

Anomaly Intrusion Detection in Wireless Sensor Networks

Vijay Bhuse, Ajay Gupta
{vsbhuse, gupta}@cs.wmich.edu

Western Michigan University, Kalamazoo, MI-49008, USA

Abstract: We propose lightweight methods to detect anomaly intrusions in wireless sensor networks (WSNs). The main idea is to reuse the already available system information that is generated at various layers of a network stack. To the best of our knowledge, this is the first such approach for anomaly intrusion detection in WSNs.

Keywords: Anomaly intrusion, Wireless sensor network, TDMA, CDMA, DSR, DSDV, FHSS.

1 Introduction

Wireless sensor networks (WSNs) consist of small devices (or sensor nodes) with radio, processor, memory, battery and sensor hardware. With widespread deployment of these devices one can precisely monitor the environment. Sensor nodes are resource constrained in terms of radio range, processor speed, memory and power. The resource constrained nature forces designers to design application specific systems. WSNs are mostly unguarded and the wireless medium is broadcast. This makes them more vulnerable to attacks. Without proper security measures, an enemy can launch various kinds of attacks in hostile environments. These attacks can disrupt the normal working of WSNs and can defeat the purpose of their deployment. Therefore security is an important aspect of these networks. The scarcity of resources forces designers to use traditional security primitives (like encryption, one way functions) sparingly. Intrusion detection is the second line of defense and it complements security primitives. To be practical to implement on WSNs, ideas to detect intrusions should be lightweight, distributed and scalable. In this paper we propose such methods to detect anomaly intrusion in WSNs.

Distributed networked embedded systems like WSNs are layered. They run different protocols at different layers. These protocols generate data that is used for specific purpose. For example routing layer generates forwarding tables, which are used for routing. At MAC layer protocols like CDMA, TDMA and S-MAC [15] are used. TDMA generates a schedule. S-MAC generates sleep/wake-up schedules. The physical layer runs FHSS, which uses data like frequency hopping pattern, hopping set etc. In this paper we propose methods, which use data, generated by such protocols for anomaly intrusion detection. We use the existing system information such as neighbor lists, routing tables, sleep wake up schedules, MAC layer transmission schedules etc. At physical layer we use received signal strength indicator (RSSI) values. At MAC layer we use TDMA and S-MAC schedules. At routing layer we use forwarding tables [3]. At application layer, we propose mechanism in which nodes guard each other. We try to detect anomaly at multiple layers using only the existing system information. If adversary escapes at one layer, there are still possibilities that it will be detected at other layers. The multilayer approach makes intrusion detection system robust. Mechanisms proposed at physical, MAC and application layer can be used to detect masquerading whereas the one at routing layer can be used to detect packet forging attacks. These mechanisms could be used concurrently to increase probability of anomaly detection. In this paper our focus is on detecting intrusions and not on handling them. We assume that the adversary does not interfere with route discovery.

Since we reuse the already available data generated by different protocols, our approach incurs very little additional cost and thus is ideally suited for resource constrained WSNs. We analyze the techniques and give tight bounds on the probability of detection, probability of false positives (or false alarms) and probability of false negatives (or misses). The rest of the paper is organized as follows. We discuss related work in section 2. In sections 3, 4, 5 and 6 we discuss techniques at physical, MAC, routing and application layer respectively. In section 7 we present summary of our results and conclude the paper.

2 Related work

In literature the term *intrusion* means both intrusion by outsider and insider abuse. In this paper we address intrusions caused by outsiders. It does not include insider abuse. Kumar has categorized intrusions into two types as listed below [8],

- *Misuse or Signature-based detection*: Intruder takes advantage of weaknesses in the system and finds out a way to get in. We can formally define these attack patterns. These attack patterns are called as signatures. So if new adversary tries to use known attacks to intrude then he will be caught if his pattern of attack matches some signature.
- *Anomaly detection*: In this type of intrusion detection, normal user behavior is defined and the intrusion detection system looks for anything that is anomalous hence suspicious. Anomaly detection assumes that intrusion is a kind of anomalous activity. So if it detects anomalous behavior, it can detect an intrusion.

One of the earliest works on intrusion detection is commonly considered to be the one reported by Anderson [1], which introduced the idea of doing anomaly detection by creating profiles of normal use and detecting deviations from those profiles. This idea was later formally presented by Denning [4] in what is considered to be the seminal paper for modern intrusion detection. For a review of intrusion detection in wireless ad hoc networks, we refer the reader to [10]. Zhang and Lee [16] proposed architecture for a distributed and cooperative intrusion detection system for ad hoc networks based on statistical anomaly detection techniques. This article does not discuss the actual detection techniques [10]. Bhargav *et al.* [2] proposed an intrusion detection and response model to enhance security in AODV [11]. Marti *et al.* [9] proposed two techniques: *watchdog* and *pathrater* that improve throughput in ad hoc network in the presence of nodes that agree to forward packets but fail to do so. We next discuss our lightweight solution to anomaly detection. We propose mechanisms, which can be implemented at multiple layers of a network stack.

3 Physical layer

The problem of protecting radio interface (like prevention of eavesdropping and jamming) has been intensively researched for virtually all wireless networks and many solutions have been proposed and deployed, such as spread spectrum communication and frequency hopping [5]. When a node receives a packet, it is difficult to find out if the packet came from the claimed sender unless explicit authentication is used. We try to address this problem by using Received Signal Strength Indicator (RSSI). Recently proposed embedded operating systems like TinyOS [14] provide functionality to get the RSSI value. For wireless medium, received signal strength is related to the distance between nodes.

3.1 RSSI value

We associate a neighbor with an estimated RSSI value. After deployment when nodes perform neighbor discovery, they record RSSI value for each neighbor. These recorded values can be used to detect intrusion afterwards. The packet received with RSSI value that is not in the range can be flagged. Similarly a sender can also be flagged for all further communication. Once intrusion is detected, various kinds of actions (like dropping a packet, flagging a neighbor etc.) can be taken. However in this paper we focus only on *intrusion detection* and hence do not discuss solutions to handle intrusions. There are many factors like background noise, weather conditions etc. that can lead this approach to produce higher percentage of false positives. Therefore this approach should be used in combination with others (such as the ones proposed later in this paper).

4. MAC Layer

If scheduling based protocols are used for media access then there is a specific time slot allocated to each node. If an adversary wants to masquerade as some node it has to do that in the time slot allocated to that node. If adversary does not follow this schedule and tries to masquerade as some node at a time when that node is not supposed to transmit, then nodes can detect an intrusion if they keep track of transmission schedule of other nodes. Below we show how this idea works for TDMA and S-MAC.

4.1 Time Division Multiple Access (TDMA)

TDMA is a digital transmission technology that allows a number of users to access a single radio-frequency channel without interference by allocating different time slots to different users within each channel.

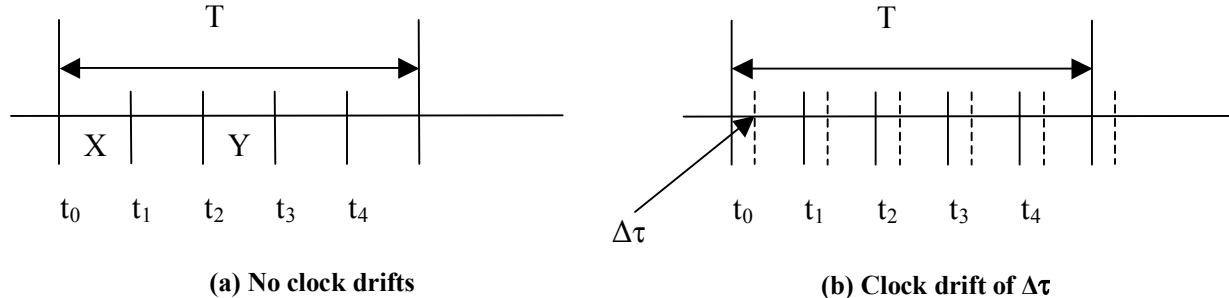


Figure 1: TDMA schedule for nodes

Suppose nodes keep track of TDMA schedules of other nodes that they communicate with. Node X is allocated time slot t_0 and node Y is allocated time slot t_2 (see figure 1(a)). For simplicity we assume that all time slots are of length τ . Suppose an adversary tries to send a packet with sender field set to X in time slot t_2 . For a node that receives a packet with source field in the packet set to X in a time slot that is not allocated to X, is an anomaly. In this way the data used by TDMA protocol can be used to detect intrusion. If an adversary sends a packet in the time slot t_0 by changing the source field to X then that packet will not be detected. If we assume that the adversary sends packet randomly in any time slot of length τ ($\tau = T/n$) and there are n nodes sharing the medium (in n slots), then the probability of detection is $(n-1)/n$. The probability of false negatives is $1/n$. If clocks are synchronized, there will be no false positives. If there is a clock drift of $\Delta\tau$, then the probability of false positives is $(\Delta\tau * n)/n * \tau = \Delta\tau/\tau$. The probability of false negatives is same as the probability of false positives.

4.2 S-MAC

Many MAC protocols like S-MAC have been proposed which use sleep/wake-up schedule for energy conservation (see figure 2). If those protocols are in use and node A receives a packet with source field set to X at a time when node X should be sleeping, then node A can easily detect that the packet is sent by an adversary. Node A can detect this intrusion because it is an anomaly. Above we propose anomaly detection technique for schedule-based MAC protocols. These techniques use the available data and hence incur very little overhead, which suits resource constrained nature of WSNs.

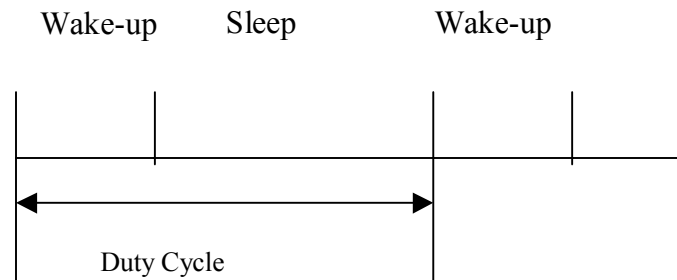


Figure 2: S-MAC sleep/wake-up schedule for a node

5 Routing Layer

For ad hoc networks, table driven and source initiated on demand routing protocols whereas for WSNs data dissemination mechanisms (or routing protocols) like Directed Diffusion [6] have been proposed. WSN is typically an ad hoc network of nodes with sensing abilities. So routing protocols proposed for ad hoc networks could also be used for WSNs. We propose an idea of using forwarding tables generated by routing protocols for anomaly intrusion detection. This idea has been initially proposed by us in [3]. In that paper we propose a protocol *information authentication for sensor networks* (IASN). We analyze and extend IASN in this paper. We show how it works with routing protocols like DSR [7], DSDV [12] and Directed Diffusion. We then extend it where a node running IASN keeps track of neighbors that are k hop away for $k > 1$.

We term high-level data as *information*. The main purpose of sensor networks is to sense some environmental variables and send readings periodically to a base station or send readings whenever someone demands them. Since multiple sensors are deployed to sense some environmental variables it is expected that they collaborate among themselves to generate meaningful information (if sensors are deployed to sense fire in woods, then detection of fire is a meaningful information). Sensor nodes have limited battery power and replacing batteries is a tedious job, which may require human intervention. Using explicit security primitives to provide authentication is an overhead. So in IASN we proposed an idea of authenticating *information* instead of authenticating the source of information. Taking into consideration the working of WSNs, path discovery should be on demand (like DSR or Directed Diffusion) instead of explicitly discovering, maintaining and updating routes (like DSDV). It can be source initiated (like DSR), receiver initiated (like Directed Diffusion) or both depending upon the application.

In IASN we try to secure information that is sent through the source to destination route. We try to detect intrusions that can occur on this route. We keep track of neighbors and the type of information expected from them. Upon information arrival we match the information against the neighbor and verify it. If that information is not supposed to come from a certain node, we guess that it is an attempt to infuse malicious packets. We show that IASN can work with source initiated, receiver initiated or table driven routing protocols. We tested our protocol IASN with DSR and DSDV. Results show considerable detection of forged packets at low cost. We briefly summarize it next and then extend the idea from the one presented in [3].

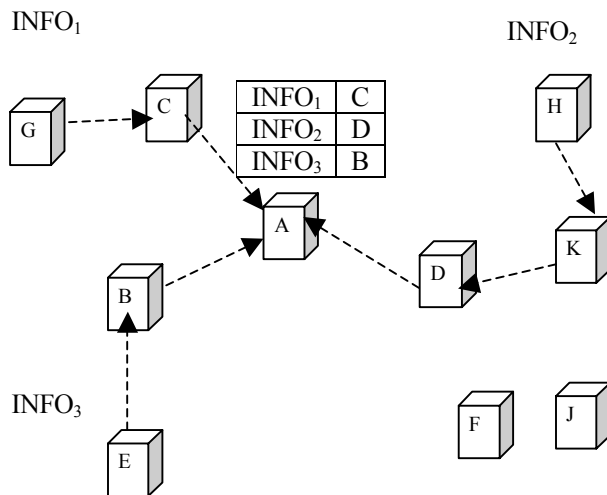


Figure 3: ADT for node A. Node A expects INFO₁, INFO₂ and INFO₃ from neighbors C, D and B, respectively.

receives from neighbors $N_1, N_2, N_3, \dots, N_p$ respectively.

Assume that the route discovery process is performed securely by using a secure routing protocol such as the one proposed in [18]. Once a path is established by a routing protocol and nodes know what information to expect from each of the neighbors, they can detect an anomaly if they receive that information from different neighbor. Suppose node E of figure 3 sends a packet to node A via node B containing sensed temperature (INFO₃) data, then node A expects temperature data from node B and node B expects temperature data from node E. If nodes A and B receive the temperature data from any other nodes except nodes B and E, respectively, then they can detect and filter (drop) forged packets. In general INFO _{i} is meaningful information like an answer (for example sensed temperature of some area at some particular time) to some query etc. Let INFO₁, INFO₂, INFO₃, ..., INFO _{p} indicate the p *informations* that a node

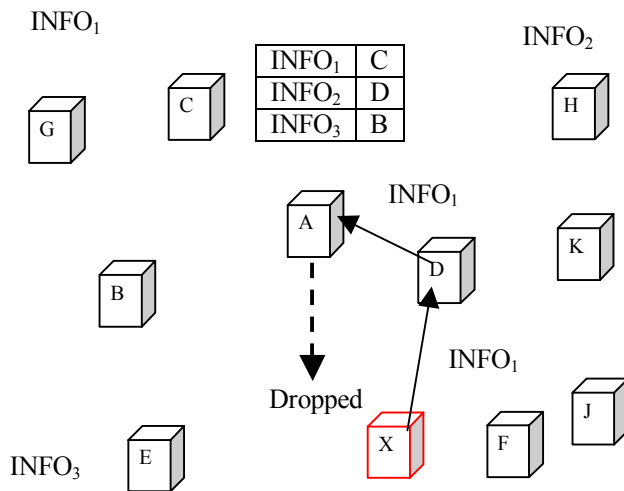


Figure 4: A detects forged packet from adversary X because it expects INFO₁ only from C.

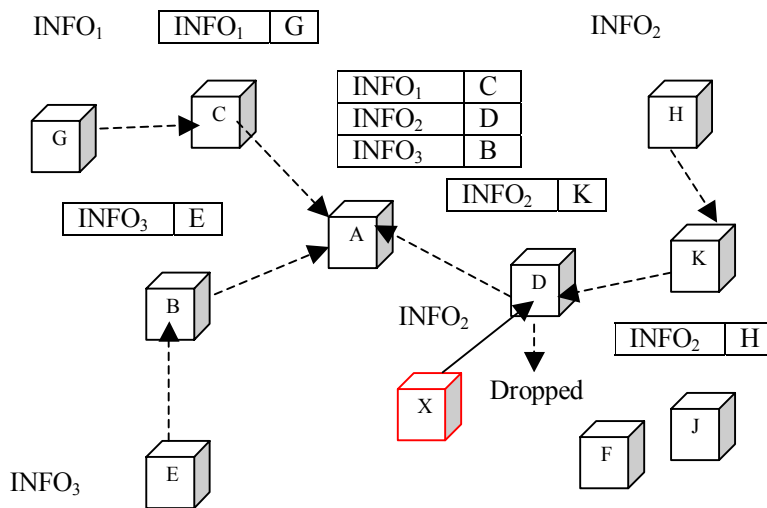


Figure 5: X sends INFO₂ to node D, which drops it.

system information to detect intrusions.

In Figure 5, nodes G, H, E, F and J have empty ADTs, as they are not receiving any information. Whereas A, B, C, K and D have some entries in their tables. In this example, if all nodes run IASN and if we assume that nodes do not masquerade (detecting masquerade without an explicit node authentication mechanism is a challenge) then all the forged packets will be detected and dropped. In the previous example, a node receives INFO of a certain type from a single neighbor. The ADT can be easily extended to accommodate the situations where (i) more than one neighbors are allowed to forward the same INFO, and (ii) multiple INFOs are forwarded by a neighbor.

It is easy to see that the storage overhead of ADT is proportional to the number of neighbors. ADT can be easily derived and maintained from the underlying data dissemination mechanisms or the routing protocols. The forwarding-tables or next-hop information of the routing protocols can be used to build the ADT. Furthermore route-update messages can be used to keep ADT in concurrence with the routing path changes. Figure 6 shows the updated ADT after a H→D path changes from Figure 5. This method of detecting anomaly intrusions is thus lightweight because it uses existing routing data.

Every node running IASN can maintain an anomaly detection table (ADT) containing the list of neighbors that may forward some particular *information* to that node. We embed the process of constructing ADTs in route discovery part of routing protocols so that ADTs are constructed and updated securely. IASN protocol detects and drops forged packets by comparing the *information* in the packet and the source of the packet against the entries in the ADT. Consider the WSN scenario of Figure 3 in more detail. Suppose nodes G, H, and E have established paths reaching node A to send INFO₁, INFO₂, and INFO₃ respectively. Node A gets INFO₁ from node C, INFO₂ from node D and INFO₃ from B. Node A can maintain an ADT. Now suppose there is an adversary X, which sends a forged packet containing INFO₁ to node A via node D (see Figure 4). If node A gets this packet from node B or D then it can detect that the packet is forged because it expects INFO₁ only from node C. If node A keeps information about route updates and keeps the ADT consistent with the current routing paths then it can very easily detect packet forging. Note that an adversary X can forge a packet with INFO₂ via node D. However, if node D also maintains ADT and runs IASN protocol, then this forged packet can be detected at node D. Figure 5 explains that case. Designers can thus use the

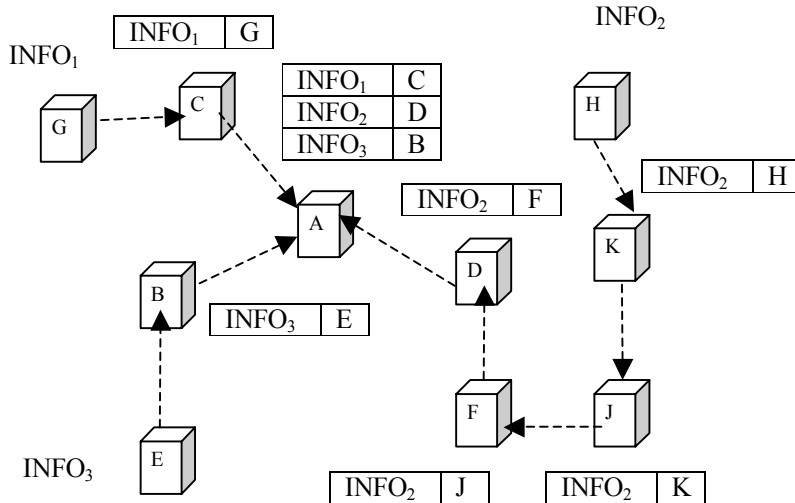


Figure 6: ADTs at nodes after change in route from H to A.

not needed anymore then a Cancel message has to be sent by a source to all the nodes on the route to delete ADTs.

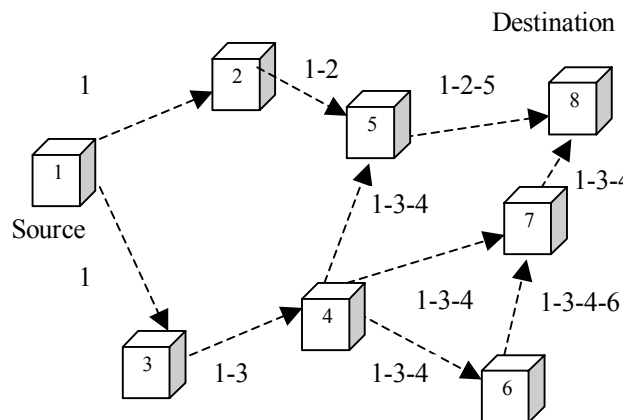


Figure 7: Node 1 floods route request packet.

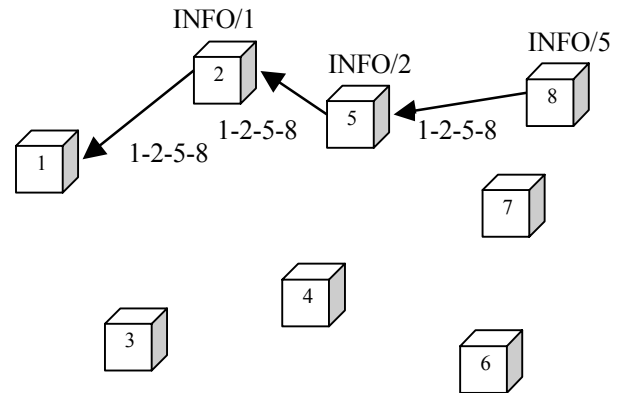


Figure 8: ADTs built during route reply phase.

5.1 Constructing ADTs for source initiated on demand ad hoc routing protocols (DSR)

DSR is an on demand routing protocol. It has *route request* and *route reply* phases. Initially a source floods a route request packet as shown in figure 7. A route reply is generated either by the destination or an intermediate node, which contains an unexpired route to a destination in its route cache [13]. In route reply phase when packet is coming back to the source, ADTs can be built at all the nodes that are on the path as shown in figure 8. If that route is

Let the cost of writing one byte into a packet be w , the cost of reading one byte from a packet be r , the cost of sending a packet be s and the cost of receiving a packet be v . Let n be the number of nodes, m be the maximum number of neighbors any node has. Let p be the maximum length of any path. The approximate cost of route request and route reply is $n(w + s + mv)$ and $p(s + v)$ respectively. Therefore the cost of DSR path discovery is $n(w + s + mv) + p(s + v)$. The cost of constructing ADTs at the nodes is pr . Let us assume that q messages are sent once a path is discovered and ADTs are constructed. The cost of writing source-id in a packet is qpw , whereas the cost of reading source-id from a packet is qpr . Therefore the cost of checking whether the information is coming from an intended neighbor is $qpr + qpw$. The cost of Cancel message is $p(s+v)$. Therefore the overhead is $(qpr + qpw + pr + p(s + v))$. The cost of message transmission over p hops is $qp(s+v)$. The ratio of overhead to the energy network consumes on DSR path discovery and transmission of packets using that path over some time period is $(qpr + qpw + pr + p(s + v)) / (n(w + s + mv) + p(s + v) + qp(s+v))$. To get an idea of the above ratio empirically, consider a practical situation in which $n=q=100$, $m=6$, $s=v$, $r=w$ and $p=10$, the ratio is $(2010 + 20(s/r)) / (2720(s/r) + 100)$. The cost of sending a packet is much greater compared to reading a byte from a packet. If we assume s/r to be 100 and all p nodes run IASN, then they can detect all the forged packets at the cost of 1.4 per cent

energy the network consumes on DSR path discovery and transmission of packets using that path over some time period.

5.2 Constructing ADTs for Directed Diffusion

Directed diffusion is a receiver-initiated protocol. In directed diffusion, destination (sink) floods the network in search of some data (called as “Interest”) as shown by dotted arrows in figure 9. Whenever that message reaches source, it floods the network back as shown by solid arrows. Then it reinforces only one path to sink, which is used for all future communication (see figure 10). It is during the reinforcement phase that we can construct ADTs on nodes that are on the path. A Cancel message has to be sent in order to wipe out ADTs if that path is no longer valid. We analyze overhead of IASN similar to DSR. The ratio of overhead to the energy network consumes on path discovery and transmission of packets using that path over some time period is $(qpr + qpw + pr + p(s + v)) / (2n(s + mv) + p(s + v) + qp(s+v))$. For $n=q=100$, $m=6$, $r = w$, $s=v$, $p=10$ and $s/r= 100$, the above ratio is 0.011 which is very small. If all p nodes run IASN, then they can detect all the forged packets at the cost of 1.1 per cent energy the network consumes on path discovery using directed diffusion and transmission of packets using that path over some time period.

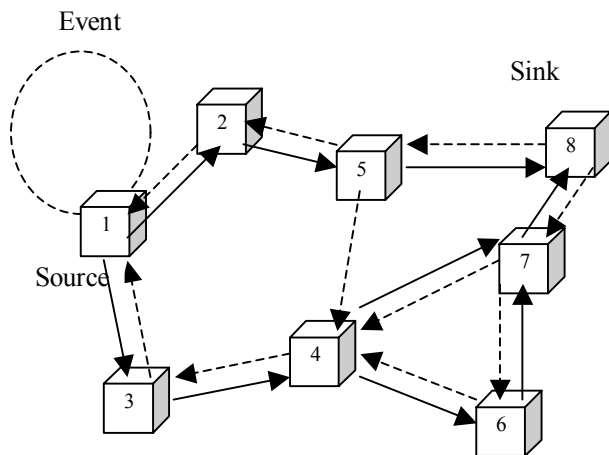


Figure 9: Sink floods interest (dotted arrows). Source replies with gradient message (solid arrows).

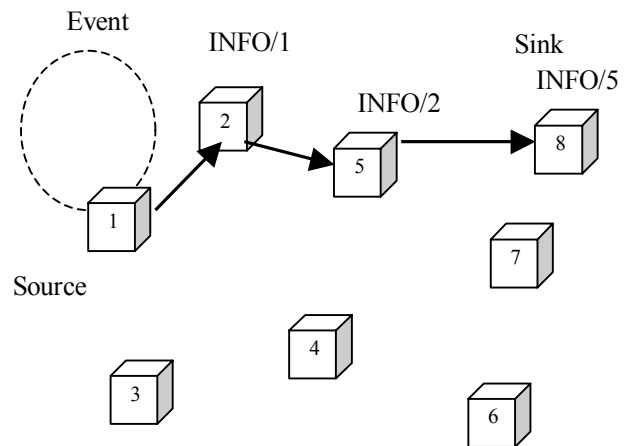


Figure 10: Source reinforces one route. ADTs can be built during reinforcement of a route.

5.3 Constructing ADTs for table driven protocol (DSDV)

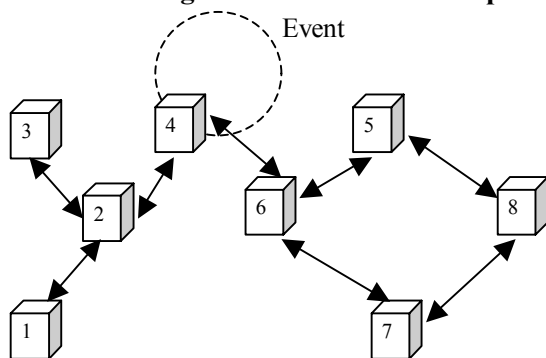


Figure 11: All nodes maintain routing tables as shown in table 1 for node 4 in DSDV ([12]).

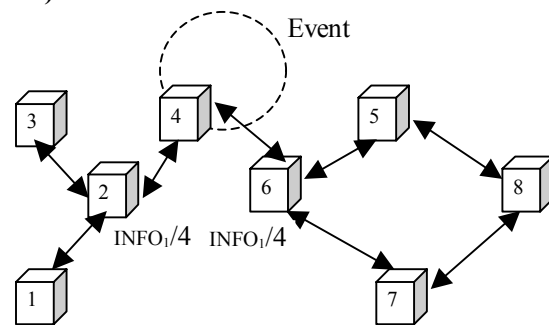


Figure 12: ADTs are built after node 4 advertises.

In DSDV every node maintains a routing table in which all the possible destinations and the number of hops to it are recorded (see table 1). Routing table updates are done in two ways: *full dump* and *incremental*. Full dump carries all the routing information whereas incremental carries only those updates, which have happened after the previous full dump. Table 2 is the advertised route table by node 4. In that

table we can append a field that will tell neighbors what INFO they should expect. When a route table is advertised, neighbors can construct their ADT (see figure 12). Thus constructing ADT at nodes running DSDV incurs a very little cost.

Destination	Next Hop	Metric	Sequence No.
1	2	2	S406_1
2	2	1	S128_2
3	2	2	S564_3
4	4	0	S710_4
5	6	2	S392_5
6	6	1	S076_6
7	6	2	S128_7
8	6	3	S050_8

Table 1: Forwarding table of node 4 for the network of figure 11 (modified from [12])

Destination	Metric	Sequence No.	INFO
1	2	S406_1	
2	1	S128_2	INFO ₁
3	2	S564_3	
4	0	S710_4	
5	2	S392_5	
6	1	S076_6	INFO ₁
7	2	S128_7	
8	3	S050_8	

Table 2: Modified advertised route table of node 4 to construct ADTs (courtesy [12])

5.4 Analysis of IASN

Let us consider a path from source 1 to destination N through nodes $2, 3, \dots, N-1$ as shown in figure 13.

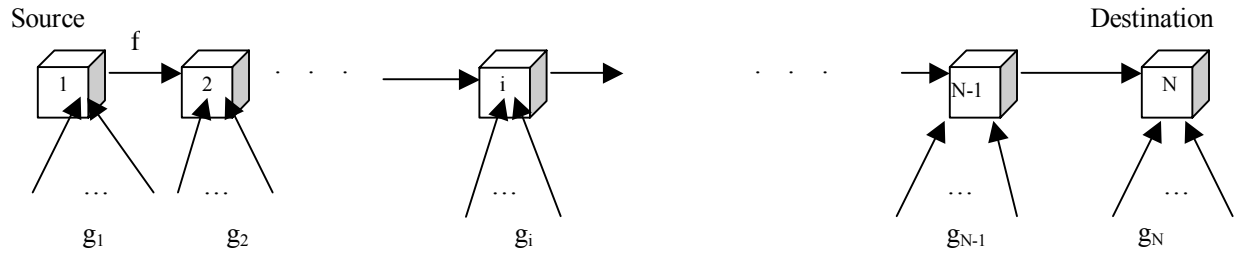


Figure13: A route from 1 to N

Node 1 is sending information, INFO to node N . Let f be the number of packets containing INFO sent by the source to node 2 . Let g_i be the number of packets with INFO forwarded by an adversary directly to node i or through neighbors of node i except its predecessor $i-1$ to node i for $1 \leq i \leq N$. All g_1 packets will be dropped by source (node 1) because source itself generates INFO. Let m_i be the number of packets that reach node i for $1 \leq i \leq N$. m_1 is 0 because node 1 is the source of INFO and it does not accept packets containing INFO from any other node. We assume that all the packets an adversary sends are destined for the destination, i.e., node N . We also assume that all the nodes on the path forward packets to their successor, i.e., internal nodes do not cheat. For the purpose of this analysis we assume that adversary only tries to infuse INFO packets into the network. It does not masquerade as any other node.

Lemma 1: If node i runs IASN and no node $j < i$ runs IASN, then the probability of detecting a forged packet at node i is $g_i / \sum_{j=2}^i g_j$.

Proof: Out of nodes 2 through i only node i runs IASN. It receives g_i packets from its neighbors except its predecessor. It will detect and drop all those packets, as those are forged packets sent by adversary. The number of packets that reach node i is $m_i = f + \sum_{j=2}^i g_j$. Only f valid packets are sent by the source.

Therefore the total number of malicious packets that reach node i is $m_i - f$. Hence the probability of

$$\text{detecting a forged packet} = g_i / (m_i - f) = g_i / \sum_{j=2}^i g_j. \blacksquare$$

Lemma 2: Let $Q = \{X_1, X_2, \dots, X_p\}$ be the set of nodes that run IASN such that $X_j \leq i-1$ and $X_i \neq X_j$ for $i \neq j$ and $2 \leq j \leq p$. Then the probability of detecting a forged packet at node i that runs IASN is

$$g_i / (g_i + \sum_{\substack{j=2 \\ j \in Q}}^{i-1} g_j).$$

Proof: Node i will drop g_i packets. The number of packets that reach node i is $m_i = f + g_i + \sum_{\substack{j=2 \\ j \in Q}}^{i-1} g_j$. Out

of these only f valid packets are sent by the source. Therefore the total number of malicious packets that reach node i is $m_i - f$. It follows that the probability of detecting a forged packet = $g_i / (m_i - f) =$

$$g_i / (g_i + \sum_{\substack{j=2 \\ j \in Q}}^{i-1} g_j). \blacksquare$$

Corollary 3: When all the nodes on the route (except source which does not have to run IASN) run IASN then the probability of detecting a forged packet is 1.

Proof: This is a special case of lemma 2 with $p=N-1$ and $i=N$. Since all nodes $j < i$ run IASN, $\sum_{\substack{j=2 \\ j \in Q}}^{i-1} g_j$ is

0. The corollary now follows. \blacksquare

Lemma 4: Let $Q = \{X_1, X_2, \dots, X_p\}$ for $1 < p \leq N$ be the set of nodes that run IASN on a route from node 1 to node N . Then the probability of detecting a forged packet destined for node N on the route is

$$(g_1 + \sum_{\substack{i=2 \\ i \in Q}}^N g_i) / \sum_{i=1}^N g_i.$$

Proof: Node 1 will drop g_1 packets. Each of the node i that runs IASN will drop g_i packets. Therefore the total number of packets dropped is $(g_1 + \sum_{\substack{i=2 \\ i \in Q}}^N g_i)$. Adversary sends $\sum_{i=1}^N g_i$ packets. The probability of

$$\text{detecting a forged packet on the route from } 1 \text{ to } N \text{ is } (g_1 + \sum_{\substack{i=2 \\ i \in Q}}^N g_i) / \sum_{i=1}^N g_i. \blacksquare$$

5.5 Simulation Results

We simulated IASN protocol using ns2 [17]. For our simulations we considered an area whose boundary is defined as 100m x 100m. We tested IASN with two routing protocols DSDV and DSR. For each routing protocol we considered two types of topologies: fixed and random to simulate regular versus irregular (ad hoc) placements of sensor nodes. This results in four scenarios. In fixed topology 100 nodes are arranged in a 10 x 10 grid and are uniformly distributed over the area. In random topology, we placed the nodes randomly in the 100m x 100m area. This experiment was repeated for 1 to 100 nodes that run IASN protocol for all four scenarios. For all four scenarios an adversary is in one corner and a node under attack is in a diagonally opposite corner. Node under attack is receiving four different *informations* from four sources. When adversary manages to forward forged packet to any node that does not run IASN then that packet reaches the node under attack and we cannot detect it. So adversary wins when it finds any

such node. We varied the number of nodes that are running IASN protocol. The nodes that run IASN protocol are selected randomly. The experimental results show that the number of packets that are detected increases as the number of nodes that run IASN protocol increased. This was observed for both the routing protocols DSDV and DSR.

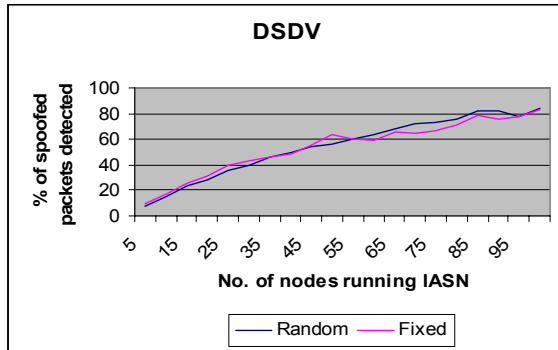


Figure 14: IASN with DSDV.

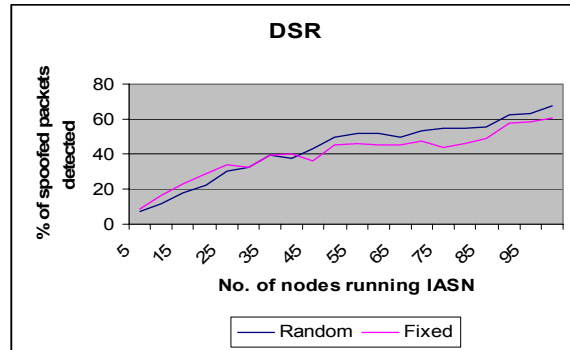


Figure 15: IASN with DSR.

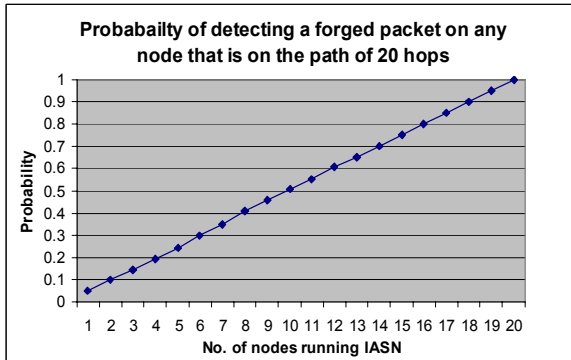


Figure 16: Probability of detecting a forged packet on a path.

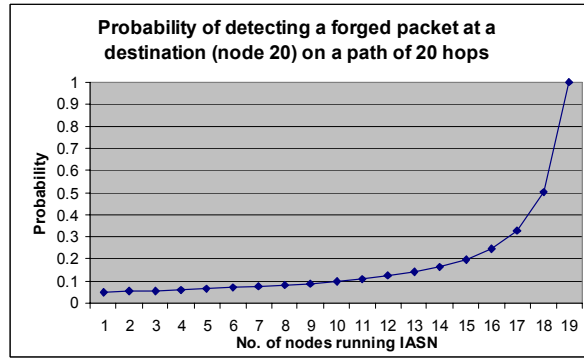


Figure 17: Probability of detecting a forged packet at a destination.

From our analysis we found that the probability of detecting a forged packet on a path is proportional to the number of nodes that run IASN (Figure 16 and lemma 4). The probability of detecting a forged packet at a destination is as shown in figure 17 (lemma 2).

5.6 Keeping multi-hop information

So far we were associating INFO with immediate neighbor. Let us consider a scenario where a node runs IASN and it keeps information about 2 predecessors that are forwarding INFO. In general if a node keeps track of k predecessors that are forwarding INFO then we call it a k -hop IASN. Figure 18 illustrates various scenarios of multi-hop IASN (i.e. k -hop IASN). We will see later that there is a tradeoff between k , storage space and number of nodes that run multihop IASN. In figure 18(a) node 5 runs 1-hop IASN and it drops the packets forwarded by neighbors of node 5 except its predecessor. In figure 18(b) node 5 runs 2-hop IASN. In this case it drops packets forwarded by neighbors of node 5 except its predecessor (node 4) and packets forwarded by neighbors of node 4 except its predecessor (node 3). In figure 18(c) nodes 4 and 5 run 2-hop IASN. In this case they detect forged packets forwarded by neighbors of node 3, 4 and 5 except their predecessors (2, 3 and 4 respectively). Note that even if we run 1-hop IASN at node 5 (and node 4 still runs 2-hop IASN) the same number of packets will be detected. This implies that running 2-hop IASN on successive nodes is same as running 2-hop IASN on the first node and 1-hop IASN on the second node. Figure 18(d) implies that running 2-hop IASN on alternate nodes is same as running 1-hop IASN on every node. But with this approach we have to send *ids* of 2 nodes in a packet. So we double the number of *ids* that we send in a packet. We also generate and store ADTs on only half of the nodes compared to a 1-hop approach. We have to update packet only on alternate nodes instead of every node.

The above approach can be easily generalized for k -hops. If nodes that are k -hop apart run k -hop IASN then it is same as running 1-hop IASN on all nodes.

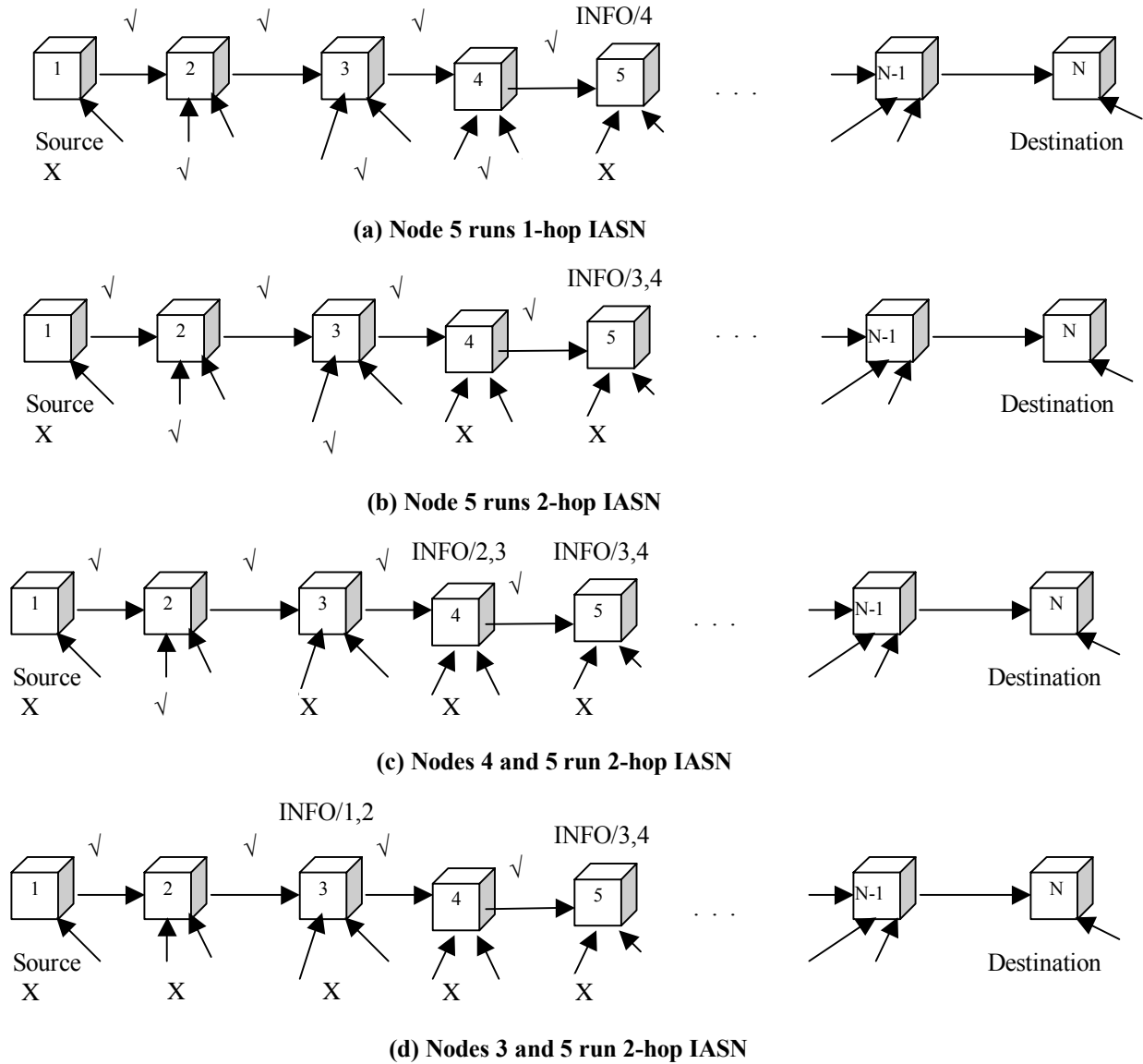


Figure 18: Multi-hop IASN.

Legend: ✓ are neighbors whose packets will be accepted by nodes running IASN and X are neighbors whose packets will be dropped.

Lemma 5: If node i runs 2-hop IASN and no node $j < i$ runs IASN or 2-hop IASN, then the probability of detecting a forged packet at node i is $(g_i + g_{i-1}) / \sum_{j=2}^i g_j$ for $2 \leq i \leq N$.

Proof: Out of nodes 2 through i only node i runs 2-hop IASN. Node i receives g_i packets from its neighbors except its predecessor node $i-1$. Node $i-1$ receives g_{i-1} packets from its neighbors except its predecessor $i-2$. Node i will detect and drop $(g_i + g_{i-1})$ forged packets. The number of packets received by

node i is $f + \sum_{j=2}^i g_j \mid 2 \leq i \leq N$. Out of these packets only f valid packets are sent by the source.

Therefore the number of forged packets that reach node i is $\sum_{j=2}^i g_j$. The probability of detection=

$$(g_i + g_{i-1}) / \sum_{j=2}^i g_j \cdot \mathbf{1}$$

6 Application layer

6.1 Round trip time

At application layer, round trip time can be used for intrusion detection for a bi-directional communication. We associate round trip time with a neighbor. If the round trip time for some neighbor is not in an estimated range, that neighbor can be flagged. Similar to RSSI technique, there are many factors like background noise, weather conditions etc. that can lead this approach to produce large number of false positives. Hence this approach should be used in combination with others.

6.2 Mutual Guarding

Another approach that can then be used at application layer involves nodes, which guard each other from an adversary that tries to masquerade as one node to another. We use the broadcast medium (which is a weakness as far as security is concerned) to our advantage. In the following subsection we explain this novel approach for 2 nodes. Throughout the rest of the paper, for simplicity and clarity, we assume that nodes have omnidirectional antennas and their transmission area is a circle of radius R .

6.2.1 Two nodes guard each other:

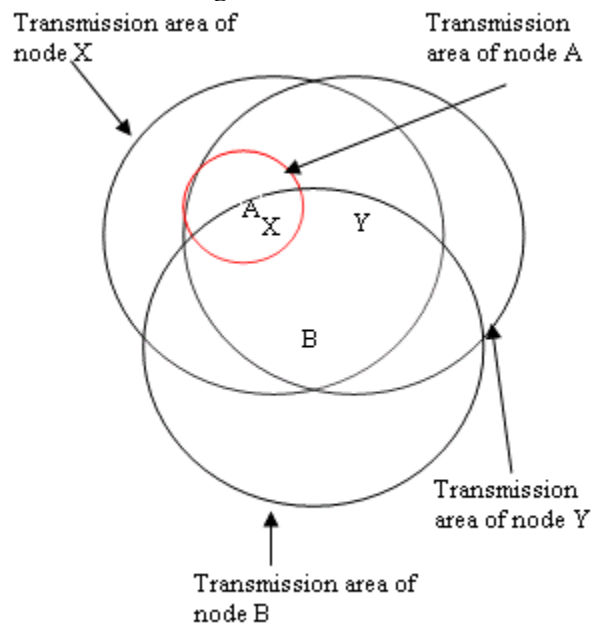


Figure 19: Two nodes X and Y that are distance r apart have transmission range of R .

Two nodes X and Y (of communication range R) have discovered that they are neighbors and hence they can receive packets from each other. An adversary of range R has to be inside the area occupied by two circles to communicate with either node (see figure 19 where node B is such an adversary). If adversary sends a packet to Y with source field set to X from the common area occupied by 2 circles, then X will receive that packet as well. X can thus detect that someone is trying to masquerade as him. If adversary is in the common area occupied by two circles then masquerade can be detected. However note that we cannot detect an adversary (even if it sends from the common area), if it has a very small range because it can go very close to one node and send a packet (it is like whispering to someone so that others in the room do not hear anything. In figure 19 node A can be such an adversary). Another node will not be able to listen in on the transmission and hence will not receive the packet. Let r be the distance between X and Y. The common area is $2R^2 (\cos^{-1}(r/2R) - (r/2R)(1-(r/2R)^2)^{0.5})$. Total area occupied by two circles is $2\pi R^2 - 2R^2 (\cos^{-1}(r/2R) + (r/2R)(1-(r/2R)^2)^{0.5})$. If adversary chooses the location randomly then the adversary is equally likely to be anywhere in the area occupied by the two circles.

Therefore the probability of detection is $2R^2 (\cos^{-1}(r/2R) - (r/2R)(1-(r/2R)^2)^{0.5}) / (2\Pi R^2 - 2R^2 (\cos^{-1}(r/2R) + (r/2R)(1-(r/2R)^2)^{0.5}))$. Ratio r/R can take values form 0 to 1. Figure 20 shows the probability of detection against the ratio r/R . If two nodes are very close to each other, the probability of detection is very high. Similarly if they are far apart the probability of detection becomes very low. This also gives insights on how to place nodes in a WSN. Adversary of range greater than R may go undetected because it can position itself such that it is able to communicate with only X or Y. In the next subsection, we show how the above idea can be extended for three nodes.

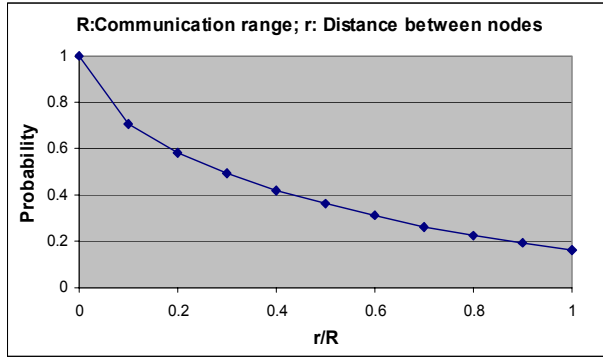


Figure 20: Probability of detection against r/R

other. Similar to 2 nodes case, adversary of very small communication range will go undetected. Sometimes we cannot detect adversary of range greater than R because it can position itself in such a way that the packets it sends are received by only one node.

6.2.3 Detection of masquerade using triangulation: The techniques discussed above can be extended for 4 nodes. The triangulation technique for detecting masquerade can be used as shown in figure 22. Let R be the radio range of all the nodes. Suppose node X has discovered that A, B and C are its neighbors. A, B and C are placed in such a way that the area in which they can transmit, completely occupy the area in which X can transmit. So if adversary has range less than or equal to R and it wants to send a packet to X, it must be in the inner circle. One or two neighbors of X will receive that packet. A, B and C have

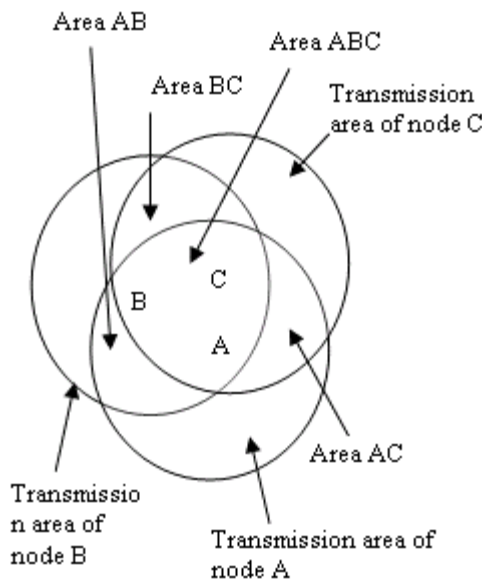


Figure 21: Three nodes A, B and C

neighbor information and they know that they are not reachable from each other. So if A gets a packet with source field set to B or C then A will guess that it is a forged packet. If adversary tries to masquerade as A from area-A then A will receive that packet and will guess that there is an intrusion and someone is trying to masquerade as him. In some cases 2 nodes will detect masquerade. If adversary tries to send from area-AC, node A and C can detect it, similarly for area-BC and area-AB. Total area occupied by 4 nodes is $(\Pi + 3^{3/2}) * R^2$. The area where packets from the adversary will be detected is ΠR^2 . Therefore the probability of detection is $\Pi / (\Pi + 3^{3/2}) = 0.3768$. Probability of false positives is 0 and the probability of false negatives=0.6232.

6.2.2 Three nodes guard each other: If three nodes are in the communication range of each other then they can guard each other from a masquerade attack better (see figure 21). For example if an adversary tries to send a packet to B by changing the source field to A from an area AB then A will receive that packet as well and can detect an intrusion. But adversary can masquerade as C from some part of an area AB. If an adversary sends a packet from area AB, area BC or area AC, it will be received by at least two nodes. Those nodes can guard each

other. Similar to 2 nodes case, adversary of very small communication range will go undetected. Sometimes we cannot detect adversary of range greater than R because it can position itself in such a way that the packets it sends are received by only one node.

6.2.4 Placement of nodes: The above idea can be extended for a whole network. If we place nodes in such a way that the radio range of inner nodes is completely covered by surrounding nodes, then inner nodes are secured against masquerade attack. The nodes on the boundary are not completely covered, so

they are not completely guarded. Let N be the number of nodes. Let n_i be the set of neighbors any node i has for $1 \leq i \leq N$. If node i receives any packet with source field set to x such that $x \notin n_i$, then node i can drop that packet. If a node receives a packet with source field set to its own id then it can conclude that someone is trying to masquerade as him. If any node drops a packet that does not come from its immediate neighbor then a node can prevent masquerade. If there is a mechanism to securely inform other

nodes about the intrusion then the attempts of masquerade can be prevented.

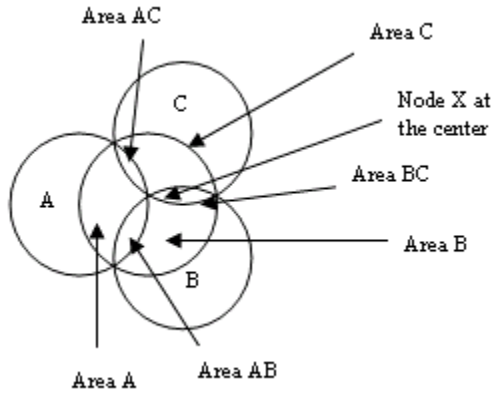


Figure 22: 4 nodes with transmission range R

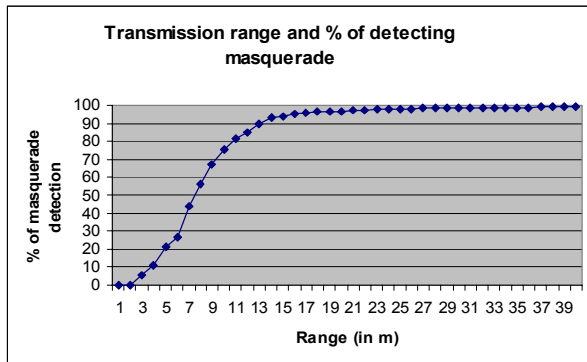


Figure 23: Effect of transmission range on percentage of detection of masquerade.

For simulation of the above idea, we randomly placed 100 nodes in an area 100m x 100m. We assume that the adversary has the same transmission range as all the nodes. Every node is equally likely to be masqueraded. For each node, adversary finds out its neighbors and tries to masquerade as its neighbor. For each node, we place adversary at 100 random locations from where it can transmit a message to that node. It tried to attack all the nodes. Figure 24 shows the percentage of detecting masquerade attack across the network against the transmission range of nodes. It is obvious that more peers guard each node if transmission range is greater. Therefore the percentage of detection increases with increase in transmission range. But a larger transmission range consumes more energy for transmission. Therefore higher percentage of detection can be achieved at the cost of more consumption of energy due to larger transmission range. The above technique also involves the cost of overhearing. Our future work involves proposing sleep wakeup schedules for nodes to minimize overhearing while achieving the same percentage of detection.

7 Discussion and Conclusion

We propose lightweight techniques that detect anomalies at all layers of a network stack in a sensor network. We assumed that the adversary comes into play after deployment and does not interfere with initial setup. At physical layer we propose using RSSI values of neighbors to detect masquerading. This mechanism generates many false positives due to background noise, weather conditions etc. and hence must be used in combination with other approaches. At MAC layer we propose mechanisms that work for schedule based and newly proposed sleep/wake-up based MAC protocols. At routing layer we propose IASN protocol. We have shown that it can be used for source initiated routing protocols, table driven routing protocols and data dissemination mechanisms like directed diffusion. The probability of detection increases linearly with the number of nodes running IASN. Running k-hop IASN on nodes that are k hops apart is same as running IASN on all nodes with 1-hop information. We have shown that ADTs can be built while discovering routes for source initiated routing protocols, table driven routing protocols as well as data dissemination mechanisms like directed diffusion. The storage overhead of ADT is proportional to the number of neighbors. Approximate overhead of IASN is 1.4 percent of the energy network consumes on DSR path discovery and transmission of packets using that path subsequently. Similar overhead is 1.1

percent for directed diffusion. At application layer we propose a novel idea in which nodes guard each other from masquerade. Notably all the techniques have very low false positive rates except RSSI and round trip time. Mechanisms proposed at physical, MAC and application layer for detecting masquerading should be used concurrently to increase the probability of detection and to decrease the probability of false positives. Investigating interrelationship between the proposed mechanisms and new attacks that could be detected by using different combinations of these mechanisms is part of our future work. The main advantage of all techniques is the low overhead, which makes our approach energy efficient. Table 3 summarizes our results. Our future work also involves extending these techniques for detecting insider abuse.

Layer	Protocols/Techniques for anomaly detection	Use	Overhead	Drawbacks
<i>Physical</i>	RSSI value.	Detects masquerade.	Calibration of RSSI value for each neighbor.	Large number of false positives.
<i>MAC</i>	TDMA: Check if adversary follows TDMA schedule.	Detects masquerade.	Keep track of TDMA schedules of other nodes.	None.
	S-MAC: Check if sender is supposed to be sleeping.	Detects masquerade.	Keep track of sleep-wake up schedules of other nodes.	None.
<i>Routing</i>	For any routing protocol, check if neighbor and the expected information matches.	Guarantees information authentication.	Constructing ADTs. Updating previous hop in a packet.	None.
<i>Application</i>	Use triangulation to detect intrusions.	Detects masquerade.	Nodes always have to listen.	Overhearing.
	Round trip time.	Detects masquerade.	Precise calibration of range of round trip time for each neighbor.	Large number of false positives.

Table 3: Summary of the proposed techniques.

Acknowledgements

We would like to thank Zijiang Yang, Mark Terwilliger and Zille H. Kamal for their comments as we worked on initial part of this paper. Research is supported in part by the National Science Foundation, under grants ACI-0000442, ACI-0203776, and MRI- 0215356, by the Department of Education grant R215K020362, and a Congressional Award, administered by the US Department of Education, Fund for the Improvement of Education. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies or institutions.

References

- [1] Anderson, J. P., "Computer Security Threat Monitoring and Surveillance", Technical report, James. P. Anderson Co., Fort Washington, Pennsylvania, 1980.
- [2] S. Bhargava and D. Agrawal, "Security enhancements in AODV protocol for wireless ad hoc networks", in Proceedings of Vehicular Technology Conference, 2001.
- [3] Bhuse V., Terwilliger M., Gupta A., Kamal Z. and Yang Z., "Using routing data for Information authentication in sensor networks", 3rd International Trusted Internet Workshop, HiPC, December 2004.
- [4] Denning, D., "An Intrusion Detection Model", IEEE Transactions on Software Engineering, 13(2):222—232, 1987.
- [5] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling", Proceedings of the IEEE, 75(1):56-73, Jan. 1987.
- [6] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", Mobile Computing and Networking, pages 56-67, 2000.
- [7] Johnson, D. and Maltz, D. (1996). "Dynamic source routing in ad hoc wireless networks," Mobile Computing (ed. T. Imielinski and H. Korth), Kluwer Academic Publishers, Dordrecht, The Netherlands.

- [8] Kumar, S., "Classification and detection of computer intrusions", PhD thesis, Purdue University, 1995.
- [9] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking, 2000.
- [10] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks", IEEE Wireless Communications, vol. 11, no. 1, pp. 48-60, Feb 2004.
- [11] C. E. Perkins, "Ad-hoc on-demand distance vector routing," in MILCOM '97 panel on Ad Hoc Networks, Nov. 1997.
- [12] C. Perkins and P. Bhagwat. "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", proceedings of the ACM SIGCOMM, October 1994.
- [13] E.M. Royer and C-K Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", IEEE Personal Communications, Apr. 1999.
- [14] <http://www.tinyos.net/>
- [15] Wei Ye Heidemann, J. Estrin, D., "An energy-efficient MAC protocol for wireless sensor networks", INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies.
- [16] Y. Zhang and W. Lee, "Intrusion detection in wireless ad hoc networks," ACM MOBICOM, 2000.
- [17] Network simulator, University of California, Berkeley, 1997, <http://www.isi.edu/nsnam/ns/>.
- [18] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in The 8th ACM International Conference on Mobile Computing and Networking, September 2002.