving a route request message, a node checks its local route information to see if any previously found route for the destination exists. In case of several possibilities, one of them is chosen using a heuristic rule, such as the shortest one, or the shortest one with longest expected lifetime [5].

Large route caches and route tables may contain stale routing information, and so, are often avoided. Due to the size limitation, only the most recent or active routes are maintained. However, small caches or tables can more easily be exploited by an attacker that overflows them with incorrect (i.e., non-existing) routes to replace the correct ones to the victim.

V. Building Blocks for Attacks

Before we describe the attacks, we will describe common building blocks used in these. These building blocks use the weapons outlined in the introduction, and are, in turn, used in the attacks. The building blocks we have in mind allow the adversary to *remove* entries from respectively *add* entries to routing tables, route caches or any data structure containing routing information, originate collisions, i.e., violate the collision avoidance protocol and consume power to change the operational mode of devices.

The first two of these building blocks can be implemented in two principal ways: one in which the adversary never exposes his identity through requests and responses (but employs impersonation), and one in which the adversary initiates some requests and responses. We describe both variations below in the context of the different types of routing protocols.

β_1: Removing an entry using impersonation

Impacts of removing an entry are twofold. First it prevents a victim from *receiving* traffic from honest nodes. Second, it prevents a victim from *sending* traffic to honest nodes. It can be achieved by portraying a receiver as unreachable or down node to a sender.

In proactive protocols, the attacker $\mathcal{A}$ takes advantage of routing updates. It can generate malicious routing tables and advertise them during periodic updates using the identity of a honest node. An attacker can simply impersonate a neighbor N of the victim V and claim that the link to the victim is down. In the case that some other nodes have a route to the victim, $\mathcal{A}$ creates a fictionary node that claims to have the shortest distance to the victim. Once the routing tables of the honest nodes remove the old route to the victim and mark their route to the victim via the fictionary node, the attack will have succeeded.

In a reactive protocol, on the other hand, finding a route to the destination involves flooding the network with control messages of a route discovery protocol. Since there is no routing information exchange, attacks must aim at the route discovery process. An attacker $\mathcal{A}$ can generate *route error* messages which will be interpreted by honest nodes that there is no route found to the indicated destination. This can be done by impersonating two nodes such that the first one makes a route discovery request while the second one – which is on the path of the request – replies with a "route error" message. Thus, nodes located between the two colluding nodes will believe the requested node to be unreachable, and remove the corresponding route from their caches. However, if some honest nodes know that the route exists and report that, this attack will not work. Thus, knowing the topology and its stability (i.e., low mobility) helps the attacker to identify the connectivity and to decide on the feasible set of nodes to target.

β_2: Removing an entry without impersonation

In reactive protocols, the attacker $\mathcal{A}$ can simply force the dropping of the route discovery messages to create the false impression that no route is found to the destination (victim). For example, suppose an honest node N_1 has data for destination D but it does not have the routing information to reach it. Therefore, N_1 generates a route discovery message and floods the network. Consider a valid route $N_1-N_2-\ldots-N_k-V$ that can be reported back to N_1 after the route discovery message reaches to V. Suppose a node N_i, controlled by the adversary, intercepts the route discovery message, and causes it to be dropped. This can be done either before the control message reaches V, or before V's response reaches N_1. Thus, the nodes in the path $N_1 - N_2 - \cdots - N_{i-1}$ will not discover the route.

In proactive protocols, $\mathcal{A}$ simply attacks the routing table update process. As before, the attacker can either distribute routing tables omitting certain entries or create new nodes (that he controls), where the latter claim to have the shortest path to the victim node thus causing currently used paths to reach the victim node to be dropped.

β_3: Adding an entry using impersonation

In proactive protocols, the attacker $\mathcal{A}$ propagates routing tables containing non-existing routes. The attacker $\mathcal{A}$ can initiate this attack remotely from anywhere in the network using the name of an honest node as identifying information for the sender.

In reactive protocols the attacker $\mathcal{A}$ can corrupt the routing information of victim devices by introducing or modifying received route reply messages using fictionary node IDs. In order to succeed, the attacker has to impersonate two nodes such that the first one makes a route discovery request while the second one – which again is on the path of the request – replies with either with faked or modified routes.

β_4: Adding an entry without impersonation

Adding an entry without impersonation is possible if the attacker $\mathcal{A}$ uses its real identity to abuse the routing protocol. In proactive protocols, $\mathcal{A}$ simply propagates routing tables that contain routes that do not exist, thus causing honest nodes to change their routes. The attacker $\mathcal{A}$ can initiate this attack remotely from anywhere in the network since all it needs to do is to inject the malicious table to the networks.

In reactive protocols the attacker $\mathcal{A}$ can create a non-existent route to an honest node during route discovery process. It can either fabricate a route reply message to report a non-existent route, or it can intercept and modify a route reply message coming from an honest node.

β_5: Jamming

Since there is no handshake for reliable delivery in flooding, an attacker can simply violate the CSMA/CA protocol. It can generate traffic to collide with control messages, e.g., for route discovery or periodic exchange of routing information.

β_6: Consuming power to change operation mode

Nodes is an ad-hoc network must be in *promiscuous mode* (ready to receive any transmission) to be able to route other traffic. As the power level decreases, a node may not be able to afford acting as a router and may be switched off from the promiscuous mode. However, in this case the device will not contribute to the routing protocols thus will be assumed down and isolated. Thus, our last building block is to force power consumption on a victim (or the entire network) with the goal of partitioning the network.

For example, in proactive protocols, an attacker $\mathcal{A}$ can report a link failure and thus cause a costly routing table update process. For this attack to work, the link in question does not need to exist and $\mathcal{A}$ can claim the failure remotely using broadcasting. Furthermore, $\mathcal{A}$ can impersonate and advertise false routes to increase the volume of traffic to a victim. Transmission of each packet consumes battery power of the victim node [17], [18], [2]. For example, radio transmission consumes 1.6W; reception requires 1.2W and 1W is consumed for idle listening.

In reactive protocols, the route discovery process is based on flooding, which has similar cost to the route updates in the proactive case. Thus, $\mathcal{A}$ can broadcast a route request from a remote location to a non-existing destination and force power consumption. By changing the destination address frequently it can avoid route caches. Furthermore it can hide its identity to avoid being detected for issuing frequent route requests. $\mathcal{A}$ can also change the route reply messages to increase volume of the traffic forwarded to the victim.

Moreover, there is another attack that exploits the well known counting-to-infinity problem in distance-vector based algorithms. Consider three nodes N_1, N_2, N_3 such that N_1 and N_3 are within the transmission range of N_2, but they cannot hear each other. N_3 is a *fictionary node* created by $\mathcal{A}$ using impersonation. Suppose N_2 reports its distance to N_3 as 1, and that N_1 makes a note of it in its routing table. When the routing tables are exchanged, N_2 learns that the distance between N_1 and N_3 is 2. Shortly afterwards, N_3 *disappears*, emulating a failure and becomes unreachable from N_2. $\mathcal{A}$ can do that just by moving away or staying quiet. However, it remains in the routing tables of victims N_1 and N_2. So N_2 sets its route to N_3 via N_1 and sets the distance to 3. In the next round of routing table exchanges, N_1 finds out that its distance to N_3 via N_1 is no longer 2 and it sets it to 4. This process can continue until an upper bound (set as a remedy for counting-to-infinity problem in a non-malicious operation) on the distance to N_3 is violated. This technique can also be used by $\mathcal{A}$ for perpetrating power attacks.

VI. Attacks

There are several ways to target one or more nodes using the above introduced building blocks. We will consider these one by one in the following.

A. Disconnection and Goodput Reduction

An attacker may disconnect a victim in several ways. The first three ways we will describe have in common that the attacker causes a large number of packets to be sent to the victim and its neighbors. This can be done either by "brute force", i.e., by simply sending these packets, or by what we refer to as the "stealth DoS", in which the attacker causes large amounts of traffic to be rerouted by inducing incorrect entries in routing tables of selected nodes. This may be done – as described in the previous section – by first removing the chosen entry (using the building block β_1 or β_2) and then adding a replacement entry by means of β_3 or β_4.

First, the attacker may route such considerable amounts of traffic through the victim that the victim either runs out of power, since each packet received or sent carries a cost in terms of the battery power consumed. The discussion in [17], [18], [2] on the exact amount of power consumptions support that this is a real threat for standard portable power sources.

Second, the attacker may perform a power attack on all known neighbors of the victim node. This will cause disconnection as well, but may be overcome by the victim by him moving into another neighborhood.

Third, an attacker may succeed in disconnecting a victim from its neighbors without performing a power attack. Namely, if the attacker could route large enough quantities of traffic to the victim and its neighbors, causing a portion of these to be dropped (due to insufficient bandwidth), then this could result in a disconnection. This is so since when a router fails in reaching a given node a certain number of times (which is often a parameter of the protocol), the router concludes that the recipient is unreachable.

For both reactive and proactive protocols, the attacker succeeds in disconnecting the victim nodes by making other nodes believe that the former are unreachable (and thus actually *making* them unreachable.)

In a *fourth* type of attack, the attacker does not rely on large quantities of packets being sent to the victim or its neighbors, but simply uses the weapon for removing an entry (building block β_1 or β_2) to make the victim node "disappear".

In the case of proactive protocols there could be multiple nodes which know how to reach the victim thus making the attack more difficult. However, if $\mathcal{A}$ knows the topology, it can compute the routes and jam the traffic to create a link failure on the route to the victim. Once a link failure is reported (falsely) then the routing table update process will be initiated and then $\mathcal{A}$ can use the building blocks to isolate the victim.

In reactive protocols, there is no periodic exchange of routing information and network topology information is not maintained. Routing information for any destination is confined to a subset of the nodes and it may be even localized. The lack of global information at the ad-hoc nodes helps $\mathcal{A}$ to target only selected routers and remove the victim from the routing tables of these nodes only. $\mathcal{A}$ can attack the reactive protocols by jamming (β_5), intercepting, or forging the route discovery messages to convince the source node and other honest nodes that no route to the victim can be found.

Receiving a packet from an unreachable node does not yield any routing information unless the packet *carries* some routing information (e.g., source routing). Moreover, in reactive protocols, if such a disconnected node were to send a packet to one of its neighbors, only that neighbor would know that the victim is reachable. This information will not be advertised to the rest of the network and can therefore be learnt only by a neighbor of the victim who is involved in the route discovery process associated with the attack.

Goodput Reduction: We note that disconnecting one or more nodes generally implies a reduction of the goodput of a network. An attacker may mount the attack in several ways. In particular, by disconnecting a large number of nodes, the resulting traffic through the articulation points comes to a crawl; the attacker can corrupt a large enough number of routing tables to increase the de facto traffic through each node (by taking a large number of packets for a ride); and he can degrade the power supplies of a large enough portion of the routers (building block β_6) to force them switch to "egotistic" routing, i.e., only handle their own packets. We note that this may then result in a total disconnection or partition of the network.

B. Active Eavesdropping

A second class of attacks aims to "hi-jack" traffic in order to eavesdrop on selected victim nodes. The simplest way to achieve this is to corrupt routing tables of nodes on the path between a victim and the respective sender/receiver. The attacker can remove valid routing table entries and add incorrect ones in order to force rerouting. This can be achieved using the previously introduced building blocks β_1, β_2, β_3, respectively β_4.

For incoming traffic (i.e., packets going *to* the victim), the attacker simply forces all traffic to the victim to be sent through a node he has corrupted. In order to select traffic only from certain sources, the attacker may corrupt the routing tables more selectively, allowing those on the path from "not so interesting" sources to remain correct.

For outgoing traffic (i.e., packets sent *from* the victim to another node in the network), the attacker modifies routing tables of the victim and/or nodes close to the victim (with respect to all "interesting" recipients) thereby causing traffic to be rerouted through a node he controls.

The main difference between proactive and reactive protocols with respect to active eavesdropping is again on how the routing information is tampered and how rerouting is achieved. In proactive protocols, the attacker can simply propagate respective routing tables in which entries are dropped or added. In reactive protocols, the attacker will make use of the route discovery process to advertise new routes or report route error messages.

We note that rerouting not only affects traffic from the victim and to the selected receivers, but *everybody* sending/receiving packets through any of the routers whose tables are corrupted. The resulting traffic through the eavesdropping node can be reduced by averting all traffic from the corrupted

routers, except for traffic from the victim of the eavesdropping attack.

Simpler still, in a protocol such as DSR, where each packet carries its path (if known by the sender) the attacker may plainly modify the routing tables of its victim to make it route traffic to select receivers through a node the attacker controls.

VII. Prevention Mechanisms

From the previous descriptions, we can see that all the attacks rely on the adversary being able to modify routing tables and caches of victim routers. This in turn relies on (a) removing unwanted entries, (b) adding wanted entries, and (c) knowing the connectivity and other properties of the network topology to choose its collaborators. If these were not possible, the attacks we described would fail.

The most threatening of our techniques (i.e., those with the least visibility and cost) are those that employ impersonation mechanisms. Therefore, the use of cryptographic authentication methods would improve the resistance against stealth attacks, since cryptographic authentication cannot be forged, as IP addresses, etc., can. However, even if we do not consider the negative ramifications of a full-blown authentication structure (namely the cost of performing the cryptographic operations and maintaining the necessary infrastructure), it is clear that the mere reliance on authentication is insufficient. This is so since we must make the assumption that nodes that previously have ”well-behaved” later become compromised, and so, correct authentication of control messages does not correspond to correctness of the control information. This difficulty is enhanced by the fact that it is not common knowledge among the honest servers who exactly is an honest server – whether this set is static or not.

Our proposed technique to deal with the above dilemma is for each router to keep (and possibly exchange) reputation based information. Routers can then use this to resolve conflicting updating information, and to determine what control messages to handle and act on. While this leads to a certain reduction of the speed with which routers will react to *real* topology changes, it will also reduce the degree to which they are affected by attacks.

The idea of *reputation based control* is simple, and draws from the real world: Each person shapes an opinion of the trustworthiness of all entities surrounding her: co-workers, merchants, media, stock brokers, etc. This information is used when making decisions; similarly, routers may keep ”reputation tables” or ”reputation caches” that list nodes they trust. (It is not so meaningful to list those that are not trusted, since these may change their guise by taking new identities.) While *personal* experience is most valuable when making real-world decisions, people often consider other people’s opinions. To this extent, a router may request reputation tables when moving into a new network neighborhood. (We note that this will be a bad move if it receives such information from a corrupt router; this will be dealt with below.)

In order to determine the reliability of a particular router, a variety of methods can be used, alone or in conjunction with each other. Two useful heuristics are the average number of retransmissions (per packet) to a given router; and the number of successful exchanges of data involving a given router (whether on the path to another router, or the end-point of the path in reactive protocols.)

In reactive protocols, the nodes involved in reporting a route discovery are accountable for the performance of this route. In proactive protocols, one can maintain a similar “source-of-information” attribute with each routing table entry learned from others during routing table exchange phase. For example if node A receives a routing table update information from B which learned from C a new route to node X then A will hold B and C “accountable” for the routing failures to node X.

Similarly one can judge the reliability of a ”recommending router” (one that shared reputation data) by the reliability of the routers it recommended. (If these are bad, so is the recommending router with a reasonable probability, or it would not have recommended them.)

In conjunction with reputation mechanisms, one also has to employ cryptographic authentication mechanisms. For these not to open up the protocol to overloading attacks, they should be as lightweight as possible. Therefore, we suggest the use of Message Authentication Codes instead of digital signatures. We note that this is possible since the goal of using the mechanisms is not to establish *exactly who* originated some information, but merely to be able to recognize the same entity in consecutive interactions.

For example, message authentication codes can be employed to protect the integrity of routing tables (in proactive protocols) and route reply messages (in reactive protocols). As long as none of the already trusted nodes become corrupted, this is sufficient to obtain security. However, corruptions require a dynamic treatment of the situation, as does mobility, as both force ”new neighbors” onto nodes.

Thus, there is no need for any certification of identities, but one could simply use message authentication codes, for which each pair of entities use a previously exchanged unique and random secret key. (This value is exchanged upon the first encounter, and can be done in a variety of well-known ways, with varying cost and security against attacks.) As an alternative, message authentication codes with a keying schedule based on hash-chains (as proposed by Perrig et al. [15]) may be employed, allowing authenticated broadcasts.

We refer the reader to articles discussing authentication techniques for a discussion of their exact pros and cons.

Our techniques for reputation based control can be employed to both existing routing protocols and taken into consideration in the design of new ones. However, it is important to note that reputation based control is not a panacea, as an attacker may try to soil the reputation of honest routers by causing packets to be dropped when sent on routes recommended by his ”slander victims”. On the other hand, this is an attack that is more difficult to perpetrate remotely. This is so since – in the setting we consider – there are only two ways in which an attacker can cause packets to be dropped. If the attacker is not on the path, he needs to cause

a router on the path to drop packets. If there is no way to route massive amounts of traffic to this victim router (which is what our prevention measures aims to achieve), then the only remaining method is to remove entries from the victim's routing table. (This only applies to proactive methods, since the path typically is carried with the data for reactive schemes.) By implementing a certain inertia in terms of when entries are dropped from routing tables, the "erasure attack" can be made harder.

The exact degree of increased inertia to real topology changes depends on the degree of interconnectedness of the "trust web" that corresponds to the distributedly held reputation tables. The strengthened routing protocol will be robust to benevolent errors (i.e., adapt to valid changes) if a sufficient degree is achieved. The routing protocol will be robust to malicious errors (i.e., resistant against attacks) given a sufficient degree of interconnectedness as well; if most routers are within the transmission range of some honest and trusted router most of the time, the protocol will be resistant to attacks. However, as mentioned above, this proposed technique is heuristic, and is susceptible to "slander attacks". Given an appropriate tuning of the mechanism for determining the reputation of known routers, the two types of robustness can be balanced in accordance to the current threat situation. An important and difficult future research problem is to assess how exactly to achieve and maintain the appropriate balance.

VIII. Conclusion

In this paper we have studied routing attacks on ad-hoc networks. By introducing the notion of reputations based control we have furthermore described new techniques for protecting routing information thus providing the foundation for secure protocol design. One key observation of the paper is the insight that protection against one type of attacks weakens the network against a second type of attacks. Finding the right right balance is extremely difficult. While the paper presents some initial heuristic approaches finding an optimal solution is a matter of future research. Ongoing work includes the implementation of the various attacks based on AODV.

Acknowledgments

The authors would like to thank Levente Buttyan and Nicolas Girard for their valuable comments and helpful suggestions on earlier drafts of this paper.

References

[1] J. Broch and D. B. Johnson and D. A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks. *Internet Draft, draft-ietf-manet-dsr-03.txt*, 1999.

[2] L. M. Feeny and M. Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad-Hoc Networking Environment. *Proceedings of IEEE INFOCOM*, 2001.

[3] E. Gafni and D. Bertsekas. Distributed Algorithms for Generating Loop-Free Routes in Networks with Frequently Changing Topology. *IEEE Transactions on Communications*, 1981.

[4] Z. J. Haas and M. R. Perlman. The Performance of Query Control Schemes for the Zone Routing Protocol. *IEEE/ACM Transactions on Networking*, 2001.

[5] Y-C Hu and D. B. Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad-Hoc Networks. *Proceedings of ACM Mobicom*, 2000.

[6] Y-C Hu, A. Perrig and D. B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. *Proceedings of IEEE INFOCOM*, 2003.

[7] Internet Security, Applications, Authentication and Cryptography Research Group. Security of the WEP Algorithm. `http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html`, 2001.

[8] M. Jakobsson and F. Menczer. Untraceable Email Cluster Bombs: On Agent-Based Distributed Denial of Service. `http://www.markus-jakobsson.com`, 2003.

[9] V. Kärpijoki. Signalling and Routing Security in Mobile and Ad-hoc Networks. `http://www.hut.fi/~vkarpijo/iwork00/`, 2000.

[10] LAN MAN Standards Committee of the IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHYY) Specification. *IEEE Standard 802.11*, 1999.

[11] J. Lundberg. Routing Security in Ad-Hoc Networks. `http://www.tml.hut.fi/~jlu`, 2000.

[12] R. Ostrovsky and M. Yung. How to Withstand Mobile Virus Attacks. *Proceedings of ACM Symposium on Principles of Distributed Computing*, 1991.

[13] V. D. Park and M. S. Carson. A Highly Distributed Routing Algorithm for Mobile Wireless Networks. *Proceedings of IEEE INFOCOM*, 1997.

[14] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[15] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes* `http://www.rsasecurity.com/rsalabs/cryptobytes/` , 2002.

[16] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks. *Proceedings of International Workshop on Security Protocols*, 1999.

[17] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, 1997.

[18] Y. Xu, J. Heidemann and D. Estrin. Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks, Technical Report 527, USC/ISI, Los Angeles, CA, `http://www.isi.edu/ johnh/PAPERS/Xu00a.pdf`, 2000.

[19] L. Zhou and Z. J. Haas. Securing Ad-Hoc Networks. `http://www.ee.cornell.edu/~haas/Publications/network99.ps`, 1999.