Malicious attacks on ad hoc network routing protocols

PO-WAH YAU, SHENGLAN HU and CHRIS J. MITCHELL
Information Security Group,
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{P.Yau, S.Hu, C.Mitchell}@rhul.ac.uk

Abstract

The main purpose of an ad hoc network routing protocol is to enable the transport of data packets from one point to another. This paper examines the potential attacks on this transport service which arise from the realisation of threats from internal malicious nodes. The prerequisite of a routing service is a distributed mechanism for the discovery and maintenance of routes; network integrity and availability are required to ensure the correct operation of an ad hoc network. This paper also provides a qualitative analysis of how proactive and reactive protocols cope with malicious internal attacks, and whether one type of protocol offers inherently better resistance to the various attacks than the other.

Keywords: Routing protocols, network security, mobile networks.

1. Introduction

Ad hoc routing protocols can be classified as either proactive or reactive [2, 4, 38, 65], depending on the method used to discover and maintain routes. Proactive routing protocols discover and maintain a complete set of routes for the lifetime of the network. In contrast, reactive routing protocols only find routes when needed, and maintain those routes for the duration of communication. Examples of proactive routing protocols for ad hoc networks include Destination Sequenced Distance Vector (DSDV) [40, 41], Optimised Link State Routing (OLSR) [10, 25], and Topology Based dissemination based on Reverse Path Forwarding (TBRPF) [35]. Examples of reactive routing protocols include Ad hoc On-demand Distance Vector (AODV) [36, 37, 42], and Dynamic Source Routing (DSR) [26, 27].

This paper presents a qualitative analysis of how the characteristics of reactive and proactive protocols affect the seriousness of attacks by malicious nodes, i.e. a category of misbehaving internal nodes which have the necessary information to participate in distributed operations as trusted principals [60, 61, 62]. Attacks posed by internal nodes, a concept introduced by Zhou and Haas [64], are difficult to detect because they arise from trusted sources.

Malicious nodes aim to deliberately disrupt the correct operation of the routing protocol, denying network services if possible. Such nodes can use or modify sensitive routing information. For example, the distance metrics used in conventional protocols, such as the number of hops, do not reveal a destination's location. This is mainly because of the arbitrary distances between routers participating in the routing protocol. However, with knowledge of the wireless range of each node, it is feasible to use routing information to calculate an upper bound on the physical distance between neighbouring nodes, and hence infer information about the geographical location of a node. This may help malicious nodes to target their attacks.

Both data packets and control packets, as used by the routing protocol, are vulnerable to attacks. In this paper, we consider whether proactive or reactive routing protocols are inherently more secure against attacks by nodes exhibiting malicious behaviour.

Section 2 discusses the relevant literature. Section 3 begins the analysis by discussing the data packet transport service, and how this is vulnerable to threats arising from malicious behaviour. Section 4 looks at attacks on the route discovery and maintenance mechanisms. An analysis comparing the inherent robustness of proactive and reactive protocols is given in section 5. Finally, section 6 contains conclusions for the paper.

2. Related work

Independent work has been carried out by Wang and Bhargava [54], who compare the properties and resulting vulnerabilities of the Ad hoc On-demand Distance Vector (AODV) [36, 37, 42] and Destination Sequenced Distance Vector (DSDV) [40, 41] protocols. Wang and Bhargava's main conclusion is that attacking the sequence number mechanisms by advertising a falsely high value has a greater impact than attacking the distance vector by advertising a falsely low value. They provide simulation results together with an analysis, concluding that it is easier to detect attacks on the destination sequence number in proactive protocols than reactive protocols.

In this paper, we look at specific mechanisms rather than concentrating on complete protocols. Thus, the research is intended to be applicable to any ad hoc network routing protocol. Also, when discussing proactive routing protocols, we consider only link-state routing protocols. Both AODV and DSDV are distance vector protocols, and proactive distance vector protocols, such as DSDV, are no longer being considered for standardisation by the Internet Engineering Taskforce (IETF) because of their poor performance [39]. Hence it is no longer particularly helpful to consider DSDV when making a comparison between reactive and proactive routing protocols.

3. Attacking the routing of data packets

Before examining attacks on route discovery and maintenance procedures, which essentially means attacks involving the protocol control packets, we examine attacks on the routing of data packets. In the attacks we consider, the methods and results are the same for proactive and reactive protocols. The attacks we identify are:

- Denial of service,
- Modifying the packet header,
- Flooding attacks, and
- Replaying and reordering data packets.

Denial of service attacks include deliberately dropping packets instead of forwarding them, as well as actively interfering in the communication of neighbouring nodes. The latter can be achieved using internal knowledge to attack the medium access control protocol (see section 3.2). The effects of denial of service are similar to those arising as a result of selfish behaviour; we do not consider such behaviour in this paper, and we refer the reader to [3, 5, 6, 12, 58, 59, 60, 61, 62] for more on this topic.

In the remainder of this section we explore the other three types of attack. Before proceeding, note that we have ignored the threat of possible modification of the data packet payload. This is because security mechanisms in the upper layers of the protocol hierarchy will be better placed to provide protection to the payload.

3.1 Modifying the packet header

Ad hoc routing protocols typically use the Transport Control Protocol (TCP) [11] or the User Datagram Protocol (UDP) [44] as the transport layer protocol, both of which use the Internet Protocol (IP) [45]. Hence, every packet

will contain an IP header, containing, amongst other data, the source address, destination address and Time-To-Live (TTL) value.

Malicious nodes could modify the destination address of a data packet to reroute it. This could be targeted at a specific node, in order to cause a denial of service attack against that node, or at the intermediate nodes which will forward the modified packets. Modifying the source address would disturb a TCP connection, causing additional packets to be sent to recover the connection.

The inclusion of the TTL value is also a potential vulnerability. An intermediate node will not forward packets if the TTL value in a received packet is 1 or less. Since the TTL mechanism is not protected, a malicious node could reduce the TTL in received packets to an artificially low value. As a result the packet may not reach its destination, since a downstream intermediate node will drop the packet. This is also a means to achieve a gratuitous detour attack (see section 4.3), as traffic may be directed away from the route involving the malicious node, as it is believed to be unusable. Thus, this attack achieves the same results as dropping packets, but it allows the malicious node to avoid being identified as the source of packet dropping. However, it is generally easier to protect the integrity of packets than to detect packet dropping.

One of the reasons for the use of the TTL value is to prevent packets from propagating indefinitely around routing loops. Hence, conversely, a malicious node could increase the TTL in received packets to a much larger value. The degree of denial of service achieved by this attack depends on how effective the routing protocol is at preventing, or reducing, the frequency of routing loops.

If a source route based routing protocol, such as the Dynamic Source Routing (DSR) protocol [26, 27], is being used, then a malicious intermediate node can alter the source route of any packets it receives. This could be used to misdirect a data packet along a different route to that intended by the packet originator.

3.2 Flooding attacks

As with conventional networks, the threat of a flooding attack applies. Such an attack is difficult to distinguish from a sudden but legitimate increase in network traffic [8]. A malicious node could attempt to flood the network with its own unicast data packets, potentially using many different destination addresses. Gupta, Krishnamurthy and Faloutsos [20] show how two colluding nodes, at opposite edges of a network, can partition the network by sending a high volume of data between them. While the authors focus on capturing the wireless medium by exploiting the vulnerabilities of IEEE 802.11, the attack also causes denial of service by exhausting the intermediate nodes which forward the high volume of traffic generated. Thus, one attack can simultaneously achieve more than one type of denial of service.

A related but more localised attack arises when a malicious node sends its neighbours packets to forward at a rate at which the neighbours become overwhelmed. The pacing protocol is a mechanism used by the DARPA Packet Radio Network (PRNET) [18, 28], in which nodes measure the forwarding delay of their neighbours in order to pace the rate at which to send packets to their neighbours for forwarding. Thus, a malicious node could deliberately not follow the pacing protocol and engulf its neighbours with packets. Gupta, Krishnamurthy and Faloutsos [20] describe the effects of such an attack on the link layer. Maliciously sending a high volume of packets not only prevents immediate neighbours from accessing the wireless medium, but also deprives nodes in the 2-hop neighbourhood of the malicious node of network connectivity. This exploits the 'capture effect' [19], whereby a node with a heavy traffic load will capture the wireless medium and prevent a node with a lighter traffic load from accessing the medium.

When using routing protocols which can concurrently route data along multiple paths, a flooding attack can affect an even greater proportion of the network. Finally, note that the payload of each packet does not necessarily have

to contain any meaningful information; the attacker only has to ensure that the packet headers contain the correct information.

3.3 Replaying and reordering packets

Replaying packets in an ad hoc network is an attack that differs from replay attacks in conventional wired networks, in terms of both time and space. Malicious nodes can move to different areas of the network to replay data packets. A malicious node could move as far away as possible from the destination node before replaying packets in order to involve more intermediate nodes, and to deplete their resources while forwarding the packets.

Replay attacks are generally prevented using some form of freshness mechanism. Atypical example is the use of sequence numbers, also known as logical timestamps. However, routing protocols in the network layer generally do not use any freshness mechanisms to prevent the replay of data packets. While the IP header includes a sequence number, it is only used to reconstruct a packet which has been fragmented, so it cannot be relied upon to identify unique packets. Sequence numbers are primarily used by TCP to maintain the order of packets sent via a connection. Although TCP sequence numbers could be used to ensure freshness, this is not advisable because:

- This is only possible when TCP is being used as the transport control protocol;
- This introduces a cross-layer mechanism, which defeats the purpose of compartmentalising network functions into different layers; and
- Any confidentiality service in the transport layer may hide TCP sequence numbers, thereby preventing them from being used by the network layer.
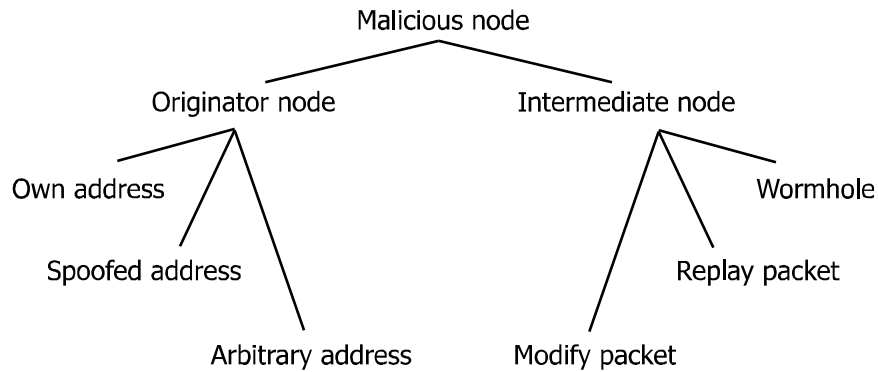
Sequence numbers are used in the network layer for control packets, and we will examine their use in section 4.2.

A related attack arises when there is a TCP connection between two nodes, and a malicious intermediate node reorders packets sent between the two communicating nodes [1]. This can be done to such an extent that it breaks the TCP connection, therefore causing denial of service. Note that, if protection from replay attacks is needed in upper layer applications or services, then they must also implement freshness mechanisms, located in the layer in which protection is required.

4. Attacking the routing protocol

Two possible threats from malicious nodes are misdirection of traffic, one of the consequences of which may be denial of service, or denial of service as a means to an end itself. These threats can be further subdivided, as in the attack model shown in Figure 1.

Attacks arising from malicious behaviour can be divided into those where packets are originated by the malicious node, and those where a malicious node is an intermediate node and receives control packets for for-warding. When a malicious node is originating packets, it can send control packets using its own source address, an address which belongs to an existing node in the ad hoc network, or an arbitrary address which does not belong to any node. Malicious intermediate nodes can either modify or replay received packets.

Malicious node

Originator node   Intermediate node

Own address   Wormhole

Spoofed address   Replay packet

Arbitrary address   Modify packet

Fig 1: Malicious attack tree

This section concentrates on possible attacks on the various mechanisms used to discover and maintain routes in both proactive and reactive protocols. In particular, we investigate if the type of routing protocol used has a bearing on the effort needed to successfully perform such attacks. First, we discuss issues which are common for both types of protocol: the scope of malicious attacks and the use of sequence numbers. We then look at potential attacks that exploit the two main threats from malicious nodes: misdirection of packets and denial of service.

4.1 The scope of attacks arising from malicious behaviour

Even though an ad hoc network is vulnerable to attacks from an internal node, the scope of such an attack is limited. In general, the scope of an attack will be affected by two factors — distance and node density.

A malicious node can advertise network topology information which contradicts information provided by a well-behaved node, creating a case of the well known 'Byzantine generals problem' [30, 31]. For example, if a malicious node falsely advertises a direct route to a node which is connected elsewhere in the network, then the probability of other nodes accepting the false route depends on their relative distances to the malicious node and the well-behaved node. Nodes closer to the malicious node than the well-behaved node are more likely to accept the false information.

The resources required to mount an attack, together with the number of nodes that are affected, can be used as a measure of the efficiency of that attack, i.e. the effort that a malicious node has to make to achieve a specific amount of damage. In general, the more densely populated is the area in which a malicious node is located, the more nodes will be affected, and hence the greater the efficiency of the attack.

A malicious node could find itself as part of the only route between two or more groups of nodes. In this case, the malicious node can partition the network; the node can now attack one or both of the resulting partitions independently. Such a malicious node is able to control the scope of its attack by focusing on one partition, which may help it avoid detection. Other than finding some means of preventing malicious behaviour, one method of tackling this threat is to ensure that there are always multiple routes between any two nodes.

An attack will potentially have a much greater impact if it is performed by a group of malicious nodes, possibly colluding to perform a coordinated attack. It is difficult to make any assumptions about how many malicious nodes there will be in an ad hoc network, and if they have prior relationships which can be used to launch a distributed attack. Therefore, the effectiveness of any security solution should be assessed for various percentages of malicious nodes in the network.

4.2 Attacking sequence numbers

Many routing protocols use sequence numbers both as a duplicate suppression mechanism and to ensure freshness, i.e. to determine if a control packet contains the latest routing information. This has the added security benefit of providing some protection against replay attacks, thus helping to provide network integrity. However, sequence numbers only work when a node has previously received packets from the same originator, so that it can compare the sequence number in the received packet with previously received values. Thus, replay attacks are still possible, and this is expanded upon in section 4.3. 'Wormhole' attacks, which are also described in section 4.3, also exploit the same vulnerability. Note that protocols which do not use sequence numbers in their control messages are vulnerable to all forms of replay attack (the Topology Broadcast Reverse Path Forwarding (TBRFP) protocol [35] protocol is one such example).

Perlman [43] discusses the problem which occurs when sequence numbers reach their maximum value. This limit arises because sequence numbers are typically stored in a field of fixed length. Perlman suggests using a field which will be large enough so that the maximum value is never reached. For example, a 32-bit field contains 2^{32} sequence numbers, so that 248 days of continuous transmission, at a rate of 200 packets per second, would be needed to exhaust this series of sequence numbers. Since ad hoc networks are temporary and would not operate for such a length of time, this solution seems adequate.

While the use of incrementing sequence numbers is valuable as part of a routing protocol, it also introduces certain security vulnerabilities. To avoid being exploited by an attack, the sequence numbers should be increased by random integer values rather than a value of one. Of course, one implication of such an approach is that the range of available sequence numbers will be exhausted more quickly.

Wraparound mechanisms are used to 'reset' a sequence number to a lower value when it reaches its limit. As data transmission speeds become higher, wraparound will occur more often as more individual packets are sent. Most protocols using sequence numbers suggest the use of a mechanism for dealing with wrap-around, so that nodes will be able to detect when a sequence number has rolled over to restart from zero. These mechanisms can themselves introduce vulnerabilities which a malicious node can exploit to replay packets; these vulnerabilities exist because sequence numbers used in routing protocols are not intended to prevent replay attacks.

In AODV [36, 37, 42], a received packet containing the 32-bit sequence number r is accepted if and only if $0 < r - s \leq 2^{31}$, where s is the stored sequence number and $r - s$ is computed modulo 2^{32}. That is, a packet is accepted if and only if the sequence number it contains is in one half of the possible range of values.

Therefore, if a node has stored the sequence number value x, then it will accept any sequence numbers within the range $\{x, x + 1, \dots, x + 2^{31}\}$, i.e. half the sequence numbers in the series. Conversely it will reject the other half. While replay attacks are still theoretically possible, the large range of possible values mitigates the threat, although the same caveat as above remains; the range of sequence numbers may be less than the possible maximum because of the use of random increments.

While unique sequence numbers help to provide some protection against replay attacks, this mechanism can still be exploited to cause a type of denial of service called a preplay attack. A malicious node could impersonate existing nodes and flood the network with spoofed packets containing high sequence numbers. As a result, any messages sent from the true owners of the addresses will be discarded as either duplicate or out of sequence. The malicious node will know what sequence number a node is using from receiving a previously flooded control packet. If no wraparound mechanism is used, and a malicious node attacks using the maximum sequence number, then it may be necessary for the spoofed node to use a mechanism to 'reset' its sequence number within the ad hoc network.

In the AODV protocol, it is possible for a malicious node to inject sequence numbers for a specific other node without the need to masquerade. This is most easily achieved by replying to a route request with a route reply

containing a sequence number close to $x + 2^{31}$, where x is the most recently observed sequence number for the node under attack. The destination node will respond with its own route reply, which will be dropped by nodes which received the malicious node's reply, because the inequality above will be false.

When manipulating sequence numbers, an attacker will have to be judicious in order to both have a significant effect and remain undetected. Wang and Bhargava [54] show that an attacker cannot simply add a small constant to the sequence number, as this will be quickly overtaken by the true sequence number. On the other hand, a substantial increase may provide an indication of an attack.

Finally, note that the information in the rest of the packet does not necessarily have to be meaningful. The attacker only has to ensure a small number of fields in the header contain the correct information; typically the source address, destination address, sequence number and TTL fields. It is extremely difficult to determine if the payload of a data packet is valid, particularly if it is encrypted.

4.3 Misdirection attacks

In an ad hoc network, a malicious node may attempt to misdirect traffic to itself, or to another node. The relevant attack methods are:

- Masquerading as an existing node,
- Masquerading as a previously connected node,
- Replay attacks,
- Byzantine behaviour to attract traffic,
- Byzantine behaviour to deflect traffic, and
- Misdirection using a wormhole.

Conventionally, origin authentication mechanisms are needed to prevent masquerade attacks. In this section we first analyse the inherent security of proactive and reactive protocols when origin authentication is not used for internal nodes, or when a malicious node has found a way to circumvent the mechanisms used for origin authentication. In such cases, a malicious node may impersonate another node in order to misdirect traffic. This can be achieved by sending false control packets using an incorrect source address; this address could either belong to a node currently routing in the network, or it could be an address which is not currently being used, perhaps of a node which was previously connected to the ad hoc network. Such false control packets are also referred to as spoofed packets.

It is possible for a malicious node to send control packets containing false routing information, without the need to masquerade, by exhibiting Byzantine behaviour. Old and out-of-date routing information can also be considered as false, so a replay attack can also be used as the basis of a misdirection attack.

The attacks mentioned above require manipulation of the routing protocol; an alternative is to use a private communications channel to misdirect the traffic. This is also referred to as a 'wormhole' attack [21]. The six types of misdirection attack listed above are now considered in greater detail.

4.3.1 Masquerading as existing nodes

Using an address which belongs to another node currently connected to the ad hoc network is likely to be detected, especially in the case of control packets which are flooded throughout the network. In this case the true owner of the address will receive the spoofed packet.

In order to attract traffic to itself in a network using proactive routing, a malicious node could send spoofed Hello packets for neighbour sensing, so that its neighbours falsely add the spoofed address as a new neighbour. Even though the Hello packets involved are not flooded, this information will be flooded throughout the network via a topology control message, which would alert the true owner to the misbehaviour. However, nodes which are closer to the malicious node will still route traffic intended for the true owner of the address to the malicious node, as the malicious node is closer and offers a better route.

When reactive routing protocols are being used, a malicious node could spoof route requests so that nodes which receive the request will create a reverse route pointing towards the malicious node. Once again, whether this will have an effect depends on the proximity of the malicious node. These route requests are flooded throughout the network, and this will allow the true owner of the source address to detect them.

Nevertheless, to avoid detection by the true address owner, a malicious node could exploit the TTL mechanism, as in the attack on data packets described above (see section 3). The malicious node could flood a TTL restricted packet, which will not reach the true address owner. Indeed, this is a legitimate action in some reactive routing protocols, e.g. AODV, as part of the expanding ring search mechanism [36, 37, 42]. To perform such an attack effectively, the malicious node needs to know the distance to the node being spoofed. This is possible in a proactive routing scheme, as every node maintains an up-to-date view of the network topology. In a reactive routing protocol, the distance can be learnt from route requests sent by the spoofed node. In this case, we can deduce that reactive routing is likely to be more robust, because a malicious node may not have up-to-date distances for all nodes, whereas the inter-node distances are readily available in proactive routing protocols. Thus, use of a reactive routing protocol limits the window of opportunity for an undetected attack to the time immediately after receiving a request. The other factor affecting detection is the ability for nodes to move closer to the malicious node, because this will inevitably change the distance between the malicious node and the true address owner.

However, in a reactive routing protocol, once a malicious node has successfully masqueraded as another node and injected routes into a subset of the network, then the nodes in that subset will respond to any subsequent route requests with replies. This is a legitimate action designed to increase the scalability of ad hoc routing, so that the network can cope with more communicating nodes. In this way, not only can the malicious node avoid detection, but it could increase the likelihood of misdirecting packets to itself rather than the intended destination. That is, the more intermediate nodes the malicious node can 'poison', the more nodes will reply to a request. Figure 2 illustrates this attack scenario, where node *Sc* wants to find a route to node *Dt*. The thin lines indicate connectivity. No communication involving the true *Dt* has taken place recently, so *Dt*'s neighbours have timed out and deleted any routes. However, the malicious node, *Mal*, has been periodically broadcasting spoofed requests with a TTL value of 2 and an originator address *Dt*. The bold arrows indicate the currently known routes to *Dt*. When node *Sc* broadcasts a request, nodes *Q* and *S* will reply, and node *Sc* will create a route pointing to the malicious node *Mal*, which, as a result, has successfully performed a misdirection attack.

In a reactive routing protocol, there is a further way for a malicious node to spoof another existing node without alerting the true owner. This is through the use of unicast control packets, namely route reply packets, which are sent in response to route requests. A malicious node could falsely respond to a route request by sending a spoofed route reply. The originator of the request, who may receive more than one reply, will not be able to distinguish between the spoofed reply and the true reply, but it will accept whichever provides the route with a better metric. As discussed below, most reactive protocols even allow intermediate nodes to legitimately reply to a request, so spoofing replies is not necessary. Falsely advertising routes, with or without a more attractive routing metric, is Byzantine behaviour, which is discussed below.
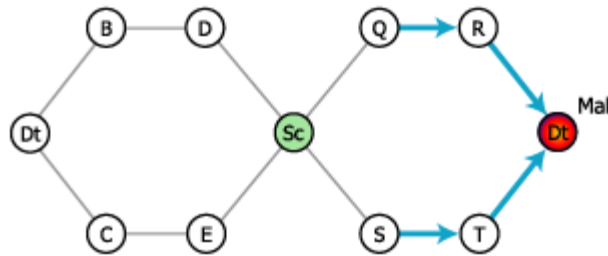
Fig 2: A misdirection attack using a reactive routing protocol

Finally, while we have discussed the possibility of detecting particular attacks, reacting to such an attack and identifying the culprit may be significantly more difficult. If address auto-configuration is being used, then any node can claim any address. After detecting an attack, the network will need some means of deciding who the legitimate owner of the address is. This issue is discussed further in section 4.4 below.

4.3.2 Masquerading as previously connected nodes

None of the previous analysis will apply when a malicious node uses an address which is not being used by an existing node. This could be an arbitrary address which has never been assigned, or the address of a node which is no longer connected to the ad hoc network. Finding an address that is not currently being used will be more difficult in a reactive routing protocol, where nodes will have an incomplete view of the nodes in the network, than in a proactive routing scheme, where nodes will know the addresses of most of the nodes participating in the network. Using arbitrary addresses for spoofed packets can form part of a denial of service attack, discussed later in section 4.4. Here, we concentrate on attacks that involve using the address of a previously connected node.

It is feasible for a malicious node to spoof a node which has momentarily, or permanently, left the network; as a result the rest of the network will falsely believe that this node is still connected. Alternatively, instead of waiting for a node to leave, a malicious node could use a denial of service attack to force a node to disconnect from the ad hoc network. The malicious node now assumes any history, security and routing decisions within the network layer, under the identity of the impersonated node[1]. In both types of protocol, the effect of the misdirection can be global. This is inherent in proactive routing because, when the malicious node sends spoofed Hello packets, the receiving neighbours will flood the new links throughout the topology. In a reactive routing protocol, in order to effect the attack, the malicious node will need to flood a route request, so that the nodes in the network create a reverse route back to the malicious node. Hence, the attack has the side effect of a denial of service attack (again, see section 4.4), because resources will be consumed dealing with the request flood.

4.3.3 Replay attacks

Replaying control packets can either be used as part of a denial of service attack, or as a means for a malicious node to misdirect traffic to itself. The malicious node does not have to replay control packets back into the network from the location at which it received the packets. As mentioned above in section 4.2, it is difficult to detect replay attacks if a node has not had any previous communication with the originator of the replayed packets, even if a sequence number freshness mechanism is used. Thus, this attack is more effective when using packets which are not propagated further than one hop, e.g. Hello packets using for proactive neighbour sensing, or those which are unicast, e.g. route reply or route error messages used in reactive routing schemes.

---

[1] An ad hoc network version of human identity theft.

In a proactive routing protocol, a malicious node can also replay topology control packets, to reintroduce old links or to coerce the network into removing existing ones. Topology control packets are generally protected with sequence numbers to enable the recipient of a packet to determine whether it is fresh, so the author of a replay attack will need to take into consideration the issues discussed in section 4.2.

A malicious node can also replay Hello packets from a node which has left the network. This is somewhat limited in its effect, as a Hello packet contains a list of neighbours used to determine bidirectional links. If none of the nodes listed in the replayed packet is a neighbour of the malicious node, then replaying the packet will not create any new links. Thus, the malicious node is limited to replaying the Hello packets in the same (or nearby) location as the node whose packets are being replayed, once the node has left the network.

In a reactive routing protocol, there is no such limitation on the location for replaying packets. In fact, if there is no benefit from attacking an area in which the nodes already have a route, then the malicious node will need to move to a different location to conduct an attack. When a route reply is replayed in an area where the nodes do not have a route to the originator, then they will create such a route in their routing tables. Even if the nodes already have a route as a result of a route request, then a malicious node could also have received the flooded request, and will know what sequence number is being used. In this case, the malicious node can determine the probability of success when replaying a route reply. Replaying a route error control packet is an example of a denial of service attack, which is discussed in section 4.4. Here we note that a malicious node can infer, from the receipt of a route error message, those areas of the network in which nodes do not have a route to a particular node.

4.3.4 Byzantine behaviour to attract traffic

In the presence of strong origin authentication mechanisms, malicious nodes can still misdirect traffic by sending false route information, i.e. they can exhibit Byzantine behaviour. In this case, a malicious node does not have to masquerade as another node. Instead, it can send control packets containing false routing information, using its own address. Alternatively, the node could modify control packets that it receives for forwarding. For example, the malicious node could advertise extra routes. To improve the likelihood of the false information being accepted by well-behaved nodes, the malicious node could falsely advertise favourable routing metrics to attract traffic. This is also known as the 'black hole' attack [14, 23, 49], by analogy to the celestial structure which sucks in all objects and matter.

Possible Byzantine behaviour in proactive routing protocols includes the possibility of a malicious node:

- Advertising high willingness to forward control packets;
- Advertising false links in a Hello packet;
- Advertising false links in a topology control packet;
- Including itself in topology control packets it receives for forwarding, and
- Removing links from topology control packets.

Each attack by itself suffers from restrictions of scope, as discussed in section 4.1. However, in combination they reinforce the effects of each another.

In a proactive routing protocol, mechanisms are used to try to reduce the number of redundant broadcasts [9, 10, 13, 32, 33, 34, 51, 52, 56, 57], so that not every node will receive control packets for forwarding. In such mechanisms, a malicious node can advertise a high relay priority or willingness value to indicate its ability to forward packets for its neighbours. This is a legitimate action designed to force neighbours to choose the malicious node for forwarding control packets. This attack does not, by itself, guarantee that a malicious node will be chosen, as other neighbours may be honestly advertising an equivalent willingness. So, in addition, the malicious node could advertise spurious neighbours in its Hello packet broadcasts. Since no other nodes will have a route to the

imaginary nodes, this will ensure that the malicious node is chosen to forward control packets, allowing the malicious node to attract control packets.

In order to attract data packets, the malicious node could advertise false links to nodes elsewhere in the network, so that other nodes believe the malicious node is close to a destination node when, in fact, it is not. As with spoofing packets above, the malicious node can use the TTL value to control the area in which its topology control packet will be flooded, to avoid detection by well-behaved nodes which do not have a link to the malicious node. Since the malicious node's neighbours could also detect the attack, the malicious node could reduce the chances of detection by advertising the false link in its Hello packets, which could be a part of the previous attack.

Another, similar, method to attract data packets is for a malicious node to add itself to any topology control packets it receives for flooding. Similarly, the malicious node could try to surreptitiously remove links from topology control packets, resulting in the malicious node being part of the perceived shortest route. These attacks help to reinforce the initial attack mentioned in the previous paragraph. There are two ways to control the scope of this attack. If the malicious node modifies the neighbour list of a topology control packet, then it will only be processed by those nodes which have not yet received it. These nodes are likely to be further away from the originator node than the malicious node, because closer nodes will detect that the packet has already been processed by examining the sequence number. If the malicious node also increments the sequence number, then all nodes will process the packet, including those closer to the originator. However, in such a case the originator node will also receive it, and will hence detect Byzantine behaviour.

Byzantine behaviour based attack methods also apply to reactive routing protocols, and we next describe four such attacks:

- *Modifying route metrics*: The process of reverse path setup in reactive route discovery means that nodes have to rely on routing metric information, contained within a control packet, to determine the best route to the packet's originator. The routing metrics used by reactive protocols are typically cumulative distance vector metrics, such as the number of hops. If a malicious node receives a route request or reply, then it could falsely decrease the hop count metric before forwarding the packet. However, even if the malicious node advertises a falsely low metric of zero hops, the number of hops in the control packet will increase as it propagates away from the malicious node.

- *Rushing attacks*: The nature of reactive route discovery means that reactive routing protocols are much more sensitive to network conditions than proactive routing protocols. This sensitivity can be exploited by malicious nodes. For example, a malicious node could rebroadcast route requests quicker than its neighbours. This is also known as a rushing attack [17, 22]. One method to achieve this is to exploit the Medium Access Control protocol, which will typically delay broadcasting of packets to avoid broadcast storms [34, 51]. A malicious node could ignore the delay, broadcast the request before its neighbours can, and, hence, increase the likelihood of being part of the final established route. Another method of realising a rushing attack is to use a wormhole, discussed in more detail below.

- *False route replies*: In order to improve the scalability of routing, reactive routing protocols typically trust intermediate nodes to reply to requests when they have an up-to-date route to the requested destination. Therefore, another way for a malicious node to coerce the network to send packets to it, is to reply to route requests regardless of whether it has a route or not. Using attractive metrics will increase the probability of success.

- *False gratuitous route replies*: AODV has a mechanism to ensure that only bidirectional routes are discovered; intermediate nodes which issue a reply also have to inform the requested destination by unicasting a gratuitous reply to it. Thus, the destination node and the intermediate nodes which receive the

gratuitous route reply will all add or update a route to the request originator. A malicious node could target nodes and send them gratuitous replies, claiming that they have been subject to route discovery. If the malicious node uses attractive metrics, then those nodes will update their routing tables to route through the malicious node.

One advantage that proactive routing protocols have over reactive protocols is that malicious attacks of this nature will have to be sustained, as otherwise the false routing information will be removed during periodic updating. This is not always the case with reactive routing. A route which is being used is maintained indefinitely. Thus, if a malicious node misdirects traffic to itself from a route which is in use, then that misdirection will remain while the nodes are sending packets over the route. On the other hand, reactive protocols can benefit from the fact that only those routes which are being used are maintained. Thus, such Byzantine misdirection attacks are difficult in the sense that they can only target routes which are being used.

4.3.5 Byzantine behaviour to deflect traffic

So far we have described attacks designed to misdirect traffic to a malicious node; however a malicious node could also direct traffic away from itself. This has also been called a gratuitous detour attack [23], and results in the network discovering and using suboptimal routes. Here we use suboptimal in the sense that there are better, more optimal routes. Of course, this attack is not possible if a malicious node is on the only route between two end points.

A malicious node could refuse to advertise connections or routes, or it could send or modify control packets so that they contain unappealing routing metrics. In a proactive routing scheme, a malicious node could refuse to advertise a link when originating topology control packets. The effects of this will be minimal, as the malicious node's neighbours will still advertise the link. The malicious node could also remove links from any topology control packets it receives for forwarding, manipulating the protocol so that nodes will calculate routes not involving the malicious node. A more effective attack would be for a malicious node to refuse to advertise links in a Hello message. Its neighbours would believe links with the malicious node are asymmetrical and, thus, would not advertise the link in their topology control packets.

In a reactive routing scheme, unless the malicious node is the target of a route request, in which case it can refuse to reply, it will need to wait for a rediscovery cycle to occur in order to misdirect traffic away from itself. Once a malicious node receives a route request, it can respond in a variety of ways:

- Modify the metric,
- Delay sending the route request, and
- Drop the route request without rebroadcasting it.

A malicious node could modify a route request before rebroadcasting it, so that it includes less attractive metrics. Adding false addresses to a source route and increasing the hop count are examples of ways in which this can be achieved. The cumulative nature of reactive routing metrics will help a malicious node to direct traffic away from itself. This contrasts with the attack described above, in which the accumulation process makes misdirection to attract traffic more difficult.

A malicious node could also delay rebroadcasting route requests. When the malicious node is part of one of several routes, then the malicious node could delay rebroadcasting long enough so that requests propagate through the rest of the network, and one of the other routes will be selected and used. Finally, a malicious node could also drop the request.

4.3.6 Misdirection using a wormhole

While the attacks we have so far described involve the manipulation of the routing protocol, it is also possible for a malicious node to misdirect traffic via a private communication channel. The 'Wormhole' attack was introduced by Hu, Perrig and Johnson [21]. A malicious node could tunnel packets to a colluding partner elsewhere in the ad hoc network, which replays them.

In a reactive routing network, tunnelling route requests to a node closer to the destination node will result in the tunnelled route being replied to, and route requests from all other routes being ignored. Thus the malicious node has injected itself into the route by using a form of rushing attack. It is debatable whether this form of wormhole attack is a disadvantage to other network nodes, as the tunnel provides a route with a lower delay than would otherwise be the case. However, there is the consequent threat of denial of service, in which the malicious node subsequently drops the packets received for transport via the tunnel.

Wormholes could also be used to tunnel Hello packets in proactive protocols. In order for such an attack to be successful, the two colluding nodes will need to tunnel Hello packets in both directions. The nodes at either end of the tunnel will incorrectly believe that they are neighbours of each other, and in turn, the network will have an incorrect view of the distance between the two nodes. The likely result is that the network will misdirect traffic to the malicious nodes. Deciding whether this attack has adverse effects on the network involves the same issues as discussed above for reactive protocols. As the wormhole is a means of misdirecting and attracting traffic, the issues with maintaining the attack also hold here.

4.4 Denial of service

The requirement of availability includes a need to prevent, or react to, denial of service attacks. In this section, we look at whether proactive or reactive routing schemes offer any inherent defences against denial of service attacks. Any action which unnecessarily uses bandwidth, computational and memory resources, and residual energy can lead to denial of service. Thus all attacks to misdirect traffic can also be viewed as denial of service attacks, i.e. a malicious node could introduce suboptimal routes which use unnecessary resources, and the network will consume even more trying to recover. These attacks were covered in the previous section. Here we are concerned with denial of service attacks in which the primary aim is to exploit the routing protocol to force unnecessary consumption of resources, so that other nodes cannot use the network for communication. Such attacks include:

- Injecting routes to false destinations,
- Flooding attacks involving control packets, and
- False removal of working routes.

As with data packets, an alternative method of attack is to refuse to provide service. This is akin to selfish behaviour, which is outside the scope of this paper.

There is another type of denial of service threat that is more subtle. Most of the misdirection threats lead to attacks which induce more resources to be used, either as a result of the attack itself, or in recovering from the attack. Also, malicious nodes can cause other nodes to exhaust their resources by forcing them to do unnecessary processing of correct or incorrect information. Those nodes which use up their power resources will eventually become unable to operate under normal circumstances. This has been referred to as a resource consumption attack [23], or, more descriptively, a 'sleep deprivation torture' attack [47, 48]. The threat of sleep deprivation only arises in ad hoc networks where one or more of the nodes are power constrained.

4.4.1 Injecting routes to false destinations

Injecting false routes to non-existent nodes will lead to resources being used to maintain those routes, and also allows further denial of service if malicious nodes send spurious packets for those non-existent destinations. Thus, nodes will waste resources trying to send packets over false routes, and may even use additional resources trying to recover an alternative route. We compare a proactive routing protocol, OLSR, to a reactive routing protocol, AODV.

In a proactive routing scheme, including links to arbitrary addresses in topology control packets will lead to increased use of bandwidth to flood larger packets, and memory being occupied storing routes to the non-existent addresses. The effects of the attack also include the use of computational resources for route table calculations involving false links.

The number of false links that can be advertised in a single topology control packet will to some extent determine the degree of denial of service that can be achieved. If we suppose that the OLSR protocol is being used with IPv4 and IEEE 802.11, then we can make the following calculations for a single topology control packet.

One topology control packet occupies one IP packet. The maximum payload of an IP packet is 65 535 bytes. 48 bytes will be used for the IP, UDP and OLSR headers if no options are utilised (20, 8 and 20 bytes respectively), so that 65 487 bytes remain, which could be used to store 16 371 false 32-bit IP addresses. However, the Maximum Transmission Unit (MTU) of one IEEE 802.11 frame is 2346 bytes[2] so that the packet has to be fragmented into several frames. The IP header of the packet has to be copied into each frame, except the first which 'acquires' the IP header of the packet. This leaves 2326 bytes in each frame to contribute to the packet payload. Thus, after taking the first frame into consideration, we are left with 63 189 bytes to fragment into 28 frames, and 29 frames equates to 68 034 bytes. This is the minimum bandwidth that will be used in the local neighbourhood of a malicious node if it sends a full OLSR topology control packet; the actual amount will be slightly greater because of the control messages used by the underlying Medium Access Control protocol.

In a reactive routing scheme, this form of denial of service attack requires more effort on the part of the malicious node, as the malicious node can only inject one false address per control packet, but in general this will then translate into a more effective denial of service attack. The implications of this are twofold.

We consider spoofed route requests rather than replies, as they have a much greater impact on the other network nodes. When using the AODV protocol, each route request occupies 24 bytes of an IP packet payload. Thus, assuming that UDP and IPv4 are being used, introducing one false destination requires 52 bytes to be sent in the local neighbourhood. Compared to the calculations for OLSR above, injecting 16 371 false addresses into the network using reactive routing requires 851 292 bytes as opposed to 68 034 bytes.

This is much more effort than sending one topology control message in a proactive routing scheme, but the denial of service is much greater. The impact of the attack can vary considerably if the malicious node adds spurious information after an AODV route request to occupy the rest of the transmission frame.

In both cases, the attacks will need to be repeated regularly to achieve a sustained effect, otherwise the non-existent routes will be deleted through periodic updating in proactive protocols, and timed out in reactive proto-cols, as no nodes will use them. However, in a reactive routing protocol, a malicious node, possibly colluding with others, could maintain the effect of the attack by periodically sending packets addressed to the spurious addresses. These packets will travel along the false routes until they reach a node with no route to forward them, where they will be

---

[2] However, most implementations of IEEE 802.11 use an MTU of 1500, so that packets travel over IEEE 802.3 Ethernet networks without further fragmentation [24].

dropped. This node will then begin the process of route maintenance and send route error packets. More resources will be wasted if nodes attempt to discover alternative routes.

Finally, another related attack, identified by Sangirzi et al. [46], applies specifically to source route based protocols such as DSR. Nodes using DSR update their route caches from any 'overheard' packets. Thus, a simple attack would be to send data packets containing false source routes, in order to 'poison' the route caches of any nodes which hear the packet.

4.4.2 Flooding attacks

Both proactive and reactive protocols use control packets which need to be flooded throughout the network. Proactive protocols require topology control packets to be flooded, whereas reactive protocols rely on the flooding of route requests. Uncontrolled flooding will potentially consume all the available bandwidth, preventing legitimate traffic travelling through the network.

There is no mechanism to control the rate at which topology control packets should be flooded in proactive routing protocols. Thus, a malicious node could flood the network with topology control packets as a denial of service attack.

The denial of service threat from repeated request broadcasts appears to be difficult to deal with [23, 47, 48]. A malicious node could flood the network with continuous route requests for destinations which do not exist, or destinations which do, in order to elicit route replies. The malicious node does not need to use the discovered route; it could wait until the intermediate nodes time out and delete the route, before flooding another request.

Although the protocols typically include rules for well-behaved nodes, for example, to perform exponential backoff between unsuccessful route discoveries, there is no mechanism to enforce good behaviour. Thus, there is no benefit in choosing either class of protocol with regard to flooding attacks.

4.4.3 False removal of working routes

Most reactive routing protocols rely on a route maintenance mechanism to prevent nodes from sending packets for routes which are no longer active. If unprotected, the route maintenance mechanism is highly vulnerable to attack. If the malicious node is part of the route, then it could send route error messages to force all upstream nodes to mark the route as inactive. Even if a malicious node is not part of a route, but is nearby, then it may masquerade as an intermediate node and send a spoofed route error message. Alternatively, a malicious node could store a route error packet, perhaps induced earlier as a result of a denial of service attack, and replay it at a later time; typically, route error messages do not include a means of checking for freshness. One effect of such an attack is that nodes will falsely believe the route is broken, and may waste resources in trying to discover another route, which could in fact be the same route that was used before the attack.

Proactive routing protocols do not have an explicit link recover mechanism. Instead, link breaks are inferred when a node stops advertising a link to a neighbour. Unless the protocol makes use of triggered updates, there will be a delayed reaction until the next periodic update packets are flooded. Thus, proactive protocols are more robust against such an attack, as there is no means of falsely inducing route breaks. Instead, the malicious node has to rely on other denial of service attacks, such as impersonating a node on the route in order to spoof Hello messages containing empty sets of links.

4.4.4 Sybil attacks

An attack which indirectly affects routing protocols is the 'Sybil' attack, introduced by Doucer [16]. This is an attack on the identity management system of any distributed system which, in this case, is an attack on the ad hoc network addressing system. In a Sybil attack, a malicious node claims as many addresses as it can, preventing other nodes from using those addresses, thus denying them a means to participate in the ad hoc network.

Routing typically assumes that each node has a unique address. In reality, this address has to be assigned or acquired, e.g. using a dynamic address auto-configuration scheme. This often includes a Duplicate Address Detection (DAD) protocol to verify the uniqueness of an address. A newly connected node will choose a tentative address and flood this throughout the network in the form of an 'Address Query' packet. If another node in the network is using the address, i.e. there is an address collision, then it replies with a 'Duplicate Address' packet. There are many variants of address auto-configuration, and more information can be found in [7, 50, 53, 55].

Certain issues with address configuration relate to the type of routing protocol used. As discussed above, malicious nodes could send a multitude of spoofed control packets. Even if a malicious node uses an address which is already being used in the network and this is detected, it is still difficult to react to such an attack, as it will not be possible to use the routing protocols to determine which node is the true owner of the address.

In a reactive routing scheme, there is no explicit neighbour sensing mechanism. Instead, neighbour nodes are inferred from the control packets received. Hence, there is no protection against a Sybil attack. This is not necessarily the case in a proactive routing protocol. When considering secure address auto-configuration, proactive routing protocols offer limited defence against Sybil attacks. In particular, the k-out-of-n neighbour sensing mechanism, as used by TBRPF [35], should be employed.

When introducing the Sybil attack [16], Doucer describes three methods of trying to decrease the achievable damage:

- Only accept identity messages within a certain time interval,
- Challenge each identity to store a certain amount of data, and
- Challenge the identity to solve a unique cryptographic puzzle.

A k-out-of-n neighbour sensing mechanism is an example of the first method. A neighbour is considered active if a Hello packet is received from the neighbour in at least k from the n most recent time intervals, where each time interval has period t. By basing neighbour sensing on the number of Hello packets received, the protocol creates more stable routes and offers some interesting security properties.

It will be impossible for a neighbour of a new node to claim ownership of an address that the new node is trying to use, without first broadcasting the appropriate number of Hello packets.

One of the effects of proactive routing is that each node will know the addresses of all nodes currently participating in the ad hoc network. Thus, a new node's neighbours will have the addresses of all the nodes in the network stored in their routing tables. Many existing DAD protocols use this property so that neighbour nodes can reply immediately to an address query, to remove the need for inefficient flooded Address Queries packets [50]. However, the possibility remains that two nodes joining the ad hoc network simultaneously in different areas of the network could try to claim the same address. This could even be the result of collusion between two malicious nodes; a malicious neighbour could easily send false Duplicate Address replies.

Therefore, for security, an Address Query message from a new node should be flooded throughout the network, as the new node may not trust its neighbours to respond truthfully. Ideally, this should be performed using an efficient flooding protocol. Flooding has two additional security benefits. Firstly, by forcing the true owner of the

address to reply, further authentication mechanisms can be used, for example, those based on digital signature mechanisms. Secondly, even if an Address Query packet is dropped by a malicious node, it could still be sent to the destination via other routes, at the cost of extra bandwidth consumption.

A malicious node could try to claim the address specified in an Address Query message, in order to send a legitimate Duplicate Reply. As above, this attack is mitigated by the k-out-of-n neighbour sensing scheme. In order for the attack to work, the malicious node will need to have broadcast Hello messages using the new node's potential address as the source address during at least k of the most recent n time intervals, before its neighbours will recognise the address as belonging to an active neighbour. If we suppose that the address space is large, which is likely if IP addressing is used, and that the probability of the malicious node guessing the address chosen by the new node is low, then the attack is prevented on the condition that there is at least one honest node between the malicious originator of the Duplicate Address reply and the new node.

When considering a Sybil attack, the k-out-of-n neighbour sensing protocol also restricts the number of addresses that a malicious node can claim. If the time taken to broadcast a Hello message is b, then the maximum number of Hello messages with different source addresses that the malicious node can get accepted is approximately nt/kb.

Hence, a malicious node can only claim a limited subset of the address space. However, this does not prevent colluding malicious nodes broadcasting a large volume of Hello messages with a variety of different source addresses. A large address space may be the only defence against this, if there is no means of linking an address to a node during its lifetime in an ad hoc network.

5 Analysis

In this section, we present an analysis of the attacks described in sections 3 and 4; particular attention is given to the relative effects of these attacks on proactive and reactive routing protocols.

5.1 Attacking the routing of data packets

Routing in an ad hoc network requires trusting intermediate nodes to follow the rules of the routing protocol. Malicious nodes could ignore these rules in order to exploit the routing protocol to attack the network. In general, both proactive and reactive routing protocols use the same mechanisms to route data packets, so the malicious threats described in section 3 apply equally to both types of routing protocol.

One example of such an attack is to modify the destination of a data packet in order to misdirect it to a different location, or, alternatively, so that it will be dropped before it reaches its destination. This highlights the vulnerability of using unprotected IP headers. In conventional wired networks, this header and the information within can be protected using the IPsec architecture [15, 29]. However, IPsec could be too bandwidth intensive for use in an ad hoc network, especially if it is used in conjunction with other security mechanisms for securing the routing protocol itself (see section 5.2).

Data packets ultimately contain information for use by services and applications in the upper layers of the protocol stack. By replaying data packets, a malicious node can try to force a principal to accept out-of-date information. To mitigate such an attack, applications and services in the upper layers typically use freshness mechanisms to recognise replayed data. Such mechanisms are not used in the network layer, to avoid duplicating these services in multiple layers. This approach was adopted because the routing infrastructure was never considered to be vulnerable to replay attacks. However, with ad hoc networking, replay attacks can cause significant denial of service when using either type of routing protocol, which may require this assumption to be revisited.

Even when following the rules of either type of routing protocol, a malicious node can still execute denial of service attacks. The current definitions of ad hoc network routing protocols mean that an ad hoc network is an open resource with no restrictions placed on communication. Thus a malicious node can emulate the behaviour of a communication intensive node, and cause denial of service by consuming the communication bandwidth to prevent other nodes from communicating. This vulnerability is present in both types of routing protocol.

5.2 Attacking the routing protocol

From section 4, two types of routing protocol threats are identified — misdirection attacks and denial of service; we analyse both by comparing the attacks on proactive and reactive routing protocols.

5.2.1 Misdirection

If a malicious node can masquerade using a variety of addresses, then it could perform several attacks to misdirect traffic to itself. By trying to impersonate another node in the network, a malicious node is in contention with the true owner of the address. The malicious node can avoid detection by the true owner by judicious control of the TTL value; this is more difficult in reactive routing schemes because distance information for a node may not be available. However, because routes are not kept if they are not being used, the scope of an attack for a malicious misdirect can increase; this is not the case with proactive routing. With the added vulnerability of spoofing unicast route replies, reactive protocols are more vulnerable to spoofing attacks which misdirect traffic to a malicious node.

Instead of spoofing an existing node, it is easier to masquerade as a node which has disconnected itself from the network. In this case there is no difference, in effort or effect, between reactive and proactive routing: the malicious node simply resumes normal operation under a different identity. However, replaying packets to misdirect traffic is much easier when reactive routing is being used. The use of sequence numbers and the two-way dialogue and verification of bidirectional routes implicitly prevent replay attacks in proactive routing protocols. In reactive routing protocols, the potential lack of complete topology information allows a malicious node to replay packets in areas where routes to the originator of those packets are not stored.

Proactive routing is pervasive, in that all nodes receive information about all routes. Since this information is maintained by all nodes, any irregularities or conflicts that are a result of Byzantine behaviour can be detected. This contrasts with the lack of pervasive information in reactive routing. While this potentially contributes to greater efficiency in reactive routing schemes, it makes attacks more difficult to detect. However, having pervasive routing information means that any false information will influence the routing decision of all nodes, rather than just a subset as is the case for reactive routing.

Perlman [43] states that distance vector protocols need more cooperation than link state, which also applies to proactive and reactive routing. The only requirement for cooperation in proactive routing is for nodes to rebroadcast unmodified topology control packets; since they are flooded, even if nodes drop them they will still reach the majority of the network. Reactive routing protocols, on the other hand, require nodes to cooperate at different stages of the route discovery cycle. Intermediate nodes are trusted to modify and forward control packets in a timely manner. While cumulative distance vectors such as the number of hops offer some resistance against misdirection attacks to attract traffic, they are a means for a malicious node to deflect traffic. Note that the sensitivity of reactive routing to the time at which a request is flooded, is not only affected by rushing attacks and gratuitous detour attacks, but also the network environment. Therefore, distinguishing an attack from a natural event will be difficult.

Proactive routing protocols separate the functions of route maintenance and data packet routing. Thus, Byzantine attacks need to be sustained because, otherwise, periodic updating will remove information regardless of whether a route is being used or not. This is not the case with reactive routing schemes, in which the two functions are

dependent on each other: if a route is being used, then it will be maintained in the routing tables of the nodes transmitting over the route, and a misdirection attack will remain as long as the route is being used and does not break. An advantage that reactive routing does have, however, is that Byzantine misdirection attacks can only target routes which are being used.

Finally, we consider wormhole attacks. Although they can certainly be used to attract traffic by providing a route with minimum delay, this effect is analogous to ad hoc networking nodes finding a shortest route. If the attackers withdraw the wormhole, then both types of routing protocol will react in the same way as for any other route break.

5.2.2 Denial of service

Injecting false routes is relatively easy with both proactive and reactive routing protocols; the difference is once again the separation of functions. With proactive routing, a malicious node will need to maintain the routes by sending false topology control packets, in addition to sending spurious packets. With reactive routing, once the false routes have been set up, the malicious node can send spurious packets which trigger route maintenance.

The other main difference between the two types of routing protocol is the extra bandwidth needed by reactive routing to inject an equivalent number of false destinations. This means extra effort is required on the part of the malicious node, but that extra effort translates to greater denial of service. Also, we can conclude that this attack is more efficient on proactive routing protocols.

Proactive and reactive protocols both use flooded control packets, which are vulnerable to a variety of attacks, as has been identified above. A malicious intermediate node could modify any control packets it receives for forwarding. In a proactive routing protocol, the flooding of false topology control packets can be prevented using conventional origin authentication and integrity mechanisms, as they are not modified during flooding.

This is not the case with reactive routing protocols. The metric information in the control packets of the route discovery cycle, e.g. the DSR source route or the AODV hop count, need to be updated by intermediate nodes before control packets are transmitted. Zapata and Asokan [63] refer to such control packets fields as mutable, and those fields which should not be modified, e.g. the source address, as immutable. The presence of mutable control packet fields in reactive routing protocols means that origin authentication and integrity mechanisms will need to be applied in a different, possibly more complicated, way than in proactive routing protocols.

6   Conclusions

There are two types of ad hoc network routing protocol, proactive and reactive, each with defining properties that also determine the inherent robustness of the type of protocol. Routing in an ad hoc network consists of the transport of data packets, using routes that are discovered and maintained through the exchange of protocol control packets.

The effects of attacking the transport of data packets do not appear to differ with respect to the type of routing protocol being used. Therefore, generic mechanisms should be designed to mitigate the attacks arising from modifying packet headers, flooding packets and replaying and reordering data packets.

This is not the case with the route discovery and maintenance protocols, where there are significant differences in the vulnerabilities in a proactive routing protocol compared to a reactive routing protocol.

A malicious node has several options of attack in both types of routing protocol. When trying to misdirect traffic, a malicious node can masquerade as another existing node and this is less likely to be detected by the impersonated

node when using reactive routing protocols. However, while it is important to prevent masquerading using origin authentication mechanisms to protect the nodes currently connected to the network, it is also essential to detect masquerades using either the identities of nodes which were previously connected or spurious identities invented by a malicious node; both types of routing protocol are equally susceptible to the latter two methods of attack.

Preventing masquerade attacks may lead to malicious nodes demonstrating Byzantine behaviour to attract or deflect traffic. There are attack strategies in both proactive and reactive routing for attracting data packets, but these attacks are more likely to be detected in proactive routing protocols because of the period and pervasive distribution of routing information; while conflicting information may not identify the malicious attacker, it will at least reveal that a potentially disruptive routing event is taking place. The periodic updating of information means that Byzantine attacks require more sustained effort when attacking pro-active routing protocols.

Misdirection attacks also induce denial of service, as they consume resources. Denial of service attacks also induce a misdirection attack, as the network attempts to reroute around the area where denial of service is occurring. This is a dangerous cyclical chain of events that malicious nodes could exploit to render an ad hoc network unusable. Thus, security mechanisms need to be deployed to try and break this cycle.

The provision of security may prove to be a key deciding factor when choosing which type of routing protocol to use. Origin authentication and integrity mechanisms can be conventionally applied to proactive routing, as the control packets used are not modified as they travel through the ad hoc network. Even though reactive routing protocols may offer significant gains in saving resource compared to proactive routing, they are more complex and may also be more difficult to secure. With the added benefit of being at least partially resistant to Sybil attacks, proactive routing protocols are inherently more secure than reactive routing protocols. These conclusions should be taken into account when choosing which ad hoc routing protocol to use.

References

[1]   I. Aad, J.-P. Hubaux, and E. W. Knightly. Denial of service resilience in ad hoc networks. In Z. J. Haas, S. R. Das, and R. Jain, editors, Proceedings of the 10th annual international conference on Mobile computing and networking, MobiCom '04, Philadelphia, PA, US, October 26, 2004, pages 202–215. ACM Press, Oct 2004.

[2]   M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks, (2):1–22, 2004.

[3]   G. Athanasiou, L. Tassiulas, and G. S. Yovanof. Overcoming misbehaviour in mobile ad hoc networks: An overview. Crossroads The ACM Student Magazine, (114):23–30, 2005.

[4]   J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '98, Dallas, Texas, US, October 25-30, 2002, pages 85–97. ACM Press, Oct 1998.

[5]   S. Buchegger and J.-Y. Le Boudec. Self-policing mobile ad hoc net-works. In Mohammad Ilyas and Imad Mahgoub, editors, Mobile Computing Handbook, pages 435–456. CRC Press, 2004.

[6]   S. Buchegger and J.-Y. Le Boudec. Self-policing mobile ad hoc networks by reputation systems. Communications Magazine, IEEE, 43(7):101– 107, Jul 2005.

[7]   F. Buiati, R. Puttini, R. de Sousa, C. J. Barenco Abbas, and L. J. García Villalba. Authentication and autoconfiguration for MANET nodes. In L. T. Yang, M. Guo, G. R. Gao, and N. K. Jha, editors, Embedded and Ubiquitous Computing: International Conference, EUC 2004, Aizu-Wakamatsu City, Japan, August 25-27, 2004, pages 41–52. Springer-Verlag (LNCS 3207), Aug 2004.

[8] G. Carl, G. Kesidis, R. R. Brookes, and S. Rai. Denial-of-service attack-detection techniques. IEEE Internet Computing, 10(1):82–89, Jan/Feb 2006.

[9] J. Carle and D. Simplot-Ryl. Energy-efficient area monitoring for sensor networks. Computer, 37(2):40–46, Feb 2004.

[10] T. Clausen and P. Jacquet. Optimised link state routing protocol OLSR. RFC 3626, Internet and Engineering Task Force, Oct 2003.

[11] D. E. Comer. Internetworking with TCP/IP -principles, protocols and architectures. Prentice Hall, 4th edition, 2000.

[12] M. Conti, E. Gregori, and G. Maselli. Cooperation issues in mobile ad hoc networks. In Proceedings of the 24th International Conference on Distributed Computing Systems Workshops — W6: WWAN (ICDCSW '04), Hachioji, Tokyo, Japan, 23–24 March 2004, pages 803–808. IEEE Computer Society, Mar 2004.

[13] F. Dai and J. Wu. Distributed dominant pruning in ad hoc networks. In Proceedings of the IEEE International Conf. on Communications (ICC '03), Anchorage, US, May 12, 2003, pages 353–357. IEEE Press, May 2003.

[14] H. Deng, W. Li, and D. P. Agrawal. Routing security in wireless ad hoc networks. IEEE Communications, 40(10):70–75, Oct 2002.

[15] N. Doraswamy and D. Harkins. IPSec: The new security standard for the internet, intranets and virtual private networks. Prentice Hall PTR, 2nd edition, 2003.

[16] J. R. Doucer. The sybil attack. In P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, editors, Peer-to-Peer Systems: First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers, pages 251–260. Springer-Verlag (LNCS 2429), Mar 2002.

[17] D. Dreef, S. Ahari, K. Wu, and V. King. Utilizing the uncertainty of intrusion detection to strengthen the security of ad hoc networks. In I. Nikolaidis et al., editor, Ad-Hoc, Mobile, and Wireless Networks: 3rd International Conference, ADHOC-NOW 2004, Vancouver, Canada, July 22-24, 2004, pages 82–95. Springer-Verlag (LNCS 3158), Jul 2004.

[18] J. A. Freebersyer and B. Leiner. A DoD perspective on mobile ad hoc networks. In C. E. Perkins, editor, Ad Hoc Networking, chapter 2, pages 29–51. Addison-Wesley, 2001.

[19] L. Guang and C. Assi. On the resiliency of mobile ad hoc networks to MAC layer misbehavior. In Proceedings of the 2004 ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2005), Montreal, Canada, October 10–13, 2005, pages 160–167. ACM Press, Oct 2005.

[20] V. Gupta, S. Krishnamurthy, and M. Faloutsos. Denial of service attacks at the MAC layer in wireless ad hoc networks. In Proceedings of the IEEE Military Communications Conference, MILCOM 2002, Anaheim, California, October 7-10, 2002, volume 2, pages 1118–1123. IEEE Press, Oct 2002.

[21] Y.-C. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, CA, April 1-3, 2003, volume 3, pages 1976–1986. IEEE Press, Apr 2003.

[22] Y.-C. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2003 ACM workshop on Wireless security (WiSe 2003), San Diego, CA, September 19, 2003, pages 30–40. ACM Press, Sep 2003.

[23] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. Wireless Networks, Springer Science, (11), 2005. 21–38.

[24] Institute of Electrical and Electronics Engineers. ISO/IEC Standard 802.3-2002 IEEE Standards for Information Technology – Telecommunications and Information Exchange between Systems – Local and

Metropolitan Area Network – Specific Requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2002.

[25]   P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimised link state routing protocol for ad hoc networks. In Proceedings of the 5th IEEE Multi Topic Conference (INMIC 2001), Lahore, Pakistan, December 28-30, 2001, pages 1–7. IEEE Press, Dec 2001.

[26]   D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imelinski and H. Korth, editors, Mobile Computing, chapter 5, pages 153–179. Kluwer Academic Publishers, 1996.

[27]   D. Johnson, D. Maltz, and J. Broch. DSR — The dynamic source routing protocol for multihop wireless ad hoc networks. In C. Perkins, editor, Ad Hoc Networking, chapter 5, pages 139–172. Addison-Wesley, 2001.

[28]   J. Jubin and J. Tornow. The DARPA packet radio network protocols. Proceedings of the IEEE, 75:21–32, 1987.

[29]   S. Kent and R. Atkinson. Security architecture for the internet protocol. RFC 2401, Internet and Engineering Task Force, Nov 1998.

[30]   L. Lamport. The weak Byzantine generals problem. Journal of the ACM (JACM), 30(3):668–676, 1983.

[31]   L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3):382–401, 1982.

[32]   H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWIM '00), Boston, Massachusetts, US, August 20, 2000, pages 61–68, Aug 2000.

[33]   W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. IEEE Transactions on Mobile Computing, 1(2):110– 122, Apr-Jun 2002.

[34]   S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99, Seattle, Washington, US, August 15-19, 1999, pages 151–162. ACM Press, Aug 1999.

[35]   R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding TBRPF. RFC 3684, Internet and Engineering Task Force, Feb 2004.

[36]   C. Perkins and E. Belding-Royer. Ad hoc on-demand distance vector AODV routing. RFC 3561, Internet and Engineering Task Force, Jul 2003.

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

[38]   C. E. Perkins. Ad Hoc Networking. Addison-Wesley, 2001.

[39]   C. E. Perkins. Ad hoc networking: An introduction. In C. E. Perkins, editor, Ad Hoc Networking, chapter 1, pages 1–28. Addison-Wesley, 2001.

[40]   C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In Proceedings of the conference on Communications architectures, protocols and applications, SIGCOMM '94, London, UK, August 31-September 02, 1994, pages 234–244. ACM Press, 1994.

[41]   C. E. Perkins and P. Bhagwat. DSDV routing over a multihop wire-less network of mobile computers. In C. E. Perkins, editor, Ad Hoc Networking, chapter 3, pages 53–74. Addison-Wesley, 2001.

[42]   C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 25-26, 1999, pages 90–100. IEEE Press, 1999.

[43]   R. Perlman. Network layer protocols with Byzantine robustness. Technical Report MIT-LCS-TR-429, Laboratory for Computer Science, Massachusetts Institute of Technology, October 1988.

[44]   J. Postel. User datagram protocol. RFC 768, Internet and Engineering Task Force, Aug 1980.

[45]   J. Postel. Internet protocol DARPA internet program protocol specification. RFC 791, Internet and Engineering Task Force, Sep 1981.

[46]   K. Sanzgiri, D. LaFlamme, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. Authenticated routing for ad hoc networks. IEEE Journal on Selected Areas of Communications, 23:598–610, March 2005.

[47]   F. Stajano. Security for ubiquitous computing. John Wiley & Sons, Ltd., 2002.

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

[49]   B. Sun, Y. Guan, J. Chen, and U. Pooch. Detecting black-hole attack in mobile ad hoc networks. In Proceedings of the 5th European Personal Mobile Communications Conference, Glasgow, UK, April 22-25, 2003, pages 490–495. Institution of Electrical Engineers, IEE Press, Apr 2003.

[50]   S. Thomson and T. Narten. IPv6 stateless address autoconfiguration. RFC 2462, Internet and Engineering Task Force, Dec 1999.

[51]   Y.-C. Tseng, S.-Y Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. Wireless networks, 8(2-3):153–167, 2002.

[52]   Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. IEEE Transactions on Mobile Computing, 52(5):545–557, Apr-May 2003.

[53]   N. H. Vaidya. Weak duplicate address detection in mobile ad hoc networks. In J. Hubaux, J. J. Garcia-Luna-Aceves, and D. Johnson, editors, Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '02, Lausanne, Switzerland, June 9-11, 2002, pages 206–216. ACM Press, Jun 2002.

[54]   W. Wang, Y. Lu, and B. Bhargava. On security study of two distance vector routing protocols for mobile ad hoc networks. In M. Singh and V. K. Prasanna, editors, Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, Per-Com '99, Dallas, Texas, US, March 23-26, 2003, pages 179–186. IEEE Press, Mar 2003.

[55]   K. Weniger. Passive duplicate address detection in mobile ad hoc net-works. In Proceedings of IEEE Wireless Communications and Networking (WCNC 2003), New Orleans, LA, March 16-20, 2003, volume 3, pages 1504– 1509. IEEE Press, Mar 2003.

[56]   J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. In Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, CA, April 1-3, 2003. IEEE Press, Apr 2003.

[57]   C.-C. Yang and C.-Y. Chen. A reachability-guaranteed approach for reducing broadcast storms in mobile ad hoc networks. In Proceedings of the 56th ACM Vehicular Technology Fall 2002 Conference (VTC 2002), Vancouver, Canada, September 24-29, 2002, volume 2, pages 1036–1040. IEEE Press, Sep 2002.

[58]   P. Yau. Issues surrounding the operation of mobile ad hoc networks in the presence of selfish behaviour. In Proceedings of the Wireless World Research Forum Meeting 10, New York, NY, USA, October 27-28, 2003. Wireless World Research Forum, Oct 2003.

[59]   P. Yau and C. J. Mitchell. Reputation methods for routing security for mobile ad hoc networks. In Proceedings of SympoTIC '03, Joint IST Workshop on Mobile Future and Symposium on Trends in Communications, Bratislava, Slovakia, October 26-28, 2003, pages 130–137. IEEE Press, Oct 2003.

[60]   P. Yau and C. J. Mitchell. Security vulnerabilities in ad hoc networks. In Proceedings of the seventh International Symposium on Communication Theory and Applications (ISCTA '03), Ambleside, UK, July 13–18, 2003, pages 99–104. HW Communications Ltd., Jul 2003.

[61]   P. Yau and V. Sdralia. Requirements for secure routing in ad hoc networks. In Proceedings of the Wireless World Research Forum Meeting 10, New York, NY, USA, October 27-28, 2003. Wireless World Research Forum, Oct 2003.

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

[63]   M. Zapata and N. Asokan. Securing ad hoc routing protocols. In D. Maughan and N. Vaidya, editors, Proceedings of the 3rd ACM work-shop on Wireless Security (WiSe '02), Atlanta, Georgia, US, September 28, 2002, pages 1–10. ACM Press, 2002.

[64]   L. Zhou and Z. Haas. Securing ad hoc networks. IEEE Network, 13(6):24–30, Nov 1999.

[65]   X. Zou, B. Ramamurthy, and S. Magliveras. Routing techniques in wireless ad hoc networks — classification and comparison. In N. Callaos, I. Nunes da Silva, and J. Molero, editors, Proceedings of the Sixth World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2002), Florida, US, July 14–18, 2002, volume 4. IIIS, July 2002.