Attacks and Defenses of Routing Mechanisms in Ad Hoc and Sensor Networks

You-Chiun Wang and Yu-Chee Tseng

Abstract—An ad hoc or a sensor network is composed of a collection of wireless nodes. A node can only communicate with other nodes within its limited transmission range. To facilitate communication between two nodes without a direct communication link, routing protocols must be developed to support multi-hop communication. Although many routing protocols have been proposed for ad hoc and sensor networks, most of them assume that other nodes are trustable and thus do not consider the security and attack issues. To assure a source node of finding a route to its destination, most routing protocols try to invite all available nodes to participate in the routing mechanism. This provides a lot of opportunities for attackers to destroy the routing mechanism. In this chapter, we briefly introduce some existing routing protocols, discuss the weaknesses of these protocols and possible types of attacks, and provide a comprehensive survey of recent research on defense approaches to these attacks.

Index Terms—ad hoc networks, routing protocols, security, wireless sensor networks.

1 Introduction

Recently, the rapid development of wireless communication not only alleviates the wired problem of traditional networks, but also provides the capability of mobile communication and ubiquitous computing. The ad hoc network architecture is a representative example and has been proposed to rapidly set up a network when needed [1]. An ad hoc network consists of a collection of wireless nodes. Each node can directly communicate with other nodes within its transmission range. Communication between out-of-range nodes has to be routed through one or multiple intermediate nodes. Thus, each node also acts as a router. Since nodes may be mobile, the corresponding protocol should be able to handle rapid topology change.

A sensor network is also considered an ad hoc network in which nodes are extended with sensing capability. Such a network is composed of one or multiple *remote sinks* and many tiny, low-power *sensor nodes*, each containing some actuators, sensing devices, and a wireless transceiver [2]. These sensor nodes are massively deployed in a region of interest to gather environmental information, which is to be reported to remote sinks. It thus provides an inexpensive and powerful means to monitor the physical environment. The functionalities of a remote sink are to collect data from sensor nodes and to transmit queries or commands to sensor nodes. However, a sensor network differs from an ad hoc network in several aspects. First, a node in an ad hoc network is usually a laptop or a PDA, while a sensor node is typically a smaller device with a low-speed processor, limited memory, and a short-range transceiver. Thus, protocols/algorithms running on a sensor node should be simple. Second, since a sensor node is typically powered by batteries, energy consumption is a more critical design issue in a sensor network. Third, the communication patterns in sensor networks may differ from those in ad hoc networks. Fourth, sensor nodes are relatively less mobile than those in ad hoc networks.

Since the transmission between two out-of-range nodes has to rely on relay nodes, many routing protocols have been proposed for ad hoc and sensor networks. However, most of them assume that other nodes are trustable and thus do not consider the security and attack issues. To assure a source node of finding a routing path to its destination, most routing protocols [3]–[5] attempt to invite all available nodes in the network to participate in the routing mechanism. This provides many opportunities for attackers to break the network.

Although many security solutions have been proposed for wire-line networks, they may not be directly applied to ad hoc and sensor networks. The difficulties and challenges are listed as follows [1]:

- Since the transmission medium is open, ad hoc and sensor networks are more vulnerable to physical security threats than wire-line networks. Possible physical attacks range from passive eavesdropping to active interference. Besides, nodes in the network without adequate protection may be captured, compromised, and hijacked by the adversary. In this case, the authentication information is disclosed and the adversary can use these hijacked nodes to disrupt the network.
- Authentication relies on public key cryptography or certification authorities may be difficult to accomplish in an ad hoc network since such a network does not have any centralized network infrastructure. Besides, cryptography that needs complicated computation or large memory space cannot be performed on sensor nodes.
- Due to the lack of centralized network infrastructure, a node in an ad hoc network has to detect the possible attacks by itself or cooperates with its neighbors to find out potential attackers. The effect is usually limited because nodes can only obtain local information. Besides, an attacker may claim other legitimate nodes to be illegal. For sensor networks, some centralized intrusion detection schemes may be applied in the remote sink. However, this will drastically increase traffic between sensor nodes and the remote sink.
- Any security solution with a static configuration may not be suitable for ad hoc networks since nodes have

The authors are with the Department of Computer Science, National Chiao-Tung University, Hsin-Chu, 30010, Taiwan.
E-mail: {wangyc, yctseng}@cs.nctu.edu.tw

mobility and the network topology may change frequently. Nodes have to continuously detect possible attackers since their neighbors are not fixed. Similarly, for a sensor network, a malicious node with mobility can roam in the network and attack different parts of the network.
- Since nodes normally rely on batteries to provide energy, an attacker can frequently flood fake or dummy messages to exhaust other nodes' energies. An attacker can disguise its packets as normal ones or replay other nodes' packets to waste energy of normal nodes.

In this chapter, we present a survey of possible attacks and existing defense schemes of routing mechanisms in ad hoc and sensor networks. Section 2 briefly introduces some routing protocols in ad hoc and sensor networks. Section 3 discusses several possible attacks to these routing protocols. A survey of defense schemes is presented in Section 4. Conclusions are drawn in Section 5.

2 Routing Protocols in Ad Hoc and Sensor Networks

To support multi-hop communication in ad hoc networks, many routing protocols have been designed [6]. These protocols can be classified into *table-driven* (or *proactive*) and *on-demand* (or *reactive*) ones. In table-driven routing protocols, nodes need to exchange routing information regardless of communication requests. Such protocols attempt to maintain consistent, up-to-date routing information for each node to reach every other node in the network. Therefore, they require each node to maintain one or more tables to store routing information. Besides, any change of network topology may need to be propagated to the whole network to maintain a consistent network view. The *destination-sequenced distance-vector (DSDV)* routing protocol [4] is a representative example. In DSDV, each node maintains a *forwarding table* in which each entry contains a destination address, the next hop to the destination, the number of hops to the destination, and a sequence number. Nodes will periodically exchange the contents of their forwarding tables. To relay packets, an intermediate node simply has to look its forwarding table to find out the next hop to the destination. Other table-driven protocols include the *topology broadcast based on reversed path forwarding (TBRPF)* routing protocol [7], the *optimized link state routing (OLSR)* protocol [8], and the *fisheye state routing (FSR)* protocol [9].

On-demand protocols are more popular for ad hoc routing [6]. The main feature of such protocols is that nodes exchange routing information only when there are communications awaiting. This can reduce routing overhead compared to the table-driven protocols. When a node attempts to communicate with another node, it floods a *route request (RREQ)* packet in the network. Nodes receiving the RREQ packet will send back a *route reply (RREP)* packet to the source if they know how to route to the destination; otherwise, they forward the RREQ packet to other nodes. There are several variations of on-demand routing. The *dynamic source routing (DSR)* protocol [3] is derived based on *source routing*. On receiving an RREQ, an intermediate node inserts its address into the packet and rebroadcasts it. Therefore, the destination will have the entire path from the source to itself. The destination then responds to an RREP with the entire routing path to the source.

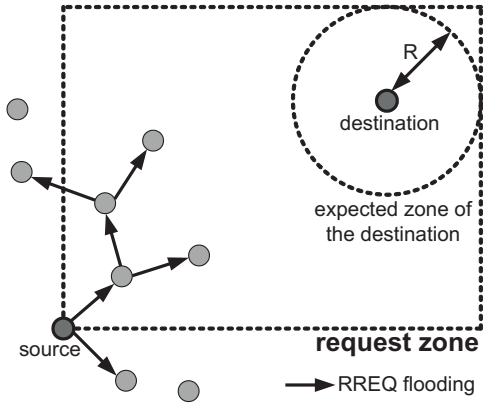

Fig. 1: Limited flooding of RREQ packets in LAR. R is the maximum distance that the destination may move from its previous known location.

The whole routing path is indicated in each data packet initiated by the source. The *ad-hoc on-demand distance vector routing (AODV)* protocol [5] also finds routes by flooding RREQ packets. However, unlike DSR, each intermediate node maintains a forwarding table to indicate the next node leading to the destination so that the source does not need to insert the whole routing path in data packets. Other on-demand protocols include the *lightweight mobile routing (LMR)* protocol [10], the *temporally order routing algorithms (TORA)* [11], and the *associativity-based routing (ARB)* protocol [12].

Several protocols assume that each node is equipped with a *global positioning system (GPS)* device to provide the node's geographic location. By knowing the approximate location of the destination, the source can limit the flooding area of its RREQ packet. For example, the *location-aided routing (LAR)* protocol [13] creates a *request zone*, which contains the expected zone of the destination, as shown in Fig. 1. Only intermediate nodes inside the request zone will forward the RREQ packet, so the overhead of flooding can be reduced. Other examples using geographic information include the *distance routing effect algorithm for mobility (DREAM)* [14], the *greedy perimeter stateless routing (GPSR)* protocol [15], and the *geographic addressing and routing (GeoCast)* protocol [16].

Routing protocols designed for ad hoc networks typically support routing between any pair of nodes. However, sensor networks have more specialized communication patterns. There are three common categories [17]:

- **Many to one**: Multiple or all sensor nodes report their collected data to a remote sink.
- **One to many**: The remote sink multicasts or broadcasts a query or a command to multiple or all sensor nodes.
- **Local communication**: Neighboring sensor nodes send localized messages to each other.

The traditional ad hoc routing protocols may not be suitable for sensor networks. In particular, a sensor network usually forms a simple spanning tree rooted at the remote sink for the routing purpose.

3 Attacks on Routing Mechanisms

Most routing protocols designed for ad hoc and sensor networks assume that nodes in the network do not misbehave. This provides many opportunities for malicious nodes to attack and disrupt the routing mechanism. These

attacks can be classified into *passive* and *active* ones [1], [18]. A passive attack typically involves only eavesdropping the routing traffic to discover valuable information. It is usually difficult to detect passive attacks since such attacks do not destroy the operations of routing protocols. Two possible solutions can be used to restrain eavesdropping. One is to adopt encryption or other security mechanisms in the application layer [17]. The other solution is to transmit parts of a message over multiple disjoint paths and reassemble them at the destination [19].

An active attack may attempt to disrupt the routing mechanism, intentionally modify the routing messages, gain authentication or authorization, or even control the whole network by generating false packets into the network or by modifying or dropping legitimate packets sent by other nodes. Active attacks can be further categorized into *external* and *internal* attacks. An external attack is generated by malicious nodes which do not belong to the network. An internal attack is caused by compromised or hijacked nodes that were formerly legitimate. Security mechanisms which rely on authentication or encryption may not handle internal attacks since these compromised nodes also have the keys and thus are treated as authorized parties in the network.

For sensor networks, [17] classifies the attackers into *sensor-class* and *laptop-class* attackers according to their capabilities. A sensor-class attacker has the similar capability of other legitimate nodes in the network, so the attacks caused by these attackers are less complicated. A laptop-class attacker has more powerful computing devices, more battery power, larger memory, and even high-power radio transmitters. Attacks caused by laptop-class attackers can be different from those in ad hoc networks. For example, a laptop-class attacker may be able to jam a large range of or even the entire sensor network, while an attacker in an ad hoc network is only able to jam the radio links in its vicinity.

Below, we introduce several possible active attacks in ad hoc and sensor networks. We divide these attacks into three classes: attacks on route discovery process, attacks on route selection process, and attacks after establishing routing paths.

3.1 Attacks on Route Discovery Process

Such attacks attempt to prevent other legitimate nodes from establishing routing paths by sending fake routing information. Moreover, a malicious node can send excessive route request messages to exhaust the network bandwidth. The former is to provide fake information to spoof the route discovery process, while the latter is to overly use the route discovery process. Both of them attempt to cause *denial of service (DoS)*.

3.1.1 Fake Routing Information

A straightforward attack against a routing protocol is to provide fake routing information during its route discovery phase. For table-driven routing protocols, a malicious node can interfere with other legitimate nodes by advertising incorrect routing information to invalidate their routing tables (or forwarding tables) [18]. For on-demand routing protocols, a malicious node can reply a non-existing route to the source or alter the addresses in an RREQ packet to spoof the destination. It can also modify an RREP packet to cause invalid route to the source. A malicious node can also silently drop all RREQ and RREP packets passing through it to refuse participating in the route discovery process.

3.1.2 Rushing Attacks

A *rushing attack* [20] mainly intends to break the route discovery process of an on-demand routing protocol. In an on-demand routing protocol, a node that wants to establish a route to a destination floods the network with RREQ packets. To reduce the overhead of flooding, each node typically forwards only the first-arrived RREQ packet originated from the source. Such property is exploited by a rushing attacker. In particular, after an attacker receives the RREQ packet from the source, it propagates its modified RREQ packet to other intermediate nodes *before* legitimate RREQ packets reach them. When the legitimate RREQ packets are received, they may be discarded. In this case, the attacker can prevent a valid path from being established or increase its chance of being selected as part of a routing path.

Note that a rushing attack can succeed only if the attacker can send its modified RREQ packets to other nodes before these nodes receive the legitimate ones. To achieve this, the attacker can reduce delays at either the MAC or routing layer to send out its packets as fast as possible. For example, the attacker can select a smaller number to run the backoff mechanism at the MAC layer after collision. A more powerful rushing attacker can utilize a wormhole [21] (refer to Section 3.2.3) to rush packets.

3.1.3 RREQ Flood Attacks

The fundamental mechanism of the route discovery process of an on-demand routing protocol is to flood RREQ packets in the network. An *RREQ flood attacker* abuses this mechanism to result in a DoS attack. In particular, the attacker can generate frequent unnecessary or false RREQ packets to make the network resources unavailable to other legitimate nodes. Besides, the attacker can make other nodes' routing tables overflow by flooding excessive RREQ packets with different or even non-existent destinations [1], so that creation of new routes by other legitimate nodes will be prohibited.

3.2 Attacks on Route Selection Process

This type of attacks attempts to increase the chance that malicious nodes are selected by other legitimate nodes as part of their routes. After establishing a route through itself, the attacker can overhear the transmitted messages or combine the attacks discussed in Section 3.3 to disrupt the network. Fig. 2 shows four possible attacks of this type, including HELLO flood, sinkhole, wormhole, and Sybil attacks.

3.2.1 HELLO Flood Attacks

Many protocols require nodes to broadcast localized HELLO messages to announce themselves to their neighbors. A node receiving such a message will assume that the sender is its one-hop neighbor. However, this assumption may be violated when there are laptop-class attackers in a sensor network. A laptop-class attacker can utilize a large transmission power to broadcast its HELLO message to

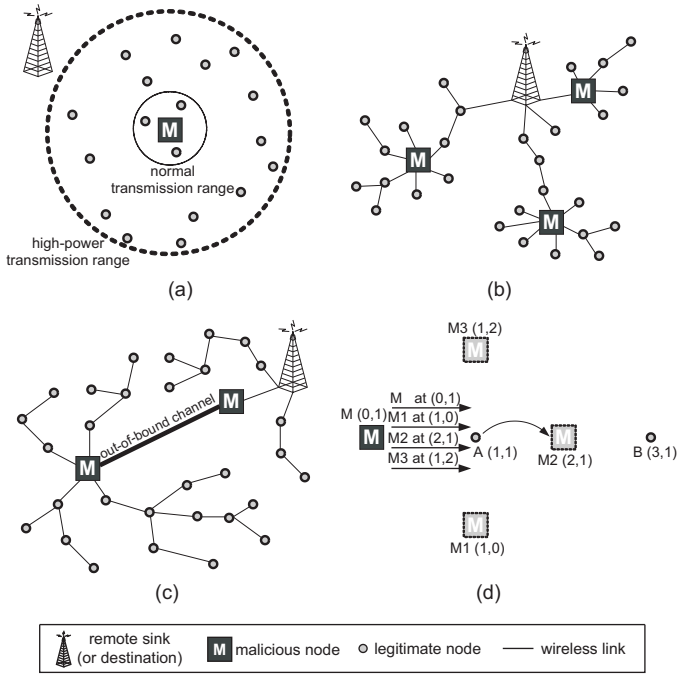

Fig. 2: Attacks on route selection processes. (a) HELLO flood attack. (b) Sinkhole attack. (c) Wormhole attack. (d) Sybil attack.

cover a large range of sensors. The receiving nodes will be convinced that the attacker is their one-hop neighbor, as shown in Fig. 2(a). Such an attack is called a *HELLO flood attack* [17].

Protocols that rely on localized information exchange between neighboring nodes for topology maintenance or flow control are vulnerable to the HELLO flood attack. Besides, the attacker can advertise a higher quality or shorter route to the destination. This may even cause other nodes to follow the same route to the destination. However, most messages from legitimate nodes may not be sent to the attacker since these nodes have smaller transmission ranges.

3.2.2 Sinkhole Attacks

The objective of a *sinkhole attack* [17] is to attract all neighboring nodes of the attacker to establish routes through the attacker, as shown in Fig. 2(b). In this scenario, all traffic from a particular area will flow through the attacker, thus creating a metaphorical sinkhole with the attacker at the center. Sinkhole attacks typically work by making a malicious node look especially attractive to surrounding nodes with respect to the routing algorithm. For example, an attacker can advertise an extreme high quality route to some destinations, or even spoof these surrounding nodes that the attacker itself is neighboring to the destination.

Unlike the HELLO flood attack, the sinkhole attacker usually utilizes a normal transmission power. Thus, a sinkhole attacker may affect only part of the network, and both ad hoc and sensor networks are vulnerable to such attacks.

3.2.3 Wormhole Attacks

Multiple malicious nodes can cooperate to generate attacks against the network. One of the representative examples is the *wormhole attack* [21]. In a wormhole attack, two distant malicious nodes utilize an out-of-bound channel available only to the attackers to tunnel messages received by one side to another side. Specifically, packets transmitted through the wormhole tunnel usually have lower latency than those packets sent between the same pair of nodes over normal multi-hop routing. This will result in a false appearance that routing through these two malicious nodes is a better choice. Therefore, neighboring nodes will select the malicious nodes as the intermediate nodes in their routes. Besides, wormholes can create a fake network topology by relaying packets between two distant nodes. In this case, these two distant nodes may be considered as neighbors to each other.

A sensor network is more vulnerable to the wormhole attack due to the following two reasons. First, laptop-class attackers can utilize out-of-bound channels to create low-latency, high-bandwidth tunnels more easily since they have more powerful transceivers compared to other sensor nodes. Second, an adversary situated close to the remote sink can control a lot of routing by creating a well-placed wormhole. Fig. 2(c) gives an example, where more than half of the sensor nodes will be guided to the wormhole tunnel.

3.2.4 Sybil Attacks

In a *sybil attack* [22], a malicious node can disguise itself as multiple different nodes by advertising multiple identities to its neighbors. Since the sybil attacker can create many fake nodes, it thus can increase the probability that the malicious node is selected by other nodes as part of their routing paths. Besides, the sybil attack can significantly reduce the effectiveness of fault-tolerance schemes such as multi-path routing [23], [24] because other nodes will treat the fake nodes generated by the malicious node as different nodes and establish different routes through the malicious node.

A sybil attacker can also spoof nodes using a geographic routing protocol, such as the GRID routing protocol [25]. Fig. 2(d) gives an example. A malicious node M at actual location $(0, 1)$ advertises not only its true identity and location, but also three forged nodes $M1$, $M2$, and $M3$ at locations $(1, 0)$, $(2, 1)$, and $(1, 2)$, respectively. After receiving these advertisements, if a node A located at $(1, 1)$ wants to transmit packets to another node B located at $(3, 1)$, it will select the fake node $M2$ (which locates at $(2, 1)$) as its forwarding node. Since the malicious node M is a neighbor of node A, it can overhear the transmission and takes a further action.

3.3 Attacks after Establishing Routing Paths

Once a source node establishes a route through a malicious node, the malicious node can unscrupulously drop the data packets from the source or modify the contents of packets if encryption is not applied. A malicious node can use the attacks discussed in Section 3.2 to insert itself to the routing path. Besides, an attacker can play as a source node to establish routes to other nodes, and then send dummy messages to exhaust their energies and the network bandwidth.

3.3.1 Blackhole Attacks

Multi-hop communications must rely on the cooperation of participating nodes to forward the received messages. In a *blackhole attack* [1], malicious nodes violate such an

assumption by dropping all received messages from the source to prevent these messages from being propagated any further.

However, most routing protocols have the *route maintenance* mechanism such that if a node in the path finds that its next-hop neighbor no longer propagates its data packets, it will notify the source to recreate another routing path. In this case, a blackhole attack is trivially defended. A more cunning form of this attack is that the attacker selectively forwards packets [17] to cheat the source that this route is still alive.

3.3.2 Spam Attacks

Like spam mails, a spam attacker [26] frequently generates a large number of unsolicited and useless messages to the network. These messages will waste the network bandwidth and the energies of nodes that receive or forward these messages. The spam attacks are more jeopardous to sensor networks. This is because in a sensor network, environmental data collected by sensors will all be transmitted to the remote sink. A spam attacker can thus generate a lot of dummy messages to the sink to consume energies of relayed sensors, especially those closed to the sink. Once the energies of these sensors are exhausted, the sink will never receive data from the sensor network. In this case, the whole sensor network is destroyed, even though most of sensors are alive.

3.4 Summary of Attacks

Here we compare attacks to ad hoc and sensor networks in Table 1. Recall that nodes in an ad hoc network have similar capability, while there can be powerful laptop-class attackers in a sensor network. From Table 1, we can observe that the fake routing information, sinkhole attacks, sybil attacks, and blackhole attacks affect both networks. Since a sensor network usually forms a spanning tree rooted at the remote sink for the routing purpose, attacks against on-demand routing protocols, such as rushing attacks and RREQ flood attacks, may not affect sensor networks. The HELLO flood attacks can only affect sensor networks because in an ad hoc network, a malicious node may not be able to generate a large power to cover most nodes in the network. The wormhole attack is based on the assumption that two malicious nodes can utilize an out-of-bound channel to communicate with each other. Besides, they have to provide a low-latency, high-bandwidth, and long-distance link to tunnel packets between them. These assumptions are sometimes difficult to accomplish in an ad hoc network. Finally, the spam attack can partially affect an ad hoc network because all nodes in an ad hoc network can become the possible destination nodes. Such an attack can succeed in a sensor network because the remote sink is usually the only destination.

Table 2 compares the attacks on table-driven and on-demand routing strategies. Most attacks will affect both routing strategies, except rushing attacks and RREQ flood attacks, which are only against the on-demand routing protocols.

attack	ad hoc network	sensor network
fake routing information	√	√
rushing attacks	√	
RREQ flood attacks	√	
HELLO flood attacks		√
sinkhole attacks	√	√
wormhole attacks	partially	√
sybil attacks	√	√
blackhole attacks	√	√
spam attacks	partially	√

TABLE 1: Comparison of attacks. A √ means that the corresponding network is vulnerable to such an attack.

attack	table-driven protocol	on-demand protocol
fake routing information	√	√
rushing attacks		√
RREQ flood attacks		√
HELLO flood attacks	√	√
sinkhole attacks	√	√
wormhole attacks	√	√
sybil attacks	√	√

TABLE 2: Comparison of attacks to table-driven and on-demand routing strategies.

4 Defense Schemes

To avoid various attacks on ad hoc and sensor networks, many defense schemes have been designed. In this section, we give a survey on these defense schemes against the attacks discussed in Section 3.

4.1 Defenses against Fake Routing Information and RREQ Flood Attacks

To prevent an external attacker from generating fake routing information or RREQ flood attacks against the network, one possible solution is to apply security mechanisms such as authentication to the routing protocol [27]–[29]. Nodes in the network share keys to authenticate their data packets and routing control messages such as RREQ and RREP. Since an external attacker does not have the keys to authenticate its packets, all its fake routing information and dummy RREQ packets will not be accepted by other legitimate nodes, so the attacks can be defended.

The *localized encryption and authentication protocol (LEAP)* [30] proposes a key management protocol for sensor networks, where different types of key managements are utilized for different security requirements. In LEAP, four types of keys are established for sensor nodes:

1) **Individual key**: Each sensor node shares a unique key with the remote sink. This key is used for secure communication between the sensor node and the remote sink.
2) **Group key**: The group key is globally shared among all sensor nodes and the remote sink. This key is used by the remote sink to encrypt broadcast messages.
3) **Cluster key**: A cluster key is shared by a sensor node and *all* its neighbors. It is mainly used for securing locally broadcast messages, such as routing control messages.
4) **Pairwise shared key**: Every sensor node shares a pairwise key with *each* of its neighbors for secure communication.

LEAP can prevent an *internal* attacker from disrupting the whole network, because a sensor node does not have a network-wide authentication key. (Note that the group key is only used to encrypt messages from the remote sink and it cannot be used for authentication.) A hijacked node can

only have local keys shared with its neighbors so that it may only affect its neighbors.

4.2 Defenses against Rushing Attacks

A rushing attack is caused by a malicious node rapidly transmitting fake RREQ packets to invalidate the legitimate ones. To defend such attacks, [20] proposes the *randomized RREQ forwarding* and the *secure neighbor detection* schemes. In the randomized RREQ forwarding scheme, each node collects a number of RREQ packets and then randomly selects one of them to forward. In this case, a malicious node can take no advantage by transmitting RREQ packets very quickly, since its neighbors will wait for other legitimate RREQ packets. However, the fake RREQ packet may still be selected, so the secure neighbor detection scheme is invoked to determine whether the sender of RREQ packet is a legitimate node. The secure neighbor detection scheme utilizes authenticated messages exchanged between two nodes to verify their identities and legitimacy.

4.3 Defenses against HELLO Flood Attacks

Since a HELLO flood attack is caused by a malicious node utilizing a large transmission power to generate asymmetric links between it and other legitimate nodes, one intuitive defense against such an attack is to verify the *bi-directionality* of a link between two neighboring nodes. The LEAP discussed in Section 4.1 takes this strategy. In LEAP, when a node u attempts to discover its neighbors, it broadcasts a HELLO message and waits for each neighbor v to respond with its identity. The response from v is authenticated by a *message authentication code* so that u can verify the response message. Note that node u will consider node v as its neighbor only if it receives a correct response from v. In this case, HELLO flood attacks will fail because in LEAP a node will only consider another node as its neighbor if there is two-way communication between them.

Another solution for sensor networks is to utilize the remote sink as a trusted third-party to help two sensor nodes verify each other [17]. Specifically, each sensor node in the network share a unique symmetric key with the remote sink. Two nodes u and v can then verify each other's identity and establish a shared key through the remote sink's authentication, as shown in Fig. 3. A pair of neighboring nodes can thus use the resulting key to communicate with each other. To avoid a mobile attacker roaming around a stationary network or using the HELLO flood attack to establish shared keys with too many sensor nodes, the remote sink can reasonably limit the number of verified neighbors for each sensor node and reject a request when a node exceeds the limitation.

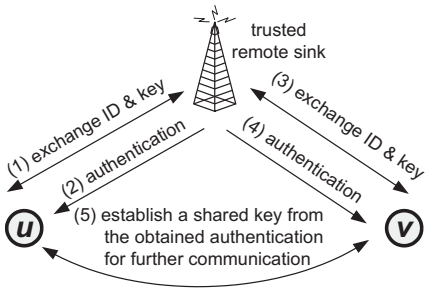

Fig. 3: Trusted third-party authentication in a sensor network.

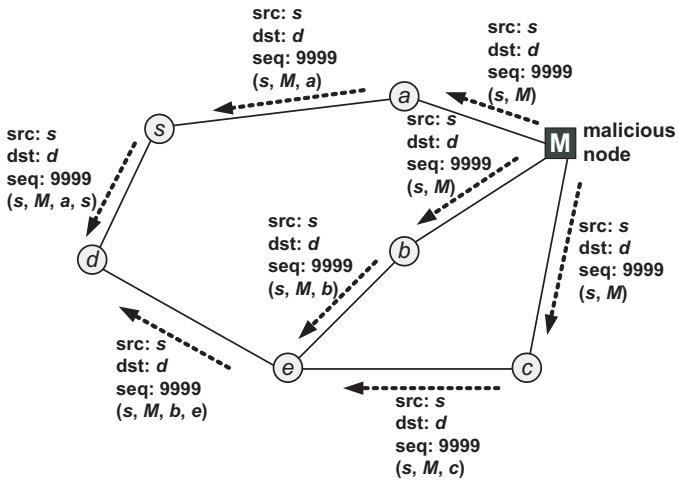

Fig. 4: An example of sinkhole attacks in DSR.

4.4 Defenses against Sinkhole Attacks

Recall that the intention of a sinkhole attacker is to attract all its neighbors to establish routes through it. To achieve this, [31] provides a possible trick for the attacker by generating bogus RREQ packets in the DSR protocol. A malicious node can broadcast bogus RREQ packets with properly selected sources and destinations, very high sequence numbers, and route records which specify one-hop routes from the sources to the malicious node itself. Such RREQ packets will make a false appearance to other nodes that the malicious node is an immediate neighbor to the source, and the information is the freshest since the sequence numbers are quite large. So these nodes receiving the bogus RREQ packets will update the fake routing information into their route caches. Fig. 4 gives an example, where the malicious node M broadcasts a bogus RREQ packet with source as s and destination as d. After receiving the bogus RREQ packet, nodes a, b, and c attempt to learn routes from the route record. By reversing the path recorded in the RREQ packet, they will falsely conclude that node M has a one-hop route to node s, and replace the old route by this fake route in their caches. Note that even if node a is a neighbor of node s, it will add this route to its cache because the sequence number in the bogus RREQ packet is larger than the sequence number for any route that node a previously learned to node s. The malicious node M can also fabricate RREQ packets with different combinations of sources and destinations. By repeating this attack periodically, all neighbors will believe that node M has the shortest path to every other node.

To detect such attacks, the *sinkhole intrusion detection system (SIDS)* [31] utilizes three indicators to determine whether there are sinkhole attackers in the network:

1) **Discontinuity of sequence numbers**: In theory, the sequence numbers of packets originated from a node should strictly increase in DSR. However, a sinkhole attacker attempts to use a very large sequence number to update the contents of other nodes' route caches. Therefore, a node can monitor the sequence numbers of receiving RREQ packets and pay attention to those that are not strictly increasing (or unusually large).
2) **Ratio of verified RREQ packets**: When a node initiates a RREQ packet, the source address should be

its own address. Such RREQ packets can be verified by the source's neighbors. However, a sinkhole attacker initiates RREQ packets with different sources and periodically broadcast these bogus RREQ packets to the network. Therefore, a lower ratio of verified RREQ packets in the overall network may indicate the presence of a sinkhole attacker.

3) **Ratio of routes through a particular node**: Since a sinkhole attack causes nodes in the network to add routes through the attacker, nodes can determine the existence of a sinkhole attacker by checking their routing caches. If a node finds that most routes in its cache are through a particular node, it will suspect that this node is a potential attacker.

4.5 Defenses against Wormhole Attacks

Two distant malicious nodes can use an out-of-bound channel to result in a wormhole attack by tunneling packets through the channel. In [21], the concept of *packet leash* is introduced to defend against such attacks. A *leash* is the information added in a packet to restrict the packet's maximum allowed transmission distance. Reference [21] proposes two leashes, *geographical leashes* and *temporal leashes*. The geographical leash guarantees that the receiver of the packet is within a certain distance from the sender, while the temporal leash ensures that the packet has an upper bound on its lifetime, which restricts its maximum traveling distance.

To use temporal leashes, all nodes in the network must have tightly time synchronization. Specifically, the maximum difference between any two nodes' clocks is bounded by Δ, and this value should be known by all nodes. When a node s transmits a packet, it includes its current time t_s and utilizes authentication to protect this packet. When the node r receives this packet at time t_r, it can determine whether there are wormhole attackers in its route, based on the claimed transmission time t_s and the propagation speed v_c. In particular, node s can embed an expiration time $t_{\text{expire}} = t_s + \frac{L}{v_c} - \Delta$ in its packet, where L is the maximum distance that the packet is allowed to travel. When node r receives the packet, it will check if $t_r < t_{\text{expire}}$. If so, node r will accept the packet. Otherwise, the packet is discarded and this indicates that the earlier RREQ has been tunneled.

4.6 Defenses against Sybil Attacks

In a sybil attack, a malicious node disguises as different nodes by impersonating other nodes or claiming fake identities to its neighbors. In [22], two major schemes, *radio resource testing* and *random key predistribution*, are designed to defend against sybil attacks. The radio resource testing scheme is based on the assumption that each physical node (including the attacker) has only one radio and cannot simultaneously send or receive on more than one channel. A node that attempts to check whether there are fake nodes pretended by a sybil attacker in its neighborhood can assign each of its neighbors a different channel to broadcast messages. The node then randomly selects a channel to listen. If a message can be received, the neighbor is indicated as a legitimate node. Otherwise, the neighbor is treated as a fake node pretended by the sybil attacker.

The random key predistribution scheme is derived from the *key pool* scheme [32], [33]. The key pool scheme randomly assigns k keys to each node from a pool of m keys. If two nodes share q common keys, they can establish a secure link. However, the key pool scheme cannot defend against the sybil attack since if an attacker compromises multiple nodes, it can use all possible combinations of the compromised keys to generate new identities. The random key predistribution scheme solves this problem by using a pseudo random hash function to assign keys and validate the identity of a node. Specifically, let $\Omega(ID) = \{K_{\beta_1}, K_{\beta_2}, \cdots, K_{\beta_k}\}$ be the set of keys assigned to the node whose identity is ID, where $\beta_i = PRF_{H(ID)}(i)$ is the index of its ith key in the key pool, H is an one-way hash function, and PRF is a pseudo random function. Since the indices of keys assigned to a node are determined by the hash value of its identity, and it is very hard to inverse the hash function to obtain the original identity, a sybil attacker cannot just collect a set of keys and claim fake identities from these keys.

4.7 Defenses against Blackhole Attacks

Blackhole attacks occur when a malicious node intentionally drops all routing messages after establishing the routing path. An attacker can even selectively forward and drop packets to avoid triggering the route maintenance mechanism at a source node to select another route. To defend against such attacks, [34] proposes a *watchdog* scheme to identify potential malicious nodes and a *pathrater* scheme to help routing protocols avoid these nodes. Using the DSR protocol, the watchdog works on the assumption that a node can overhear the packets transmitted by its neighbors. The idea of watchdog is that when a node A transmits a data packet to its next-hop neighbor B, node A will overhear the transmission from B to check whether node B has really transmitted the data packet to B's next-hop neighbor. The watchdog at each node maintains a counter to record the misbehavior of each of its next-hop neighbors. Once the value of the counter exceeds a threshold, the watchdog will infer that its next-hop neighbor may be a malicious node and reports to the source. The path-rater scheme, combined with the watchdog, helps a source node select the most reliable route. Each node assigns a non-negative rating to every normal node that it knows in the network (including itself), and a highly negative rating to each malicious node. The overall rating of a routing path is the average rating of nodes on that route. The source node then selects the routing path with the highest rating to forward its packets. Note that since a malicious node is assigned a highly negative rating, a routing path with a negative rating indicates the presence of malicious nodes. Therefore, the path-rater scheme can help a node select a route without malicious nodes.

Another solution for blackhole attacks is proposed in [35], which utilizes the following four mechanisms:

- **Source routing**: The source specifies in each data packet the sequence of nodes that the packet has to traverse.
- **Destination acknowledgements**: The destination sends an *acknowledgement (ACK)* to the source along the same route whenever it receives a data packet.

- **Timeouts**: The source and each intermediate node set a timer for each data packet, during which they expect to receive an ACK from the destination or a *fault announcement (FA)* from other intermediate nodes.
- **Fault announcements**: When the timer expires, the node generates a FA and propagates it to the source.

All data, ACK, and FA messages are authenticated by a message authentication code so that these messages cannot be modified or fabricated by a malicious node. Note that the source can detect the presence of a potential blackhole attacker when it receives an FA message and thus select another route to forward its packets.

4.8 Defenses against Spam Attacks

Spam attacks are caused by malicious nodes generating frequent dummy messages to specified targets in the network. A sensor network is more vulnerable to such attacks since the remote sink is usually the only target to be attacked. To defend spam attacks, the *detect and defend spam (DADS)* scheme [26] proposes a concept of *quarantine regions* to isolate spam attackers. In DADS, the remote sink is responsible for detecting whether there are spam attacks in the network. The remote sink can detect spam attacks by the three methods. The first one is to filter incoming messages according to their contents and detect nodes that send faulty message frequently. The second method uses the frequencies of messages sent by the sensor nodes in the same region. The third method is to observe the packet generation rate of the overall sensor network. Since an attacker may have mobility and can change its identity to spoof the remote sink, the last method is suggested in DADS. In particular, when the number of data packets arriving at the remote sink exceeds an acceptable level, the remote sink broadcasts an alarm message, called *defend against spam (DAS)*, to the network.

The basic idea of DADS is to quarantine a spam attacker by its one-hop neighbors. When a sensor node u receives a DAS message, it starts a timer t_q and only allows to relay authenticated messages before the timer expires. If node u receives an unauthenticated message from a neighbor, it asks the neighbor to resend an authenticated message. If the latter fails in authentication, node u determines that it is inside a quarantine region and will not relay any data packet unless it is successfully authenticated. Besides, node u will transmit its own messages with authentication. Fig. 5 gives an example. Nodes a, b, c, and d are inside a quarantine region because they detect that node M fails in authentication. Then they will relay and transmit only authenticated messages. All other nodes, except node e, can still transmit unauthenticated messages to the sink. Note that node e has to send authenticated messages since node c only accepts such packets.

DADS uses the *hash-based message authentication code (HMAC)* [36], which utilizes a cryptographic one-way hash function such as MD5 [37], for message authentication. To save the authentication overhead, when a sensor node inside a quarantined region does not detect any unsuccessful authentication attempt during a period of time t_q, it switches back to the normal mode to cancel the quarantine region.

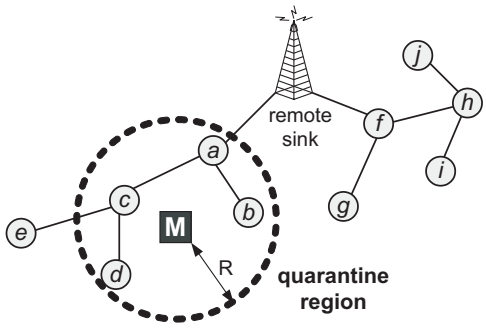

Fig. 5: An example of DADS, where R is the transmission range of the malicious node M.

5 Conclusions

Communications in ad hoc and sensor networks are much relied on multi-hop transmission. However, most routing protocols do not address the security issue and thus trust all nodes in the network. This provides many chances for attackers to disrupt the network. This motivates many researchers to find out possible attacks and develop their countermeasures. This chapter provides a comprehensive survey on current research in attacks and defenses of routing mechanisms in ad hoc and sensor networks. Various representative attacks and their defense schemes are discussed in the chapter.

References

[1] H. Deng, W. Li, and D.P. Agrawal, "Routing security in wireless ad hoc networks," *IEEE Comm. Magazine*, vol. 40, no. 10, pp. 70–75, 2002.
[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Comm. Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
[3] D.B. Johnson and D.A. Malts, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers*, pp. 153–181, 1996.
[4] C.E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *Proc. ACM Conf. Comm. Architectures, Protocols and Applications*, pp. 234–244, 1994.
[5] C.E. Perkins and E.M. Royer, "Ad-hoc on-demand distance vector routing," *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.
[6] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.
[7] B. Bellur and R. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," *Proc. IEEE INFOCOM*, pp. 178–186, 1999.
[8] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L.Viennot, "Optimized link state routing protocol for ad hoc networks," *Proc. IEEE Int'l Multi Topic Conf.*, pp. 62–68, 2001.
[9] G. Pei, M. Gerla, and T.W. Chen, "Fisheye state routing: a routing scheme for ad hoc wireless networks," *Proc. IEEE Int'l Conf. Comm.*, pp. 18–22, 2000.
[10] M.S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *Wireless Networks*, vol. 1, no. 1, pp. 61–81, 1995.
[11] V.D. Park and M.S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *Proc. IEEE INFOCOM*, pp. 1405–1413, 1997.
[12] C.K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Comm.*, vol. 4, no. 2, pp. 103–139, 1997.
[13] Y.B. Ko and N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.
[14] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," *Proc. ACM Int'l Conf. Mobile Computing and Networking*, pp. 76–84, 1998.
[15] B. Karp and H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking*, pp. 243–254, 2000.
[16] J.C. Navas and T. Imielinski, "Geocast – geographic addressing and routing," *Proc. ACM Int'l Conf. Mobile Computing and Networking*, pp. 66–76, 1997.

[17] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Proc. IEEE Int'l Workshop Sensor Network Protocols and Applications*, pp. 113–127, 2003.
[18] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Comm.*, vol. 11, no. 1, pp. 48–60, 2004.
[19] H. Chan and A. Perrig, "Security and privacy in sensor networks," *IEEE Computer*, vol. 36, no. 10, pp. 103–105, 2003.
[20] Y.C. Hu, A. Perrig, and D.B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," *Proc. ACM Workshop Wireless Security*, pp. 30–40, 2003.
[21] Y.C. Hu, A. Perrig, and D.B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," *Proc. IEEE INFOCOM*, pp. 1976–1986, 2003.
[22] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," *Proc. Int'l Symp. Information Processing in Sensor Networks*, pp. 259–268, 2004.
[23] J. Chen, P. Druschel, and D. Subramanian, "An efficient multipath forwarding method," *Proc. IEEE INFOCOM*, pp. 1418–1425, 1998.
[24] K. Ishida, Y. Kakuda, and T. Kikuno, "A routing protocol for finding two node-disjoint paths in computer networks," *Proc. IEEE Int'l Conf. Network Protocols*, pp. 340–347, 1995.
[25] W.H. Liao, Y.C. Tseng, and J.P. Sheu, "GRID: A fully location-aware routing protocol for mobile ad hoc networks," *Telecomm. Systems*, vol. 18, no. 1, pp. 37–60, 2001.
[26] S. Sancak, E. Cayirci, V. Coskun, and A. Levi, "Sensor wars: detecting and defending against spam attacks in wireless sensor networks," *Proc. IEEE Int'l Conf. Comm.*, pp. 3668–3672, 2004.
[27] Y.C. Hu, D.B. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," *Proc. IEEE Workshop Mobile Computing Systems and Applications*, 2002.
[28] Y.C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking*, pp. 12–23, 2002.
[29] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," *Proc. Int'l Symp. Computers and Comm.*, pp. 567–574, 2002.
[30] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," *Proc. ACM Conf. Computer and Comm. Security*, 2003.
[31] B.J. Culpepper and H.C. Tseng, "Sinkhole intrusion indicators in DSR MANETs," *Proc. Int'l Conf. Broadband Networks*, pp. 681–688, 2004.
[32] A.P.H. Chan and D. Song, "Random key predistribution schemes for sensor networks," *Proc. IEEE Symp. Security and Privacy*, pp. 197–213, 2003.
[33] L. Eschenauer and V.D. Gligor, "A key-management scheme for distributed sensor networks," *Proc. ACM Conf. Computer and Comm. Security*, pp. 41–47, 2002.
[34] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking*, pp. 255–265, 2000.
[35] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, "Highly secure and efficient routing," *Proc. IEEE INFOCOM*, pp. 197–208, 2004.
[36] H. Krawczyk, M. Bellare, and R. Canetti, "RFC 2104 – HMAC: Keyed-Hashing for Message Authentication," 1997.
[37] R.L. Rivest, "RFC 1321 - The MD5 Message-Digest Algorithm," 1992.