NETWORK SECURITY
Scott Huang, David MacCallum, and Ding Zhu Du(Eds.) pp. – - –

Attacks and Countermeasures in Sensor Networks: A Survey

Kai Xing [†], Shyaam Sundhar Rajamadam Srinivasan [†], Manny Rivera [†], Jiang Li [‡], Xiuzhen Cheng [†]

[†] *Computer Science Department*
George Washington University, Washington, DC 20052

[‡] *Department of Systems and Computer Science*
Howard University, Washington, DC 20059

Contents

1 Introduction

A wireless sensor network (WSN) is comprised of a large number of sensors that collaboratively monitor various environments. The sensors all together provide global views of the environments that offer more information than those local views provided by independently operating sensors. There are numerous potential applications of WSNs in various areas such as residence, industry, military and many others. For instance, people can use WSNs to build intelligent house, to gather machine information for real-time control in factories, and to track enemy movements in battle fields.

To collect data from WSNs, base stations and aggregation points [1] are commonly used. They usually have more resources (e.g. computation power and energy) than normal sensor nodes which have more or less such constraints. Aggre-

gation points gather data from nearby sensors, integrate the data and forward them to base stations, where the data are further processed or forwarded to a processing center. In this way, energy can be conserved in WSNs [2, 3] and network life time is thus prolonged.

WSNs have some special characteristics that distinguish them from other networks such as the Internet. The characteristics, listed as follows, demand careful considerations for protocol and algorithm designs that can lead to the use of WSNs in the real world:

- Sensors have limited resources, such as energy, memory and computation capacity. Light-weight protocols and algorithms are preferred to achieve longer sensor life.
- Sensors have limited reliability, partially because of the resource constraints.
- WSNs usually have dynamic topologies. Aside from sensors' leaving the network for reliability issues, new sensors may be added or activated and join the WSNs.
- WSNs can well have a large number of sensors.
- WSNs are usually centralized in terms of data processing and sometimes control as well. Data flow from sensors towards a few aggregation points which further forward the data to base stations of a fewer number. Base stations could also broadcast query/control information to sensors.

Among the designs of WSNs, security is one of the most important aspects that deserves great attention, considering the tremendous application opportunities. This chapter will lead readers into this area by presenting a survey of various potential attacks and solutions in WSNs. To ease the presentation, we classify the attacks based on the layering model of Open System Interconnection (OSI) (actually only four layers are used). We will present the mechanisms and effects of the attacks in four layers (physical, MAC, network and application), along with some potential countermeasures. A summary discussion is at the end.

2 Physical Layer

The physical layer is concerned with transmitting raw bits of information over wired/wireless medium. It is responsible for signal detection, modulation, encoding, frequency selection and so on, and is hence the basis of network operations.

2.1 Attacks in the Physical Layer

Many attacks target this layer as all upper layer functionalities rely on it. Adversaries can do "non-technical" things such as destroying sensors, or conduct"technical" actions such as wiretapping. In general, the following three types of attacks are categorized as physical layer attacks:

- Device Tampering
- Eavesdropping
- Jamming

2.1.1 Device Tampering

As imaginable, the simplest way to attack is to damage or modify sensors physically and thus stop or alter their services. The negative impact will be greater if base stations or aggregation points instead of normal sensors are attacked, since the former carry more responsibility of communications and/or data processing. However, the effectiveness of these attacks against physical sensors is very limited due to the high redundancy inherent in most WSNs. Unless large amount of sensors are compromised, the operations of WSNs will not be affected much.

Another way to attack is to capture sensors and extract sensitive data from them. As more complicated attacks (e.g. spoofing and denial of services) are made possible by this step (based on the sensitive data), such attacks are probably more threatening.

2.1.2 Eavesdropping

Without senders and receivers' awareness, eavesdropping [4, 5, 6] attackers monitor the traffic in transmission on communication channels and collect data that can later be analyzed to extract sensitive information. WSNs are especially vulnerable to such attacks since wireless transmission is the dominant method of communication used by sensors. During transmission, wireless signals are broadcast in the air and thus accessible to the public. With modest equipment, attackers within the sender's transmission range can easily plug themselves into the wireless channel and obtain raw data. By and large, the capability of eavesdropping depends on the power of antennas. The more powerful the antennas, the weaker signals attackers can receive, and thus the more data can be collected. Since eavesdropping is a passive behavior, such attacks are rarely detectable.

2.1.3 Jamming

Unlike device tampering attacks that are physical, jamming attacks disrupt the availability of transmission media. The approach is to introduce intense interference to occupy the channels and bereave normal sensors of the chances to communicate. With a device jamming its surrounding sensors, adversaries can disrupt an entire sensor network by deploying enough number of such devices. The problem of such attacks is that jamming devices have the risk of being identified, since sensors close to a jamming device may detect higher background noise than usual.

2.2 Countermeasures in the Physical Layer

Some attacks in the physical layer are quite hard to cope with. For example, after sensors are deployed in the field, it is difficult or infeasible to prevent every single sensor from device tampering. Therefore, although there are some mechanisms that attempt to reduce the occurrences of attacks, more of them focus on protecting information from divulgence.

2.2.1 Access Restriction

Obviously, restricting adversaries from physically accessing or getting close to sensors is effective on all the attacks aforementioned. It is good to have such restrictions if we can, but unfortunately, they are either difficult or infeasible in most cases. Therefore, we usually have to fall back on another type of restrictions: communication media access restriction.

A few techniques exist nowadays that prevent attackers from accessing the wireless medium in use, including sleeping/hibernating and spread spectrum communication [7]. The former is fairly simple as it switches off sensors and keeps them silent until the attackers go away. However, its effectiveness is at the expense of sacrificing the operations of WSNs. The latter is more intelligent, with frequencies varying deliberately. This technique uses either analog schemes where the frequency variation is continuous, or digital schemes (e.g. frequency hopping) where the frequency variation is abrupt. By this way, attackers cannot easily locate the communication channel, and are thus restrained from attacking. With current technology, powerful devices are required to perform such functionalities. Therefore, spread spectrum communications are not yet feasible for WSNs that are usually constrained in resources. Nonetheless, given the rapid advancement of technologies, this technique is very promising in the future.

Directional antenna [8, 9, 10, 11, 12] is another technique for access restriction. By confining the directions of the signal propagation, it reduces the chances of adversaries accessing the communication channel. Again, similar to spread spectrum

communication, its production cost is high at present and unsuitable for large-scale sensor networks, but may be more useful in the long run.

2.2.2 Encryption

In general, cryptography is the all-purpose solution to achieve security goals in WSNs. To protect data confidentiality, cryptography is indispensable.

Cryptography can be applied to the data stored on sensors. Once data are encrypted, even if the sensors are captured, it is difficult for the adversaries to obtain useful information. Of course, the strength of the encryption depends on various factors. A more costly encryption can yield higher strength, but it also drains the limited precious energy faster and needs more memory.

More often, cryptography is applied to the data in transmission. There are basically two categories of cryptographic mechanisms: asymmetric and symmetric. In asymmetric mechanisms (e.g. RSA [13, 14, 15]), the keys used for encryption and decryption are different, allowing for easier key distribution. It usually requires a third trusted party called Certificate Authority (CA) to distribute and check certificates so that the identity of the users using a certain key can be verified. However, due to the lack of *a priori* trust relationship and infrastructure support, it is infeasible to have CAs in WSNs. Furthermore, asymmetric cryptography usually consumes more resources such as computation and memory.

In comparison, symmetric mechanisms are more economical in terms of resource consumption. As long as two nodes share a key, they can use this key to encrypt and decrypt data and securely communicate with each other. However, the problem of lacking *a priori* trust relationship and infrastructure support persists. How to establish a shared key for two communicating parties is a challenging issue.

For key establishment, some researchers have proposed random key distribution schemes [16, 17, 18], in which each sensor randomly picks a set of keys from a large pool. As a result, each sensor has a shared key with any of its neighbors with some probability after deployment. Alternatively, we can have a full pairwise scheme in which each sensor shares a unique key with any other sensor in the network. Thus any pair of sensors is guaranteed to share a key. However, since each sensor needs to store $n - 1$ (assuming the total number of sensors is n) keys, this scheme suffers from a high memory cost of $O(n)$.

In the peer intermediaries scheme for key establishment protocol (PIKE) [19], authors use intermediary sensors as trusted parties to establish symmetric keys. Each node shares a unique key with each of $O(\sqrt{n})$ nodes [1]. When nodes i and

[1] We use the terms "node" and "sensor" interchangeably in this chapter.

j need to communicate but have no common key, they first find out a node k that shares a unique pairwise key with each of them. A path key will be computed for i and j through k. This protocol improves the memory cost to $O(\sqrt{n})$ compared to the full pairwise scheme, but sacrifices some security due to the possible unreliability of intermediary sensors.

Another key pre-distribution scheme is proposed by Du et al. [20], in which multiple key spaces based on Blom's method [21] are computed off-line and each sensor is preloaded randomly with information from one or more key spaces. As long as two sensors have information from the same key space, they can compute a shared key. In Blom's method, a key space is defined by a matrix pair (G, D), where G is public while D is private. Each node stores a column of G and a row of A, which is computed from G and D. To get a shared key, two nodes first exchange their columns of G, then compute the shared key using their private rows of the matrix A. It allows any pair of nodes to find a secret pairwise key by using $\lambda + 1$ units of memory space. Blom's method has the $\lambda - secure$ property, which means as long as no more than λ number of sensors are compromised, the corresponding key space remains perfectly secure.

Two in-situ based key management schemes, iKMS and sKMS, have been proposed in literature [22, 23]. In iKMS, service sensors, with each carrying a key space, and worker sensors, with no *a priori* knowledge, are deployed at the same time. Worker sensors obtain security information through an asymmetric secure channel from service nodes after deployment and then compute shared key with their neighbors. In sKMS, homogeneous sensors are preloaded with several system parameters and they differentiate their roles as either service nodes or worker nodes after deployment. Each service node constructs a key space based on Blom's method, and distributes the key information to a number of worker sensors through a secure channel established by Rabin's algorithm. sKMS is "perfect" in against node capture attack, achieves high connectivity (close to 1) in the induced key-sharing graph, and consumes a small amount memory in worker sensors.

3 MAC Layer

Sensors rely on Medium Access Control (MAC) layer to coordinate their transmissions to share the wireless media fairly and efficiently [24]. In wireless MAC protocols, typically nodes exchange control packets (e.g. CTS and RTS in IEEE 802.11) to gain the right for data transmission over the channel for a certain period of time. Node identifications are embedded in the packets to indicate senders and receivers.

3.1 Attacks in the MAC Layer

Due to the openness of wireless channels, the coordinations between sensors based on MAC protocols are subject to malicious manipulation. Adversaries can disobey the coordination rules and produce malicious traffic to interrupt network operations in the MAC layer. They can also forge MAC layer identifications and masquerade as other entities for various purposes.

3.1.1 Traffic Manipulation

The wireless communication in WSNs (and other wireless networks) can be easily manipulated in the MAC layer. Attackers can transmit packets right at the moment when legitimate users do so to cause excessive packet collisions. The timing can be readily decided by monitoring the channel and doing some calculations based on the MAC protocol in effect. The artificially increased contention will decrease signal quality and network availability, and will thus dramatically reduce the network throughput [25, 26]. Besides, in widely used MAC schemes where packet transmissions are carefully coordinated, attackers can compete for channel usage aggressively disobeying the coordination rules [27, 28, 29]. This misbehavior can break the operations of the protocols and result in unfair bandwidth usage. In either way, the network performance is degraded. Eventually, the collisions and unfairness lead traffic distortion.

3.1.2 Identity Spoofing

MAC identity spoofing is another common attack in the MAC layer [30]. Due to the broadcast nature of wireless communications, the MAC identity (such as a MAC address or a certificate) of a sensor is open to all the neighbors, including attackers. Without proper protection on it, an attacker can fake an identity and pretend to be a different one. A typical MAC identity spoofing attack is the Sybil attack [31, 32], in which an attacker illegally presents multiple MAC identities.

To gain access to the network or hide, an attacker can spoof as a normal legitimate sensor. It can even spoof as a base station or aggregation point to obtain unauthorized privileges or resources of the WSN. If successful, the entire network could be taken over.

Spoofing attacks are usually the basis of further cross-layer attacks that can cause serious consequences. For example, Sybil attacks [31, 32] may expose legitimate information to the adversary or provide wrong information for routing to launch false routing attacks (Section 4.1.1).

3.2 Countermeasures in the MAC Layer

To counter attacks in the MAC layer, current research focuses on detection. It allows for many kinds of further actions to stop the attacks, such as excluding the attacking nodes from interactions. There also exist some prevention approaches, which are mainly against spoofing attacks.

Many solutions presented below are actually proposed for ad hoc networks. We believe they can be easily extended to wireless sensor networks.

3.2.1 Misbehavior Detection

Because attacks deviate from normal behaviors, it is possible to identify attackers by observing what has happened. Various data can be collected for this purpose, and various actions can be taken after detection.

In a countering scheme [33] for the IEEE 802.11 protocol, a receiver assigns and adjusts the backoff values to be used by the corresponding sender. Whenever detecting the sender's misbehavior in manipulating backoff value, the receiver may add some penalty to the next backoff value assigned to the sender. The idea was applied to ad hoc networks [29], and similarly can also be applied to WSNs.

Another solution uses "watchdogs" [34] on every node to monitor whether or not the neighbors of a node forward the packets sent out by this particular node. A neighbor not forwarding packets will be identified by the watchdog as a misbehaving node. A similar scheme for MANET [35, 36] requires an intrusion detection system (IDS) on each node. The IDS monitors all the local activities (of users, system and communication) in the neighborhood. If abnormal behaviors are detected, the IDS will trigger some local actions, for example, alert the local user. In addition, the IDS may request neighboring nodes to cooperate for a global intrusion detection. Each node will propagate its information to its immediate neighbors. If the majority of such information received by a node indicates intrusions, the misbehaving nodes can be identified and precluded from the network.

Some other solutions use ratings to distinguish between good and bad nodes. In CORE [37], the rating is called "reputation", and is evaluated based on each entity's collaborativeness in communication. Misbehaving nodes will eventually gain a "bad" reputation and thus be excluded from communication by others. The mobile intrusion detection system (MobIDS) [38] is a variation of the reputation mechanism. The MobIDS on each node overhears the forwarded packets by its next hop and check whether its neighbor sensors faithfully forwards the packet or not. In addition, an iterative probing mechanism is used: when sending a packet, a sensor encrypts an intermediary node id in the packet head; when receiving the packet, the corresponding intermediary node, if normal, is supposed to decrypt the packet

head and sends back a reply to the sender. During the overhearing and probing, observations between $[-1, 1]$ are generated. A positive value represents a positive behavior while a negative value indicates otherwise. With these observations, a node has a local rating of its neighboring nodes. The rating is securely distributed to neighboring nodes with a signature. After a node collects enough local ratings for a certain node, it will average these ratings and generate a global rating for that node. Based on the global rating, that node may or may not be excluded from the communication.

Game theory has also been used for misbehavior detection. These approaches assume that misbehaving nodes take greedy actions to gain better performance such as higher share of bandwidth, and leverage the optimal point called "Nash equilibrium". Konorski [39] proposes a misbehavior-resilient backoff algorithm for ad hoc networks in which all nodes can hear each other. By adjusting the backoff value, the network may reach a fair equilibrium for bandwidth allocation. Cagalj et al. [40] considers those selfish nodes that reduce the contention window size in CSMA/CS ad hoc networks to gain higher throughput/bandwidth. At the operating point of "Nash equilibrium", all the nodes with similar traffic constraints and the same contention window size should get similar throughput. Based on this assumption, each node measures the throughput of all nodes at the point of equilibrium. If a node is observed to have a different throughput from others, it could well be a misbehaving one.

Note that the ideas presented above, such as using watchdog, rating nodes or comparing nodes' behavior at "Nash equilibrium", can also be used to develop misbehavior detecting techniques in other layers, as long as attackers' misbehavior deviates from normal. Nonetheless, considerations have to be given based on the layer-specific features, for example, how and what to watch, what metrics are used to rate nodes, and what behavior is abnormal at "'Nash equilibrium", and so on.

3.2.2 Identity Protection

Identity can be treated as yet another kind of information whose legitimacy needs to be guaranteed. Therefore, cryptography-based authentication can be used to prevent identity spoofing. Since most authentication schemes are designed for the network layer and the application layer, we will postpone the discussion of authentication schemes until in Section 4.2.1. Readers should keep in mind that the authentication techniques discussed there can also be applied to identity protection in the MAC layer.

In addition to authentication, other security measures also exist for this problem. Most of them are for false identify detection, as presented in the following:

- Radio resource testing was proposed to counter Sybil attacks [31]. It assumes that attackers consume more channel resources but can only use one single channel each time. By assigning different channels to neighboring nodes, the verifier can identify Sybil attackers through unused assigned channels.

- Position verification can be used to detect immobile attackers. If different identities appear at the same position, the node at that place can be identified as an attacker.

- Code attestation is based on the assumption that the code running on attackers or compromised nodes is different from that running on normal nodes. Therefore, attackers can be identified by validating the code running on them, for example, by verifying the memory content. One technique to verify the code running in a remote embedded device is proposed in SWATT [41]. Its design ensures that the result returned by the embedded device can be correct only if the memory contents are correct. The verifier first sends a challenge to the embedded device, for which the later computes a response through the verification procedure. After that, the verifier locally computes the answer to the challenge. By checking the two answers, the embedded device can be verified.

- Sequence checking is the method to check the sequence number in the header of 802.11 frames. First a pattern of legitimate sequence number activity for each MAC address is established. If the behavior of a node deviates from its sequence pattern, this node can be identified as an attacker.

- Identity-key association [32] can also help to reduce false identities. The key idea is to associate the node identity with keys used by the node in communication. An attacker can impersonate a node in front of another only if the communication key shared by them is cracked.

4 Network Layer

In the network layer, the key issues include locating destinations and calculating the optimal path to a destination. By tampering with routing service such as modifying routing information and replicating data packets, attackers can fail the communication in WSNs.

4.1 Attacks in the Network Layer

As in most other networks, sensors collaborate for routing in WSNs. However, the collaboration between sensors are susceptible to malicious manipulation in WSNs. Adversaries can gain access to routing paths and redirect the traffic, or distribute false information to mislead routing direction, or launch DoS attack against routing (such as flooding packets in order to block/interrupt the traffic in the network), acting as black holes to swallow (i.e. to receive but not forward) all the received messages, selectively forwarding packets through certain sensors, etc.

4.1.1 False Routing

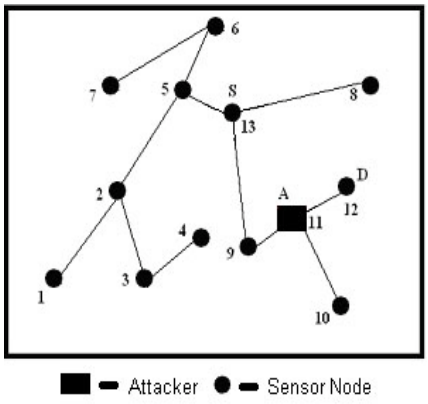
As the name suggests, false routing attacks [42] are launched by enforcing false routing information. There are three different approaches of enforcement [42]:

- Overflowing routing tables
- Poisoning routing tables
- Poisoning routing caches

Overflowing Routing Tables If the routing table of a normal network node overflows, the node will have to discard and thus ignore later incoming routing information. Therefore, attackers can inject a large volume of void routing information into the network. The injected information will eventually occupy the majority of the routing table space on normal nodes and cause overflow.

As an example, in the network of Figure 1(a), node 13 ('S') is the source, node 12 ('D') is the destination, and node 11 ('A') is the attacker. If A was a normal node, the routing table of it would be as shown in Figure 1(b). S would then be able to communicate with D. However, as an attacker, A keeps sending into the network wrong routing information about nonexistent nodes. The routing table of S will hence become the one in Figure 1(d), and the network will be visioned by S as in Figure 1(c). The visioned network does not contain any paths between S and D, and restrains S from communicating with D.

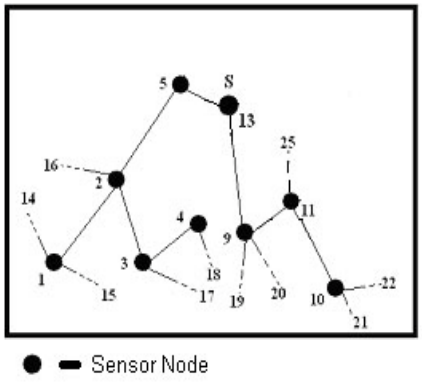
Routing table poisoning In this type of attacks, compromised nodes inside the network modify route update packets before sending or forwarding them out, i.e. make "poison". Such modifications result in wrong routing tables of all nodes inside the network. For example, in a network (Fig. 2(a)) with a compromised node (node 11) 'A', a source (node 13) 'S' and a destination



(a) Network topology

Dest.	Path	Dest.	Path
5	13→5	7	13→5→6→7
2	13→5→2	8	13→8
3	13→5→2→3	9	13→9
4	13→5→2→3→4	11	13→9→11
1	13→5→2→1	12	13→9→11→12
6	13→5→6	10	13→9→11→10

(b) Correct routing table of S



(c) Wrong network topology visioned by S

Dest.	Path	Dest.	Path
14	13→5→2→1→14	11	13→9→11
15	13→5→2→1→15	10	13→9→11→10
16	13→5→2→16	20	13→9→20
17	13→5→2→3→17	21	13→9→11→10→21
18	13→5→2→3→4→18	22	13→9→11→10→22
19	13→9→19	25	13→9→11→25

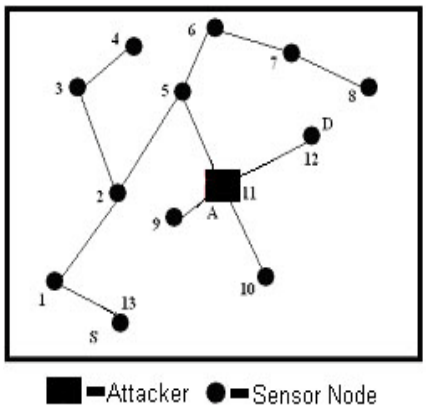
(d) Wrong routing table of S

Figure 1: A Network Before and After the Attack of Overflowing Routing Tables

(node 12) 'D', without poisoning, the routing table of S is as in Figure 2(b). With poisoning, it may become one in Fig. 2(d), giving a wrong vision of the network (Fig. 2(c)) to S.

Poisoning routing tables will direct traffic onto wrong paths, and may result in congestion or even collapse of networks. It may also lead to further attacks by putting attackers into the desired route.

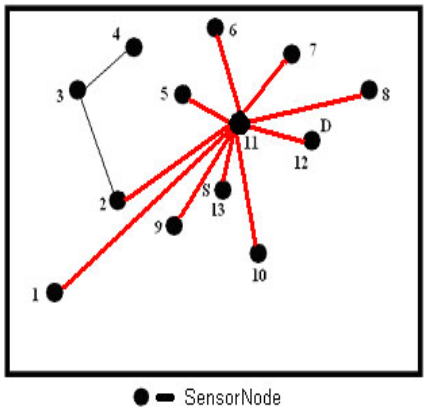
Route Cache Poisoning The third kind of false routing attacks can be achieved by poisoning the cache. Some on-demand routing protocols [43] require each node to maintain a cache with the most recent route information. This cache



(a) Network topology

Dest.	Path	Dest.	Path
1	13→1	7	13→1→2→5→6→7
2	13→1→2	8	13→1→2→5→6→7→8
3	13→1→2→3	9	13→1→2→5→11→9
4	13→1→2→3→4	10	13→1→2→5→11→10
5	13→1→2→5	11	13→1→2→5→11
6	13→1→2→5→6	12	13→1→2→5→11→12

(b) Correct routing table of S



(c) Wrong network topology visioned by S

Dest.	Path	Dest.	Path
1	13→11→1	7	13→11→7
2	13→11→2	8	13→11→8
3	13→11→2→3	9	13→11→9
4	13→11→2→3→4	10	13→11→10
5	13→11→5	11	13→11
6	13→11→6	12	13→11→12

(d) Wrong routing table of S

Figure 2: A Network Before and After the Attack of Poisoning Routing Tables

can be poisoned by the adversary, by using a technique similar to the attack used for poisoning the routing table.

In summary, there are three types of false routing attacks. A false routing attack can be used to place the adversary in its desired route, to divert route traffic from one part of the network to another, to restrain traffic on certain paths, and to bring down a part of or the entire network.

4.1.2 Packet Replication

In this type of attacks, attackers resend (replicate) packets previously received from other nodes. The packets can be broadcasted to the entire network (called *flooding attack*), or to a particular set of nodes. They can also resent irrespective of whether the sender is sending any new packets or not. With large amount of packets replayed, both the bandwidth of the network and the power of the nodes are consumed in vain, which leads to early termination of network operations.

4.1.3 Black Hole

The black hole attack is one of the simplest routing attacks in WSNs. In a black hole attack, the attacker swallows (i.e. receives but does not forward) all the messages he receives, just as a black hole absorbing everything passing by. By refusing to forward any message he receives, the attacker will affect all the traffic flowing through it. Hence, the throughput of a subset of nodes, especially the neighboring nodes around the attacker and with traffic through it, is dramatically decreased.

Different locations of the attacker induce different influences on the network. If the attacker is located close to the base station, all the traffic going to the base station might need to go through the attacker. Obviously, black hole attacks in this case can break the communication between the base station and the rest of the WSN, and effectively prevent the WSN from serving its purposes. In contrast, if a black hole attacking node is at the edge of the WSN, probably very few sensors need it to communicate with others. Therefore, the harm can be very limited.

4.1.4 Sinkhole

Sinkhole is a more complex attack [44] compared with black hole attack. Given certain knowledge of the routing protocol in use, the attacker tries to attract the traffic from a particular region through it. For example, the attacker can announce a false optimal path by advertising attractive power, bandwidth, or high quality routes to a particular region. Other nodes will then consider the path through this attacker node better than the currently used one, and move their traffic onto it.

Since affected nodes depend on the attacker for their communication, the sinkhole attack can make other attacks efficient by positioning the attacker in busy information traffic. Many other attacks, such as eavesdropping, selective forwarding and black holes, etc., can be empowered by sinkhole attacks.

4.1.5 Selective Forwarding

Selective forwarding attacks include two cases. In one case (Message Selective Forwarding), the attacker selectively send the information of a particular sensor; in the other case (Sensor Selective Forwarding), the attacker sends/discards the information from selected sensors. The former attack is considered as the application layer attack and will be discussed in Section 5.1.2, while the latter attack is considered as in the network layer and is the focus of this subsection.

Obviously, this attack can take place only when the attacker is on the route of packet transfer in a multi-hop network [45]. If the attacker happens to be on the route, it can just discard the packets from some selected nodes at its will. Otherwise, before the attack can be launched, it needs to position himself in the routing path using other attacks such as the Sybil attack, sinkhole attack and routing table poisoning attack.

4.1.6 Wormhole

A wormhole attack [46] requires two or more adversaries. These adversaries have better communication resources (e.g. power, bandwidth) than normal nodes, and can establish better communication channels (called "tunnels") between them. Unlike many other attacks in the network layer, the channels are real. Other sensors probably end up adopting the tunnels into their communication paths, rendering their output under the scrutiny of the adversaries.

4.2 Countermeasures in Network Layer

Since the functionalities of the network layer require the close collaboration of many nodes, all these nodes have to be enclosed for security consideration. It is therefore relatively difficult to mitigate attacks. Nonetheless, some countermeasures are available as follows:

- Routing Access Restriction
- False Routing Information Detection
- Wormhole Detection

4.2.1 Routing Access Restriction

Routing may be one of the most attractive attack targets in WSNs, as we saw in the previous subsection. If we can exclude attackers from participating in the routing

process, i.e. restrict them from accessing routing, a large number of attacks in the network layer will be prevented or alleviated.

Multi-path routing is one of the methods to reduce the effectiveness of attacks launched by attackers on routing paths [47][48][49]. In these schemes, packets are routed through multiple paths. Even if the attacker on one of the paths breaks down the path, the routing is not necessarily broken as other paths still exist. This alleviates the impact of routing attacks, although does not prevent these attacks.

A general way is to use authentication methods [50, 51, 52, 53, 54]. With authentication, it can be easily determined whether a sensor can participate in routing or not.

Authentication can be either end-to-end or hop-to-hop [52]. In end-to-end authentication, the source and destination share some secret and can thus verify each other. SEAD [55] and Ariadne [56] are two secure routing protocols based on end-to-end authentication. When a node receives a routing update, it always verify the sender of the update before accepting the update. In hop-to-hop authentication, each message in transmission is authenticated hop by hop. Therefore, the trust between the source and the destination is built upon the trust on all the intermediate nodes in the path. It is not as secure as end-to-end authentication, but is not so expensive as it does not require every pair of nodes share some common secret. Binkley and Trost [51] designe a link-level authentication mechanism for ad hoc routing in which IP and MAC addresses are used for hop by hop verification. Zhu et al. [53] propose an interleaved hop-to-hop authentication scheme that provides $t - security$: the injected false data packets can be detected when no more than t nodes are compromised. In this scheme, each sensor u_i associates itself to the sensor u_j that is $t+1$ hops closer to the base station. u_i is called the lower association node of u_j, and u_j is called the upper association node of u_i. Data are authenticated hop by hop between associated nodes until they reach the base station.

Hop-to-hop authentication can be combined with multi-path routing and result in multipath authentication [52]. The paths can be physical, meaning that messages are routed through multiple physically different communication paths. The paths can also be virtual, if they are actually on the same physical path, but are differentiated by other means such as encryption keys. Multipath authentication offers a tradeoff between resource constraints and security, and provides an in-between security level.

4.2.2 False Routing Information Detection

Sometimes attackers do have chances to send false routing information into the network, e.g. during route discovery stages. If the false information does not lead to network failure such as broken routes, we really cannot do much about it. Oth-

erwise, we can apply the idea of misbehavior detection discussed in Section 3.2.1.

For example, watchdog [34] or IDS [38, 35, 36] may find that some node fails to route messages along the routing path due to the wrong information it keeps. This anomaly of route failure may trigger out an alarm. Nodes can start to trace the source of false routing information. Reputation [38, 37] can also be maintained, depending on whether nodes are providing valid routing information. Nonetheless, how to trace the source of routing information can be a very difficult problem.

4.2.3 Wormhole Detection

Wormhole attacks are difficult to deal with because the information they inject into the networks is real. The most recent research work on the countermeasures focuses on the following techniques:

- Using synchronized clocks [57]. With the assumption that all nodes are tightly synchronized, each packet includes the time at which it is sent out. When receiving the packet, the receiver compares this value to the time at which it receives the packet. With the knowledge of transmission distance and consumed time, the receiver is able to detect if the packet has traveled too far. If the transmission distance is far beyond the maximum allowed travel distance, probably it is under wormhole attacks.

- Using directional antennas [8]. Directional antenna is used to discover neighboring nodes identified by zone. The zones around each sensor are numbered 1 to N oriented clockwise starting with zone 1 facing east. After receiving signals from unknown nodes, a node can get approximate direction information based on received signals and identify the unknown node by zone. After that it cooperates with its neighboring nodes to verify the legitimacy of the unknown node, e.g. by checking whether the unknown node is known by the neighboring nodes.

- Using Multidimensional Scaling - Visualization of Wormhole (MDS-VOW) [58]. MDS-VOW first constructs the layout of the network. If there exist wormhole attackers, the shape of the constructed network layout will show some bent/distorted features.

5 Application Layer

The application layer implements the services seen by users. Two examples of important applications in WSNs are data aggregation and time synchronization,

where data aggregation sends the data collected by sensors to base stations, and time synchronization synchronizes sensor clocks for cooperative operations.

5.1 Attacks in the Application Layer

Attacks in this layer have the knowledge of data semantics, and thus can manipulate the data to change the semantics. As the result, false data are presented to applications and lead to abnormal actions. In this section, the following attacks will be discussed:

- Clock Skewing
- Selective Message Forwarding
- Data Aggregation Distortion

5.1.1 Clock Skewing

The targets of this attack are those sensors in need of synchronized operations (e.g [59][60][61]). By disseminating false timing information, the attacks aim to desynchronize the sensors (i.e. skew their clocks).

For example, in IEEE 802.11 (which can be applied to WSNs), nodes are required to be synchronized with the access point. Beacon packets are broadcasted by the access point periodically. The packets contain timing information to be used by nodes for clock adjustment. Attackers can send false beacon packets with wrong timing information [59][62]. Once nodes adjust their clocks based on the wrong information, they will be out of synchronization with the access point. Although true beacon packets later can bring them back to synchronization, the nodes will oscillate between the two states and be unstable.

5.1.2 Selective Message Forwarding

For this attack, the adversary has to be on the path between the source and the destination, and is thus responsible for forwarding packet for the source. The attack can be launched by forwarding some or partial messages selectively but not others. Note that the attack is different from the other selective forwarding attack in the network layer (Section 4.1.5). To launch the selective forwarding attack in the application layer, attackers need to understand the semantics of the payload of the application layer packets (i.e. treat each packet as a meaningful *message* instead of a monolithic unit), and select the packets to be forwarded based on the semantics. In comparison, the selective forwarding attack in the network layer only requires

attackers to know the network layer information, such as the source and destination addresses. Attackers decide whether to forward packets according to those kinds of information only, and therefore operate at coarse granularity.

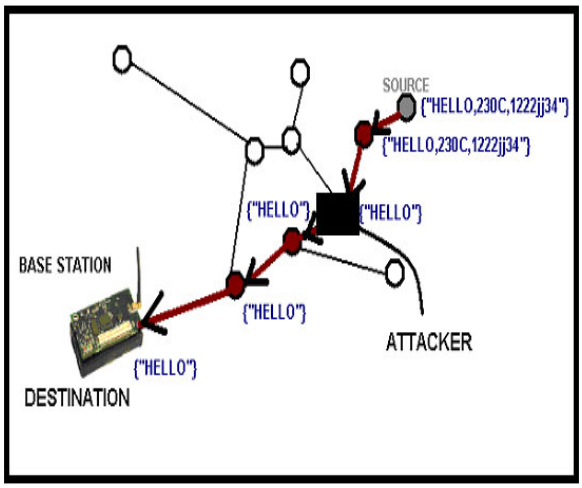


Figure 3: A Selective Message Forwarding Example

5.1.3 Data Aggregation Distortion

Once data is collected, sensors usually send it back to base stations for processing. Attackers may maliciously modify the data to be aggregated, and make the final aggregation results computed by the base stations distorted. Consequently, the base stations will have an incorrect view of the environment monitored by the sensors, and may take inappropriate actions.

Data aggregation can be totally disrupted if black hole or sinkhole attacks (Section 4.1.3) are launched. In this scenario, no data can reach the base stations. However, for those attacks, only the network layer knowledge is required. Therefore, they are categorized as network layer attacks.

5.2 Countermeasures in the Application Layer

As presented above, attacks in the application layer rely on application data semantics. Therefore, the countermeasures focus on protecting the integrity and confidentiality of data, no matter it is for control or not.

5.2.1 Data Integrity Protection

In general, authentication can be used to protect any data integrity. As discussed in Section 4.2.1, nodes can use end-to-end, hop-to-hop or multipath authentication depending on the cost they can afford and the security level they desire.

When authentication is not adopted, e.g. for feasibility reasons, or when data integrity is somehow compromised, the misbehavior detection techniques as discussed in Section 3.2.1 can be applied. The differences lie in the data to be observed in order to collect proofs of anomalies. Taking the clock skewing attack as an example: to detect such attacks, timing information in synchronization packets should be watched.

When readings (the data collected by sensors about the monitored environment) are considered, some specific detection mechanisms have been proposed, and are referred to as *false reading detection*. With an assumption that the faulty/compromised sensors produce readings remarkably deviated from the normal condition, an outlier detection algorithm [63] can locate such sensors by comparing their readings with those of their neighbors. In the online deviation detection scheme [64], an estimation of the data distribution is computed through the input data stream of the WSN. If the current reading of a sensor remarkably deviates from the data distribution (namely the normal readings in the WSN), this sensor will be detected as an outlier. There is also a centralized approach [65]. Base stations launch marked packets to probe certain sensors and try to route packets through them. If a sensor fails to respond, the base stations may conclude that this node is dead.

5.2.2 Data Confidentiality Protection

Encryption is an effective approach to prevent attackers from understanding captured data. Similar to authentication, the principles of encryption do not change for use in different layers. Readers are referred to Section 2.2.2 for the detailed discussion of encryption in WSNs.

6 Discussion

Although we discuss the attacks separately in this chapter, the attacks in fact are often launched in combination. The combination can be cross-layer in which multiple attacks in different layers are launched in a collaborative way. For example, the Sybil attack (in the MAC and network layer) provides identity spoofing for adversaries to do the wormhole attack (in the network layer). The combination can also be intra-layer in which multiple attacks in the same layer occur simultaneously. For example, in the network layer, a wormhole attack can be launched to lure traffic to

a compromised node that does sinkhole attack. Such combinations complicate the situation of WSN security and demand further research on countermeasures.

Besides, the same kind of attacks may be present in multiple layers, although they use different techniques. For instance, denial of services (DoS) attacks exist in physical layer, MAC layer, and network layer [27]; Sybil attacks exist in both MAC layer and network layer [32]. For each kind of such attacks, since their fundamentals are the same, our discussion on their characteristics is usually more detailed in one layer than in others.

We also notice that not only the same kind of attacks but also the same kind of countermeasures can appear in multiple layer. For example, misbehavior detection techniques can be applied to almost all the layers we discussed. Again, we usually discuss these techniques in more details in one layer than in others.

7 Conclusion

In this chapter, a survey is given on existing and potential attacks in wireless sensor networks. The attacks are classified according to the OSI stack model. For each layer of physics, MAC, network and application, we have discussed several typical attacks that exploit the characteristics of that layer. We have also covered the countermeasures and potential solutions against those attacks, and mentioned some open research issues. Hopefully by reading the chapter, the readers can have a better view of attacks and countermeasures in wireless sensor networks, and find their way to start secure designs for these networks.

References

[1] J. ibriq and I. Mahgoub, "Cluster-based routing in wireless sensor networks: issues and challenge," in *SPECS'04*, 2004, pp. 759–766.

[2] Y.Xu, J.Heideemann, and D.Estrin, "Energy conservation by adaptive clustering for ad-hoc networks," in *Poster session of MobiHoc'02*, 2002.

[3] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive energy-conserving routing for multihop ad hoc networks," may 2000, submitted for publication.

[4] M. Franklin, Z. Galil, and M. Yung, "Eavesdropping games: a graph-theoretic approach to privacy in distributed systems," *J. ACM*, vol. 47, no. 2, pp. 225–243, 2000.

[5] M. Abadi and J. Jürjens, "Formal eavesdropping and its computational interpretation," in *TACS '01: Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Software*. London, UK: Springer-Verlag, 2001, pp. 82–94.

[6] K. D. Murray, *Security Scrapbook Espionage and Privacy News of the Week*. [Online]. Available: $http://www.spybusters.com/SS0210.html$

[7] [Online]. Available: $http://www.faqs.org/rfcs/rfc1455.html$

[8] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Network and Distributed System Security Symposium(NDSS)*, 2004.

[9] R. R. Choudhury, X. Yang, N. H. Vaidya, and R. Ramanathan, "Using directional antennas for medium access control in ad hoc networks," in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2002, pp. 59–70.

[10] S. Yi, Y. Pei, and S. Kalyanaraman, "On the capacity improvement of ad hoc wireless networks using directional antennas," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2003, pp. 108–116.

[11] M. Takai, J. Martin, R. Bagrodia, and A. Ren, "Directional virtual carrier sensing for directional antennas in mobile ad hoc networks," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2002, pp. 183–193.

[12] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2001, pp. 95–105.

[13] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[14] J.-P. Hubaux, L. Buttyán, and S. Capkun, "The quest for security in mobile ad hoc networks," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. ACM Press, 2001, pp. 146–155.

[15] "Providing robust and ubiquitous security support for mobile ad hoc networks," in *ICNP '01: Proceedings of the Ninth International Conference on Network Protocols (ICNP'01)*. IEEE Computer Society, 2001, p. 251.

[16] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2003, p. 197.

[17] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 41–47.

[18] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 52–61.

[19] H. Chan and A. Perrig, "Pike: Peer intermediaries for key establishment in sensor networks," in *IEEE Infocom*, 2005.

[20] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. ACM Press, 2003, pp. 42–51.

[21] R. Blom, "Non-public key distribution," in *Advances in Cryptology: Proceedings of Crypto '82*, 1982, pp. 231–236.

[22] L. Ma, X. Cheng, F. Liu, M. Rivera, F. An, and J. Li, "ikms: An in-situ key management scheme for wireless sensor networks," 2005.

[23] F. Liu, X. Cheng, and L. Ma, "S-kms: A self-configured key management scheme for sensor networks," 2005.

[24] [Online]. Available: $http : //www.eweek.com/encyclopedia_term/0,2542,t = MAC + layeri = 46426,00.asp$

[25] V. Gupta, S. Krishnamurthy, and M. Faloutsos, "Denial of service attacks at the mac layer in wireless ad hoc networks." [Online]. Available: $http : //www.cs.ucr.edu/\ krish/milcom_vik.pdf$

[26] I. A. Jean-Pierre, "Denial of service resilience in ad hoc networks." [Online]. Available: http://lcawww.epfl.ch/Publications/aad/aadHK04.pdf

[27] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[28] P. Michiardi and R. Molva, "Prevention of denial of service attacks and selfishness in mobile ad hoc networks," in *Institut Eurecom Research Report RR-02-063*, 2002.

[29] A. A. Cardenas, S. Radosavac, and J. S. Baras, "Detection and prevention of mac layer misbehavior in ad hoc networks," in *Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks*, 2004.

[30] E. D. Cardenas, "Mac spoofing–an introduction," 2003. [Online]. Available: $http://www.giac.org/practical/GSEC/Edgar_Cardenas_GSEC.pdf$

[31] J. R. Douceur, "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, 2002, pp. 251–260.

[32] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*. ACM Press, 2004, pp. 259–268.

[33] P. Kyasanur and N. H. Vaidya, "Detection and handling of mac layer misbehavior in wireless networks." in *DSN*, 2003, pp. 173–182.

[34] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM Press, 2000, pp. 255–265.

[35] Y. Zhang, W. Lee, and Y.-A. Huang, "Intrusion detection techniques for mobile wireless networks," *Wirel. Netw.*, vol. 9, no. 5, pp. 545–556, 2003.

[36] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 275–283.

[37] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceedings of*

the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security. Deventer, The Netherlands, The Netherlands: Kluwer, B.V., 2002, pp. 107–121.

[38] F. K. Andreas, "Sensors for detection of misbehaving nodes in manets." [Online]. Available: http://medien.informatik.uni-ulm.de/forschung/publikationen/dimva2004.pdf

[39] J. Konorski, "Multiple access in ad-hoc wireless lans with noncooperative stations." in *NETWORKING*, 2002, pp. 1141–1146.

[40] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux, "On cheating in csma/ca ad hoc networks," in *EPFL Technical Report*, 2004.

[41] A. Seshadri, A. Perrig, L. van Doorn, and P. K. Khosla, "Swatt: Software-based attestation for embedded devices." in *IEEE Symposium on Security and Privacy*, 2004, pp. 272–.

[42] C. R. Murthy and B.S.Manoj, "Transport layer and security protocols for ad hoc wireless networks," in *Ad Hoc Wireless Networks - Architectures and Protocols*, 2004.

[43] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*. IEEE Computer Society, 1999, p. 90.

[44] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, no. 2–3, pp. 293–315, September 2003.

[45] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An empirical study of epidemic algorithms in large scale multihop wireless networks," 2002.

[46] Y. Hu, A. Perrig, and D. Johnson, "Wormhole detection in wireless ad hoc networks," 2002. [Online]. Available: citeseer.ist.psu.edu/hu02wormhole.html

[47] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, 2001.

[48] W. Lou, W. Liu, and Y. Fang, "Spread: Enhancing data confidentiality in mobile ad hoc networks," in *IEEE INFOCOM*, 2004.

[49] P. Papadimitratos and Z. J. Haas, "Secure data transmission in mobile ad hoc networks," in *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*. New York, NY, USA: ACM Press, 2003, pp. 41–50.

[50] K. Hoeper and G. Gong, "Models of authentication in ad hoc networks and their related network properties," in *Tech Reports*, 2004. [Online]. Available: $http://www.cacr.math.uwaterloo.ca/techreports/2004/cacr2004-03.pdf$

[51] J. Binkley and W. Trost, "Authenticated ad hoc routing at the link layer for mobile systems," *Wirel. Netw.*, vol. 7, no. 2, pp. 139–145, 2001.

[52] H. Vogt, "Exploring message authentication in sensor networks," in *1st European Workshop on Security in Ad Hoc and Sensor Networks (ESAS 2004)*, 2004.

[53] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-byhop authentication scheme for filtering false data injection in sensor networks," 2004.

[54] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS: security protocols for sensor netowrks," in *Mobile Computing and Networking*, 2001, pp. 189–199. [Online]. Available: citeseer.ist.psu.edu/perrig01spins.html

[55] Y.-C. Hu, D. B. Johnson, and A. Perrig, "Sead: Secure efficient distance vector routing in mobile wireless ad hoc networks," in *Fourth IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA '02), jun 2002, pp. 3–13. [Online]. Available: citeseer.ist.psu.edu/hu02sead.html

[56] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking* (MobiCom 2002), Sept. 2002, to appear. [Online]. Available: citeseer.ist.psu.edu/article/hu02ariadne.html

[57] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," 2001. [Online]. Available: citeseer.ist.psu.edu/hu01packet.html

[58] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*. ACM Press, 2004, pp. 51–60.

[59] E. SHI and A. PERRIG, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[60] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, 2002.

[61] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in *IPDPS '01: Proceedings of the 15th International Parallel & Distributed Processing Symposium*. Washington, DC, USA: IEEE Computer Society, 2001, p. 186.

[62] G. Khanna, A. Masood, and C. N. Rotaru, "Synchronization attacks against 802.11," in *Networks and Distributed Systems Symposium (NDSS) Workshop*, 2005.

[63] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[64] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Distributed deviation detection in sensor networks," *SIGMOD Rec.*, vol. 32, no. 4, pp. 77–82, 2003.

[65] J. Staddon, D. Balfanz, and G. Durfee, "Efficient tracing of failed nodes in sensor networks," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM Press, 2002, pp. 122–130.