Stealthy Attacks in Wireless Ad Hoc Networks: Detection and Countermeasure

Issa Khalil and Saurabh Bagchi

Abstract– Stealthy packet dropping is a suite of four attacks—misrouting, power control, identity delegation and colluding collision—that can be easily launched against multihop wireless ad hoc networks. Stealthy packet dropping disrupts the packet from reaching the destination through malicious behavior at an intermediate node. However, the malicious node gives the impression to its neighbors that it performs the legitimate forwarding action. Moreover, a legitimate node comes under suspicion. A popular method for detecting attacks in wireless networks is behavior-based detection performed by normal network nodes through overhearing the communication in their neighborhood. This leverages the open broadcast nature of wireless communication. An instantiation of this technology is local monitoring. We show that local monitoring, and the wider class of overhearing-based detection, cannot detect stealthy packet dropping attacks. Additionally, it mistakenly detects and isolates a legitimate node. We present a protocol called SADEC that can detect and isolate stealthy packet dropping attack efficiently. SADEC presents two techniques that can be overlaid on basic local monitoring: having the neighbors maintain additional information about the routing path, and adding some checking responsibility to each neighbor. Additionally, SADEC provides an innovative mechanism to better utilize local monitoring by considerably increasing the number of nodes in a neighborhood that can do monitoring. We show through analysis and simulation experiments that basic local monitoring fails to efficiently mitigate most of the presented attacks while SADEC successfully mitigates them.

Keywords– Local monitoring, misrouting, multi-hop wireless networks, packet dropping, transmission power control.

1 INTRODUCTION

Wireless Ad hoc and Sensor Networks (WASN) are becoming an important platform in several domains, including military warfare and command and control of civilian critical infrastructure [33][34]. They are especially attractive in scenarios where it is infeasible or expensive to deploy significant networking infrastructure. Examples in the military domain include monitoring of friendly and enemy forces, equipment and ammunition monitoring, targeting, and nuclear, biological, and chemical attack detection [33][34]. Consider a military network scenario where more powerful and less energy constrained ad hoc nodes may be carried by soldiers or in vehicles, while a large number of low cost and low-energy sensor nodes with limited energy resources may be distributed over the battlefield. This network setup can guide a troop of soldiers to move through the battle field by detecting and locating enemy tanks and troops. The soldiers can use information collected by the sensor nodes to strategically position to minimize any possible causality. Examples in the civilian domain include habitat monitoring, animal tracking, forest-fire detection, disaster relief and rescue, oil industry management, and traffic control and monitoring [33][35].

However, the open nature, the fast deployment practices, and the hostile environments where WASN may be deployed, make them vulnerable to a wide range of security attacks against both control and data traffic. Moreover, many WASN such as sensor networks are resource-constrained, primarily with respect to energy and bandwidth. Thus any security protocol needs to obey these constraints as well. Control traffic attacks include wormhole [5], rushing [4], and Sybil [10] attacks. The most notable data traffic attacks are blackhole, selective forwarding, and delaying of packets, in which respectively a malicious node drops data (entirely or selectively) passing through it, or delays its forwarding, and misrouting attack in which the attacker relays packets to the wrong next-hop which has the effect that the packet is indirectly dropped. These attacks could result in a significant loss of data or degradation of network functionality, say through disrupting network connectivity by preventing route establishment.

Cryptographic mechanisms alone cannot prevent these attacks since many of them, such as the wormhole and the rushing attacks, can be launched without needing access to cryptographic keys or violating any cryptographic check. To mitigate such attacks, many researchers have used the concept of behavior-based detection which is based on observing patterns in the behavior of neighboring nodes and flagging anomalous patterns. The notion of behavior is related to communication activities such as forwarding packets (e.g., [6]) or non-communication activities such as reporting sensed data (e.g., [37]). A widely used instantiation of behavior-based detection is *Local Monitoring* (e.g., [1],[2],[6]-[8],[24],[28]). In local monitoring, nodes oversee part of the traffic going in and out of their neighbors. This leverages the open broadcast nature of wireless communication. Different types of checks are done locally on the observed traffic to make a determination of malicious behavior. For example, a node may check that its neighbor is forwarding a packet to the correct next-hop node, within acceptable delay bounds. For systems where arriving at a common view is important, the detecting node initiates a distributed protocol to disseminate the alarm. We call the existing approaches which follow this template *Baseline Local Monitoring* (*BLM*). Many protocols have been built on top of BLM for intrusion detection (e.g., [3]), building trust and reputation among nodes (e.g. [1], [2], [23], [37]), protecting against control and data traffic attacks (e.g. [6]-[8]) and in building secure routing protocols (e.g., [8], [9]).

For specificity, we will use [6]-[8] as the representative BLM which we will use for comparison with the approach presented in this paper. In BLM, a group of nodes, called *guard nodes* perform local monitoring with the objective of detecting security attacks. The guard nodes are normal nodes in the network and perform their basic functionality in addition to monitoring. Monitoring implies verification that the packets are being faithfully forwarded without modification of the immutable parts of the packet, within acceptable delay bounds and to the appropriate next hop. If the volume of traffic is high (say for data

- *Issa Khalil is with College of Information Technology, United Arab Emirates University, UAE. Email: ikhalil@uaeu.ac.ae.*
- *Saurabh Bagchi is with Dependable Computing Systems Lab (DCSL), School of Electrical & Computer Engineering, Purdue University, USA. Email: sbagchi@purdue.edu.*

Digital Object Indentifier 10.1109/TMC.2010.249 1536-1233/10/$26.00 © 2010 IEEE

traffic in a loaded network), a guard node verifies only a fraction of the packets.

In this paper, we introduce a new class of attacks in wireless multi-hop ad hoc networks called *stealthy packet dropping*. In stealthy packet dropping, the attacker achieves the objective of disrupting the packet from reaching the destination by malicious behavior at an intermediate node. However, the malicious node gives the impression to its neighbors participating in local monitoring that it has performed the required action (e.g., relaying the packet to the correct next-hop *en route* to the destination). This class of attacks is applicable to packets that are neither acknowledged end-to-end (e.g., [30]) nor hop-by-hop (e.g., [39]). Due to the resource constraints of bandwidth and energy, much traffic in multi-hop ad hoc wireless networks is *unacknowledged* or only selectively acknowledged ([29][30][39]). This is particularly true for the more common data traffic or broadcast control traffic than for rare unicast control traffic.

In this paper, we introduce four modes of the stealthy packet dropping attack. We distinguish between an *external malicious node*, which does not possess the cryptographic keys in the network, and an *internal compromised node*, which does and is created by compromising an erstwhile legitimate node. Consider a scenario in which a node called *S* is forwarding a packet to a compromised node called *M*. *M* is supposed to relay the packet to the next-hop node *D*. The first form of the attack is called *packet misrouting*. In this mode, *M* relays the packet to an incorrect next-hop neighbor. The result is that the packet does not reach its intended next-hop (*D*) while *M* appears to the guards as doing its forwarding job correctly. The second mode is called the *power control* attack. In this mode, *M* controls its transmission power to relay the packet to a distance less than the distance between *M* and *D*. Therefore, the packet does not reach the next-hop while the attacker avoids detection by many guards. The third form of the attack is called the *colluding collision* attack. In this mode, the attacker uses a colluding node (external or internal) in the range of *D* to transmit data at the same time when *M* starts relaying the packet to *D*. Therefore, a collision occurs at *D*, which prevents the packet from being correctly received by *D*, while *M* appears to be performing its functionality correctly. The final mode of stealthy packet dropping is called the *identity delegation* attack. In this mode, the attacker colludes with a node *E* placed close to the source node *S*. *E* is allowed to use *M*'s identity and transmit the packet. Since *E* is almost at the same place as *S*, *D* does not receive the packet while the guards of *M* are deceived that *M* relays the packet to the next-hop. In each of these attack types, the adversary can successfully perform the attack without detection through BLM. Additionally, in each attack type, a legitimate node is accused of packet dropping. We acknowledge that the attack model calls for smart adversaries—e.g., they can collude, can position the adversarial nodes, can control transmission power at a fine level of granularity, or can spend significant energy in launching the attacks. On the other hand, note that these attacks are not hard to mount for motivated attackers since the requirement for successful instantiation of any of these attacks is fairly humble (Table 1) and practically viable [35]. Therefore, we believe that if the network is critical enough, we *do* have to worry about such motivated adversaries.

We provide a protocol called SADEC (**S**tealthy **A**ttacks in Wireless Ad Hoc Networks: **De**tection and **C**ountermeasure) that is built using local monitoring and that can mitigate each of the four attack types introduced above. The SADEC detection technique involves two high-level steps: *first*, having guard nodes maintain additional next-hop information gathered during route establishment; and *second*, adding some checking responsibility to each neighbor. The latter technique makes use of the fact that under three of the attacks, neighbors have differing views of a node in terms of amount of forwarding traffic generated by the node. Hence, a single one-hop broadcast cannot convince *all* the neighbors. On the other hand, we show that of the four modes of the stealthy packet dropping attack, BLM is unable to detect any instance of three attack types (all except drop through power control) while it is able to detect specific instances of the drop through power control attack, depending on how the adversary constrains the range of the forwarded packet. Any work that relies on BLM for building higher level knowledge (such as, reputation scores as in [2][23]) would suffer from the disadvantage of BLM against stealthy packet dropping.

We provide a theoretical analysis for the probability of success of the stealthy packet drop attack in a locally monitored network. We also analyze the resource consumption cost of SADEC. Our analysis shows that SADEC maintains detection coverage above 90 % for the configuration in which BLM has less than 60% coverage. Moreover, the legitimate node isolation of SADEC remains below 2% for the configurations in which BLM exceeds 99%. Additionally, we build a simulation model for both the power control and the misrouting attacks using ns-2 and perform a comparative evaluation of BLM with SADEC. Our simulation results show that SADEC can deliver 60% of packets to the destination under 20% nodes compromised launching misrouting attack, while BLM delivers less than 10%. The likelihood of framing of legitimate nodes is also 18-fold reduced with SADEC compared to BLM for the same network. The performance advantages under misrouting attack come at the expense of a slightly higher false isolation (due to natural collisions on the channel) and end-to-end delay in SADEC. Under the power control attack, the isolation probability in SADEC remains around 70%, while it drops to below 45% in BLM.

We summarize our contributions in this paper as follows:

1. We introduce the stealthy packet dropping class of attacks and detail four methods by which it can be launched in locally-monitored networks. In this class of attacks, the malicious node evades detection and a legitimate node is mistakenly deemed malicious.

2. We provide a protocol called SADEC to remedy each attack type with minimal addition to the resource consumption and responsibility of a node over BLM.

3. We show through analysis and simulations the security advantage of SADEC over BLM for two of the four attack types – misrouting and power control.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 provides the foundations and background knowledge. Section 4 describes the stealthy packet dropping attack. Section 5 describes SADEC and presents its mitigation techniques. Sections 6 and 8 present the mathematical analysis and the simulation results respectively. Finally, Section 9 concludes the paper.

2 RELATED WORK

In the last few years, researchers have been actively exploring many mechanisms to ensure the security of control and data traffic in wireless networks. These mechanisms can be broadly categorized into the following classes–authentication and integrity services, protocols that rely on path diversity, protocols that use specialized hardware, protocols that require explicit acknowledgements or use statistical methods, protocols that overhear neighbor communication.

The path diversity techniques increase route robustness by first discovering multi-path routes [9], [13] and then using these paths to provide redundancy in the data transmission between a source and a destination. The data is encoded and divided into multiple shares sent to the destination via different routes. The method is effective in well-connected networks, but does not provide enough path diversity in sparse networks. Moreover, many of these schemes are expensive for resource-constrained wireless networks due to the data redundancy. Additionally, these protocols could be vulnerable to route discovery attacks, such as the Sybil attack, that prevent the discovery of non-adversarial paths.

Examples of protection mechanisms that require specialized hardware include [5] and [11]. The authors in [5] introduce a scheme called *packet leashes* that uses either tight time-synchronization or location awareness through GPS hardware. The work in [11] relies on hardware threshold signature implementations to prevent one node from propagating errors or attacks in the whole network.

A technique proposed to detect malicious behavior involving selective dropping of data, relies on explicit acknowledgement for received data using the same channel [13], or an out-of-band channel [12]. This method would render stealthy packet dropping detectable at the end point. However, the method incurs high communication overhead and has to be augmented with other techniques for diagnosis and isolation of the malicious nodes. A natural extension would be to reduce the control message overhead by reducing the frequency of ack-ing to one in every N data messages (in the above papers N=1). However, this may delay the adversary detection which may result in significant damage. In contrast, in SADEC, the node is detected and diagnosed locally by its neighbors. Statistical measures have been used by some researchers for detection, e.g., [14] to detect wormhole attacks.

The issue of trust in ad-hoc networks has been looked at by many researchers (e.g., [1][2][23][37]). They all use Dempster-Shafer belief theory [38] for incorporating second-hand information (reports by other nodes) to create a reputation score of a node. Many reputation-based approaches (e.g., [37]) suffer from poor protection against ballot stuffing (i.e., a colluding malicious node praising another malicious node) or bad-mouthing (i.e., a malicious node implicating a legitimate node). All the reputation-based approaches are susceptible to behavior where a node functions correctly but provides wrong information about another node. Moreover, all the approaches can suffer from non convergent behavior whereby the reputation of a good node gets stuck at a low value or that of a malicious node is falsely elevated

A widely used technique for mitigating control and data forwarding misbehavior in multi-hop wireless networks is cooperative local monitoring ([3], [6]-[8], [13]). The work in [13] provides a mechanism to discover routes with certain desirable properties, such as being node disjoint. The work in [8] provides detection of a wide class of control attacks against static sensor networks. However, all the behavior-based mechanisms (both communication-based and non-communication-based) as used by all researchers to date, fail to mitigate the stealthy packet dropping attack presented in this paper.

This paper builds on our previous work [31]. In [31], we introduced the stealthy packet dropping attacks and proposed a protocol called MISPAR to mitigate the attacks. In this paper, we quantify the likelihood of mistaken isolation of legitimate nodes due to both natural errors and framing. We also present a thorough analysis of legitimate and malicious node isolation probabilities for both BLM and SADEC under the misrouting attack (in addition to that under the power control attack, which is also present in [31]). Furthermore, this paper provides a wide range of simulation experiments to evaluate the performance of both BLM and SADEC under the misrouting attack and the transmission power control attacks. Finally, this paper presents the results of a test-bed experiment using 50 Mica2 motes built to evaluate the overhead of SADEC and its feasibility for resource limited sensor networks.

3 FOUNDATIONS

3.1 Attack Model and System Assumptions

Attack model: An attacker can control an external node or an internal node, which, since it possesses the keys, can be authenticated by other nodes in the network. An insider node may be created, for example, by compromising a legitimate node. A malicious node can perform packet dropping by itself or by colluding with other nodes. The collusion may happen through out-of-band channels (e.g., a wireline channel). However, we do not consider the denial of service attacks through physical-layer jamming [22], or through identity spoofing and Sybil attacks [10]. There exist several approaches to mitigate these attacks – [22] for jamming and [10] for the Sybil attack. A malicious node can be more powerful than a legitimate node and can establish out-of-band fast channels (e.g., a wireline link) or have high-powered controllable transmission capability. The attacks do not affect only a specific routing protocol; rather, they apply to a wide class where the requirement is an intermediate node determines the next hop node toward the final destination. This includes routing protocols specific to WSNs such as the beacon routing protocol.

System assumptions: We assume that all the legitimate communication links are bi-directional. We assume that secure neighbor discovery has been performed and that every node knows both first and second hop information. This can be achieved through the protocol described in [21] as well as by approaches developed by other researchers [4]. Note that while this knowledge is enormously useful, this by itself cannot mitigate many attack types. For example, further work is needed to detect the wormhole attack. Intuitively this information subsets the nodes from which a given node will accept packets but does not eliminate the possibility of malicious nodes within that subset. Local monitoring assumes that the network has sufficient redundancy, such that each node has more than an application defined threshold number of legitimate nodes as guards. We assume a key management protocol, e.g., [15], exists such that any two nodes can communicate securely. We present SADEC for static networks. However, the technique is also valid under mobile situations after adaption to address mobility challenges. One of these challenges is the problem of determining the neighbor relation securely. Several such protocols exist in the literature [5][25]-[27]. Additional challenges that need to be addressed include time synchronization and the ability to distinguish between malicious and natural errors which become more frequent due to mobility.

3.2 Background: Local Monitoring

Local monitoring is a collaborative detection strategy where a node monitors the control traffic going in and out of its neighbors. This strategy was introduced in [6] for *static sensor* networks and here we give the background needed to understand the concepts presented in this paper.

For a node, say α, to be able to watch a node, say N_2, α must be a neighbor of both N_2 and the previous hop from N_2, say N_1. Then we call α a *guard* node for N_2 over the link $N_1 \rightarrow N_2$. We use the notation $R(N)$ to denote the set of all nodes that are within the radio range of node N and $G(N_1, N_2)$ to denote the set of all guard nodes for N_2 over a link $N_1 \longrightarrow N_2$. Formally, $G(N_1, N_2) = R(N_1) \cap R(N_2) - N_2$, where $N_2 \in R(N_1)$.

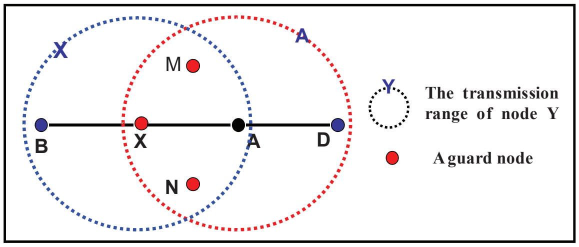

Figure 1: X, M, and N are guards of A over X→A

For example, in Figure 1, $G(X,A)=\{M,N,X\}$. Information from each packet sent from X to A is saved in a *watch buffer* at each guard. The guards expect that A will forward the packet toward the ultimate destination, unless A is itself the destination. Each entry in the watch buffer is time stamped with a time threshold, τ, by which A must forward the packet. Each packet forwarded by A with X as a previous hop is checked for the corresponding information in the watch buffer. The check can be to verify if the packet is fabricated or duplicated (no corresponding entry in the buffer), corrupted (no matching hash of the payload), dropped or delayed (entry is not matched within τ).

A malicious counter ($MalC(i,j)$) is maintained at each guard node, i, for a node, j, at the receiving end of each link that i is monitoring over a sliding window of length T_{win}. $MalC(i,j)$ is incremented for any malicious activity of j detected by i. The increment to $MalC$ depends on the nature of the malicious activity, being higher for more severe infractions. When the growth in the counter value maintained by a guard node i for node j ($MalC(i,j)$) crosses a threshold rate ($MalC_{th}$) over T_{win}, node i revokes j from its neighbor list (called *direct isolation* since it will henceforth not perform any communication with node j), and sends to each neighbor of j, an authenticated alert message indicating j is a suspected malicious node. When a neighbor N_i gets the alert, it verifies the authenticity of the alert message. When N_i gets enough alert messages about j, it marks the status of j as revoked (called *indirect isolation*). The notion of enough number of alerts is quantified by the *detection confidence index* γ. Each node maintains a memory of nodes that it has revoked through a local blacklist so that a malicious node cannot come back to its neighborhood and claim to be blameless. This constitutes *local isolation* of a malicious node by its current neighbors.

4 Stealthy Dropping Attack Description

In all the modes of stealthy packet dropping, a malicious intermediate node achieves the same objective as if it were dropping a packet. However, none of the guard nodes using BLM become any wiser due to the action. In addition, some legitimate node is accused of packet dropping. Next, we describe the four attack types for stealthy packet dropping.

4.1 Drop through Misrouting

In the misrouting attack, a malicious node relays the packet to the wrong next-hop, which results in a packet drop. Note that, in BLM [6], a node that receives a packet to relay without being in the route to the destination either drops the packet or sends a one-hop broadcast that it has no route to the destination. The authors in [6] argue that that latter case would be more expensive and dangerous since it gives malicious nodes valid excuses to drop packets. Therefore, they go with the first choice, even though it may result in some false accusations.

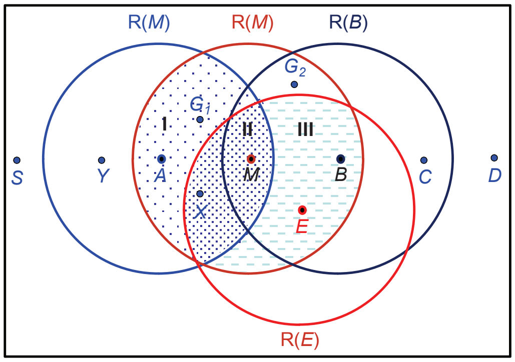

Figure 2: Misrouting scenario

Consider the example scenario in Figure 2. Node A sends a packet to the malicious node M to be relayed to node B. Node M simply relays the packet to node E which is not in the route to the final destination of the packet. Node E drops the packet. The result is twofold: (i) node M successfully drops the packet without being detected since all the guards of M over $A \rightarrow M$ (regions I & II) have been satisfied by the transmission of $M \rightarrow E$, and (ii) legitimate node E will be wrongly accused by its guards over $M \rightarrow E$ (regions II & III) as maliciously dropping the packet.

4.2 Drop through Power Control

In this type of attack, a malicious node relays the packet by carefully reducing its transmission power, thereby reducing the range and excluding the legitimate next-hop node. This kind of transmission power control is available in today's commercial wireless nodes, such as the Crossbow Mica family of nodes.

Consider the scenario shown in Figure 3. A node S sends a packet to a malicious node M to be relayed to node T. Node M drops the packet by sending it over a range that does not reach T (the dotted circle centered at M). Figure 3(a) shows the guards of M that are satisfied by the controlled transmission of M (region II) and the set of guards that detect M (region I) as dropping the packet since they did not overhear M. Figure 3(b) shows all the guards of M over $S \rightarrow M$. Figure 3(d) shows the set of guards of T over $M \rightarrow T$ that wrongly accuse T of dropping the packet. The farther T is from M the better it is for the attacker since more guards can be satisfied and therefore, the stealthier the attack. For this attack to succeed, the attacker must know the location of each neighbor and the detection confidence index γ. Typically security is not achieved through obfuscation and therefore protocol parameters such as γ are taken to be known to all and location determination is routinely run upon deployment of nodes. When the number of guards that are not satisfied by the controlled-power transmission is greater than γ-1, an intelligent attacker will refrain from lowering the transmission power since it will be detected by all its neighbors either directly or indirectly (Section 3.2). Additionally, a successful attack, not only achieves the effect of dropping the packet, but also causes a subset of the guards of T over $M \rightarrow T$ to accuse T of dropping the packet.

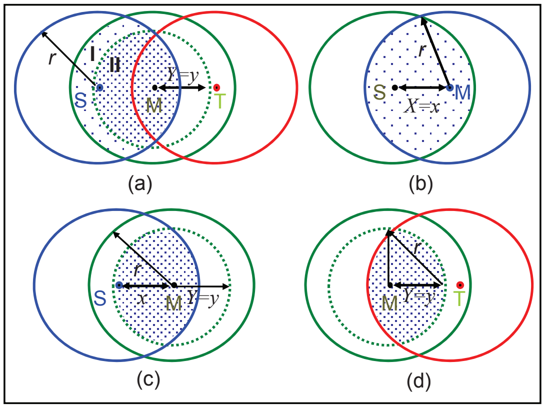

Figure 3: (a) The guards of M over S→M (I & II); (b) Separation between S and M = x; (c) the subset of guards of M over S→M that has been satisfied by the controlled power transmission of M; (d) the subset of guards of T over M→T that wrongly accuses T of dropping the packet

4.3 Drop through Colluding Collision

In many wireless sensor network deployment scenarios, the 802.11 MAC protocol RTS-CTS mechanism that reduces frame collisions due to the hidden terminal problem and the exposed terminal problem are disabled for the sake of energy saving. This is also explained by the fact that packets in some wireless networks such as sensor networks are often quite small and fall below the threshold for packet length for which RTS/CTS is turned on.

The attacker may exploit the absence of the RTS/CTS frames to launch a stealthy packet dropping attack through collision induced by a colluding node. The colluding node creates a collision in the vicinity of the expected next-hop node at an opportune time. Consider the scenario shown in Figure 4. The malicious node M_1 receives a packet from S to be relayed to T. Node M_1 coordinates its transmission with a transmission of some data generated by its colluding partner M_2 to T. It has the effect that T is unable to get the packet relayed by M_1. The damage caused by this attack is twofold: (i) M_1 successfully drops the packet due to a collision at T without being detected, and (ii) node T is accused of dropping the packet by some of its guards over the link $M_1 \rightarrow T$ (the guards that are out of the range of M_2, region I). Note that for M_2 to be able to send data to T, it is has to be a legitimate neighbor (compromised by the attacker), otherwise, the attack would be considered a physical layer jamming [22], which is assumed to be detectable through techniques complementary to that presented in the paper (e.g., [8][22]).

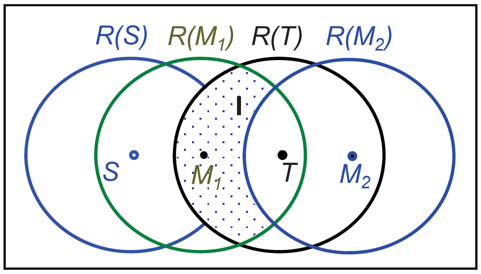

Figure 4: Colluding collision illustration scenario

4.4 Drop through Identity Delegation

In this form of the attack, the attacker uses two malicious nodes to drop the packet. One node is spatially close to the sender. The other node is the next-hop from the sender. The first malicious node could be external or an internally compromised node while the latter has to be an internally compromised node. Consider the scenario shown in Figure 5, node S sends a packet to a malicious next-hop node M_2 to be relayed to node T. The attacker delegates the identity and the credentials of the compromised node M_2 to a colluding node M_1 close to S. After S sends the packet to M_2, M_1 uses the delegated identity of M_2 and transmits the packet. The intended next-hop T does not hear the message since $T \notin R(M_1)$. The guards of M_2 over $S \rightarrow M_2$ are the nodes in the shaded areas I & II and they are all satisfied since they are in $R(M_1)$. Again, the consequences of this attack are twofold: (i) the packet has been successfully dropped without detection, and (ii) the set of nodes in the shaded area II overhear a packet transmission (purportedly) from M_2 to T. These nodes are included in $G(M_2,T)$ and will subsequently accuse T of dropping the packet.

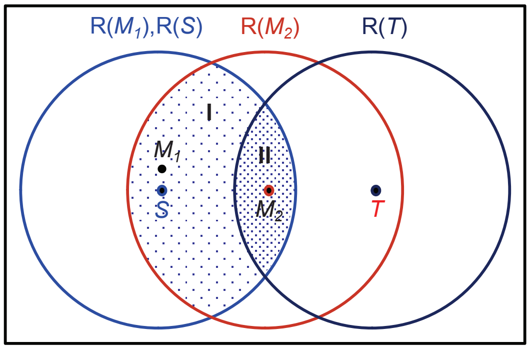

Figure 5: Identity delegation illustration scenario

Table 1 provides a summary of the attacks presented which includes the number of malicious and external attacker nodes required to launch each attack, the special capabilities of each node (if any), and the placement requirement of these nodes.

Table 1: summary of the stealthy attacks

Attack name	Attack description	Attack instantiation requirement
Misrouting	Relays the packet to the wrong next hop	One compromised node in the route between the sender and the receiver.
Power Control	Controls the transmission to exclude next hop	One compromised node in the route between the sender and the receiver with power control capability
Colluding Collision	Simultaneous transmission to create a collision at the next hop	One compromised node in the route between the sender and the receiver and one external attacker node close to the next-hop from the compromised node
Identity delegation	Delegate the relay responsibility to a colluding partner close to the sender	One compromised node in the route between the sender and the receiver and one external attacker node close to the compromised node

5 Stealthy Dropping Attack Mitigation

In this section we propose two mechanisms to augment traditional local monitoring to detect stealthy packet dropping. The first mechanism mitigates the misrouting stealthy packet drop while the second mitigates the rest of the attack types.

5.1 Mitigating Misrouting Packet Drop

To detect this attack, the local monitoring has to incorporate additional functionality and information. The basic idea is to extend the knowledge at each guard to include the identity of the next-hop of the packet being relayed.

This additional knowledge can be collected during route establishment. Many multi-hop wireless routing protocols provide this knowledge without any modification while some changes are necessary in others. The first class includes both reactive routing protocols such as Dynamic Source Routing (DSR) and its variants [16] and proactive routing protocols such as TinyOS beacon routing [18] and Destination Sequenced Distance Vector routing (DSDV [19]). In all source routing protocols, the packet header carries the identity of all the nodes in the route from the source to the destination. Therefore, no additional traffic is required to be generated for the guard nodes to be able to detect this kind of attack. Moreover, no additional information is required to be maintained at the guards since each packet carries the required information in its header. In TinyOS beacon routing, the base station periodically broadcasts a beacon to establish a breadth first search tree rooted at the base station. Each node within the transmission range of the base station overhears the beacon, sets its parent to be the base station, sets the hop count to the base station to be one, and rebroadcasts the beacon. Each beacon carries the identity of the broadcasting node, the identity of its parent, and the hop count to the base station. Each guard overhearing the beacon broadcasting saves parent node identity for each neighbor. Later, when a node, say B, is sent a packet to relay, the guard of B can detect any misrouting by B since it knows the correct next-hop *en route* to the base station.

The second class of routing protocols requires modification to the protocol to build the next-hop information at the guards. Examples of these protocols are the reactive routing protocols that use control packet flooding of route requests (*REQ*) and route replies (*REP*) to establish the route between the source and the destination (e.g., LSR [8] and AODV [17]). In these protocols, when a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a route discovery process to locate the other node. It broadcasts a route request (*REQ*) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located. Along with its own sequence number and the broadcast ID, the source node includes in the *REQ* the most recent sequence number it has for the destination. During the process of forwarding the *REQ*, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. Once the *REQ* reaches the destination, the destination node responds by unicasting a route reply (*REP*) packet back to the neighbor from which it first received the *REQ*. As the *REP* traverses along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the *REP* came.

Next, we show the required changes to the basic version of AODV to enable the guards to build the necessary knowledge for detecting the misrouting attack. The idea behind the solution is that during route establishment, when the relation about which node to forward a packet between a given source-destination pair is determined, this information is broadcast by a neighbor to the guards which will be responsible for monitoring the node. To collect the next-hop identity information, the forwarder of the *REQ* attaches the previous two hops to the *REQ* packet header. Let the previous hop of *M* be *A* for a route from source *S* to destination *D*, and the next hop from *M* be *B* (Figure 2). When *M* broadcasts the *REQ* received from *A*, it includes the identity of *A* and its own identity (*M*) in the *REQ* header *<S, D, REQ_id, A, M>*. When *B* and the other neighbors of *M* get the *REQ* from *M*, they keep in a *Verification Table* (*VT*) *<S, D, RREQ_id, A, M, ->* (last field is currently blank). When *B* broadcasts the *REQ*, the common neighbors of *M* and *B* update their *VT* to include *B <S, D, RREQ_id, A, M, B>*. When *B* receives a *REP* to be relayed to *M*, it includes in that *REP* the identity of the node that *M* needs to relay the *REP* to, which is *A* in this example. Therefore, all the guards of *M* now know that *M* not only needs to forward the *REP* but also that it should forward it to *A*.

Two tasks have been added to the functionality of the guards in monitoring the *REP* packets. First, the guard *G* of a node *N* verifies that *N* forwards the *REP* to the correct next-hop. In the example above, G_2 verifies that *M* forwards the *REP* to *A*. Second, *G* verifies that *N* has updated the forwarded *REP* header correctly. In the example shown above, G_2 verifies that when the input packet to *M* from *B* is *<REP, S, D, REQ_id, C, B, M>*, then the output packet from *M* should be *<REP, S, D, REQ_id, B, M, A>*. Thus *M* and its guards over the link *B*→*M* know that the next-hop is *A* from the information built in the *VT* table during the *REQ* flooding.

Using the additional information mentioned above, SADEC detects misrouting attacks as follows. In the example above, assume that *S* is sending a data packet to *D* through a route that includes *<Y,A,M,B,C>*. The malicious node *M* cannot misroute the data packet received from *A* to a node other than the next-hop, *B* since each guard of *M* over the link *A*→*M* has an entry in its *VT* which indicates *B* is the correct next-hop. This results in an additional checking activity for the guard node involved in local monitoring–verifying the data packet is forwarded to the correct next hop, as indicated by the entry in the guard node's *VT*. Moreover, *M* cannot frame another neighbor, say *X*, by misrouting the packet to *X*. The guards of *X* over *M*→*X* do not have an entry like *< S, D, REQ_id, Y, A, M, X>* and therefore, they would not increment the MalC of *X* when it drops the packet.

5.2 Mitigating other Stealthy Drop Attacks

The key observation behind the other types of the stealthy packet dropping attack is that the attack defeats local monitoring based detection by reducing the number of guards that overhear a packet to zero or to a number that is less than the confidence index γ. In the power control attack shown in Figure 3(a), the attacker narrows the guards that can detect the packet drop into the lightly shaded area (region I in Figure 3(a)) while the majority of the guards (region II in Figure 3(a)) are satisfied. In the colluding collision attack (Figure 4) and identity delegation attack (Figure 5), the attacker completely evades detection by satisfying all the guards (the nodes in region I of Figure 4 and Figure 5).

The countermeasure we propose against these attacks is based on the observation that an adversary evades detection of dropping packets by allowing only a subset of guards to overhear the message being forwarded. Therefore, we expand the set of nodes that can guard a node from only the common neighbors of the node being monitored and its previous-hop node to include all the neighbors. Since all neighbors are included in verifying the node, by definition, some neighbor will see evidence of stealthy packet drop. The detection technique makes use of the fact that, under the stealthy packet dropping attacks, neighbors have differing views of a node in terms of the volume of traffic it has forwarded and all the neighbors cannot be convinced by a single broadcast. To achieve this goal we need to introduce additional tasks for the nodes in the network. (i) Each node, say X, keeps a count of the number of messages each of its neighbors, say Y, had forwarded (FCount(X, Y) or FC(X, Y) for short) over a predetermined time interval and (ii) each node has to announce the number of packets it has forwarded over some period of time. The adversary evades detection of stealthy packet dropping by allowing only a subset of guards to overhear the packet being forwarded. Thus, the subset of guards that had overheard the packet forwarding would have a higher count than the nodes that did not overhear the forwarding. By forcing a node to announce the number of messages it has forwarded over some period of time, a malicious node would have the problem of satisfying two sets of neighbors that expect to hear different counts through a single broadcast.

A neighbor of a node, say *N*, that collects the number of forwarded packets by *N* and compares the result with the count announced by *N* is called a *comparator* of N, denoted by *C(N)*. For any node *N* all nodes in radio range *R(N)* act as comparators of *N*. Recall that a guard of a node *B* over the link *Y*→*B*, has been defined in the BLM as any node that lies within the transmission range of both *Y* and *B*. Therefore, each guard of *N* over a certain link is a comparator of *N*, however, not every comparator of *N* is a guard of *N*. The function of a comparator is to count the total number of packets forwarded from the node within a time period. During some time periods node *N* may be required to announce the number of messages it has forwarded in that period. If a comparator's count is not within an acceptable range of the announced forward count, the comparator increments its *MalC* for the announcing node.

In order to reduce traffic, we do not require all nodes to announce their forward count for every time period. Instead a node must announce within the time period that it receives a broadcast message request to announce. Whenever a node, *A*, overhears a packet from a node *N* that is not within the neighbor list of *A*, node *A* broadcasts a 3-hop request for *N* to announce its forward count. If node *N* and all of its neighbors are within 3 hops of the requestor, then the neighbors of *N* will act as comparators of *N* and expect to hear the correct forward count announced. The basic idea is that a malicious node that has dropped a packet faces a dilemma; some of its neighbors have overheard the dropped packet and expect it to be included in the send count while others have not heard the packet so they expect a send count of one less message. However, note that a suspicion would not be raised by a discrepancy of one due to natural losses (channel conditions and collisions). Detection is triggered only when the discrepancy crosses a predetermined threshold.

For simplicity of exposition, for the following examples, we will consider that a discrepancy of a single packet is sufficient for detection. Consider the power drop attack scenario shown in Figure 3(a), the neighbors of *M* within the dotted circle would have one more count for the number of packets forwarded by *M* as compared to the counters in the rest of *M*'s comparators. In each of the last three attack modes, the attacker is faced by two sets of neighbors that have different views about him. The best the attacker can do is to satisfy the larger set, however, the nodes of the other set would detect the discrepancy and propagate the detection knowledge to the nodes of the other set. All the nodes of the smaller set would then directly isolate the malicious node. The nodes of the larger set indirectly isolate the malicious node if the number of nodes in the smaller set is greater than or equal to γ.

6 Analysis

The analysis gives the detection probability for a malicious node indulging in the drop through misrouting and power control attack types. It also provides the probability of false detection of legitimate nodes. We analyze BLM and Sadec under different network conditions.

Assumptions: We consider a homogeneous network of nodes where the nodes are uniformly distributed in the field with density d. For simplicity, we assume that the field is large enough that edge effects can be neglected. Consider any two randomly selected neighbor nodes, S and M, as shown in Figure 3(b). Nodes S and M are separated by a distance X, and the communication range is r. X is a random variable that has the probability density function of $f_X(x) = 2x/r^2$ with range $(0,r)$. This follows from the assumption of uniform distribution of the nodes.

Attacker model: The malicious node M uses an omni-directional antenna. Its goal is to have the effect of dropping the packet from reaching the legitimate next-hop node T. The detection probability is a lower bound since we assume that the adversary can control the transmission power level to be infinitesimally smaller than that required to reach T. The reduced transmission range of M is represented as y.

Output Parameters: We define (i) the *probability of detection* as the probability that a malicious node is detected by a single guard node, (ii) the *probability of isolation* as the probability that the node is directly detected by at least γ neighbors and therefore isolated, (iii) the *probability of false detection or isolation* as the probability that a non-malicious node is detected by a neighbor or by at least γ neighbors due to natural reasons such as collision or drop in the communication channel, (iv) the *probability of framing detection or isolation* is the probability that a non-malicious node is detected by a neighbor or γ neighbors due to malicious activities.

In the following, we analyze a representative attack from each of the two proposed mechanisms to detect stealthy packet dropping. The first is the misrouting stealthy packet dropping and the second is the power control stealthy packet dropping. We provide the results for basic local monitoring (BLM) and Sadec.

6.1 Misrouting Stealthy Packet Dropping

Consider the scenario in Figure 6 below. A node A is relaying a packet (P_{in}) to the next-hop node M, which is malicious. Node M is supposed to relay the packet to the legitimate next-hop node B as P_{out}. Instead, M relays the packet to a wrong next-hop E as P_{mr}.

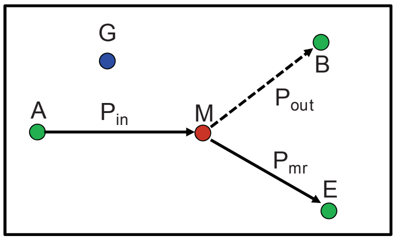

Figure 6: Misrouting stealthy packet drop scenario

There are four different possibilities for the guard G in Figure 6:

1. G misses both P_{in} and P_{mr} → missed detection
2. G misses P_{in} but gets P_{mr} → detection as fabricate (which is incorrect since the malicious action is misrouting)
3. G gets P_{in} but misses P_{mr} → detection as drop (incorrect)
4. G gets both P_{in} and P_{mr} → successful misrouting detection for Sadec and missed detection for BLM.

Assume the event of missing each packet due to natural channel errors is independent and is given by P_c. Assume that A sends ψ packets to be relayed by M within the time window T_{win}. Assume that M selectively misroutes (to evade detection) packets with probability P_{mal}. Then, the number of packet misroutes (μ) that occurs within T_{win} is $[\psi \bullet P_{mal} \bullet (1-P_c)]$. Also assume that the MalC threshold over time window of T_{win} is β and each malicious activity increases the MalC by one. The average number of guards (g) of M over A→M is computed in [6] $g \approx \lfloor 0.59 N_b \rfloor$, Where Nb is the average number of neighbors.

6.1.1 Basic Local Monitoring (BLM)

In BLM, case 1 is considered missed detection and case 4 looks normal and thus represents a successful packet drop that goes undetected. How we classify cases 2 and 3 is subjective. In both cases, the malicious activity is detected, though the activity is classified incorrectly. To obtain an optimistic estimate of the capacity of BLM, in this analysis, we consider cases 2 & 3 as detection for malicious nodes and false detection for good nodes. The probability of cases 2 & 3 is,

$$\begin{aligned} P_{2\&3} &= P_c(1-P_c) + (1-P_c)P_c \\ &= 2 \cdot P_c(1-P_c) \end{aligned} \qquad (1)$$

Using the binomial distribution, the probability of detection of a malicious node at a guard is given by,

$$P_{detect} = \sum_{i=\beta}^{\mu} \binom{\mu}{i} (P_{2\&3})^i (1-P_{2\&3})^{\mu-i} \qquad (2)$$

We consider the case $\mu \geq \beta$, since otherwise, the probability of detection is zero and the rest of the analysis becomes vacuous.

Recall that a node is considered isolated when it is detected by at least γ neighbors when the number of neighbors $\geq \gamma$ (Section 3.2), assuming that the alert propagation process is reliable. If the number of neighbors $< \gamma$, then the node is considered isolated if all the neighbors detect (and isolate) the node. Therefore, the isolation probability is,

$$P_{isolate} = \begin{cases} \sum_{i=\gamma}^{g} \binom{g}{i} (P_{detect})^i (1-P_{detect})^{g-i} & , g \geq \gamma \\ (P_{detect})^g & , g < \gamma \end{cases} \qquad (3)$$

As we discussed in the first paragraph of this section, Equations (2) and (3) also represents the probability of false detection and false isolation.

Additional harm is caused by misrouting due to the possibility of framing. In Figure 6, E ignores the packet P_{mr} since it is not the correct next-hop from M. This will trigger the guards of E to mark it as dropping packets. The probability that a guard of E gets P_{mr} is

$$P_{frame} = (1-P_c) \qquad (4)$$

Therefore, the probability of framing detection is given by,

$$P_{fdetect} = \sum_{i=\beta}^{\mu} \binom{\mu}{i} (P_{frame})^i (1-P_{frame})^{\mu-i} \qquad (5)$$

And the probability of framing isolation is given by,

$$P_{fisolate} = \begin{cases} \sum_{i=\gamma}^{g} \binom{g}{i} \left(P_{fdetect}\right)^{i} \left(1-P_{fdetect}\right)^{g-i} & ,g \geq \gamma \\ \left(P_{fdetect}\right)^{g} & ,g < \gamma \end{cases} \quad (6)$$

Combining equations (3) and (6) above, we get the probability of a legitimate node being isolated.

6.1.2 Sadec

Case 4 represents the probability of correct detection at a guard with SADEC. As in BLM, cases 2&3 are considered detection for malicious nodes and false detection for legitimate nodes. The probability of cases 2, 3, and 4 is given by,

$$P_{2,3\&4} = 1 - P_1 = 1 - P_c^2 \quad (7)$$

The probability of detection is given by,

$$P_{detect} = \sum_{i=\beta}^{\mu} \binom{\mu}{i} \left(P_{2,3\&4}\right)^{i} \left(1-P_{2,3\&4}\right)^{\mu-i} \quad (8)$$

The probability of isolation is given by,

$$P_{isolate} = \begin{cases} \sum_{i=\gamma}^{g} \binom{g}{i} \left(P_{detect}\right)^{i} \left(1-P_{detect}\right)^{g-i} & ,g \geq \gamma \\ \left(P_{detect}\right)^{g} & ,g < \gamma \end{cases} \quad (9)$$

The probability of false detection and false isolation are given by Equations (2)& (3) respectively. The probability of frame detection and isolation is zero since the guards of E in SADEC are aware that E is not the correct next-hop and therefore take no action when E drops the packet.

The probability of true isolation (of a malicious node) for the misrouting attack for BLM and SADEC is shown in Figure 7. With a high enough density, both can completely isolate the malicious node. However, SADEC achieves this with a lower density—greater than 0.9 probability with 31 neighbors compared to 38 for BLM.

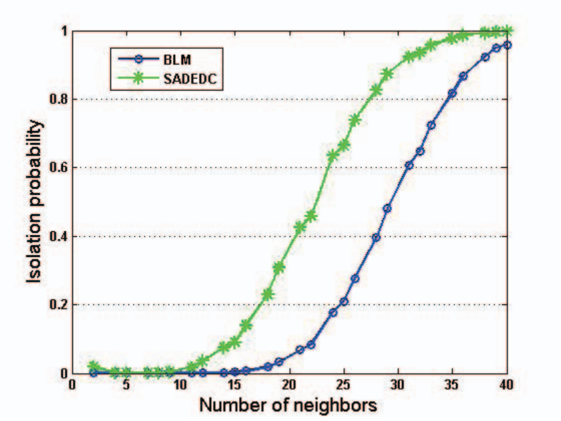

Figure 7: Probability of true isolation for the misrouting with BLM and SADEC. γ=3, P_{mal}=0.7,ψ=10, β=2.5,r=30, P_c=0.01Nb/3.

The probability of isolating a legitimate node, due to natural errors on the wireless channel and due to framing is shown in Figure 8. We see that as the density increases, this probability quickly reaches 1 for BLM. However, in SADEC, since the malicious nodes get identified and isolated, they do not get the chance to frame the legitimate nodes and therefore, the probability of mistakenly isolating legitimate nodes remains low.

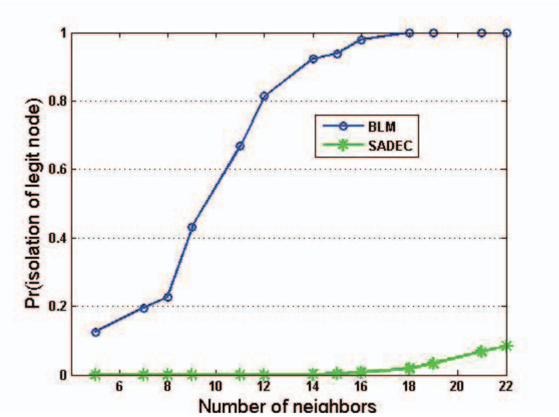

Figure 8: Probability of mistakenly isolating a legitimate node due to natural collisions and framing with the misrouting attack. γ=3, P_{mal}=0.7,ψ=10, β=2.5,r=30, P_c=0.01Nb/3.

6.2 Power Control Stealthy Packet Dropping

6.2.1 Basic Local Monitoring (BLM)

The guards of M over the link from S→M lie on the shaded area shown in Figure 3 (b). The subset of guards that can be satisfied by the controlled power transmission of M lies on the shaded area shown in Figure 3 (c), we call these guards the happy guards g_h. Finally, the subset of guards of T over M→T that wrongly accuse T of dropping the packet are shown in the shaded area of Figure 3 (d), we call these guards the fooled guards, g_f. The shaded area in Figure 3 (c) is found to be,

$$Area(c) = \begin{cases} \begin{pmatrix} y^2 \cos^{-1}\left(x^2+y^2-r^2\right) + r^2 \cos^{-1}\left(x^2+r^2-y^2\right) \\ -\sqrt{\left(r+y-x\right)\left(x+y-r\right)\left(x+r-y\right)\left(x+y+r\right)} \end{pmatrix} \\ \pi y^2 & ,when \ (x+y) \leq r \end{cases}$$

Without loss of generality, we assume that the distance between S and M is the same as the distance between M and T. This makes the shaded area in Figure 3(d) the same as that in Figure 3(c) and we can use Area(c) to represent each of them. Therefore, $g_h = g_f = Area(c) \times d$. Finally, the number of guards that can detect the power control attack is $g_d = g\text{-}g_h$.

The probability of detection at a guard is the same as in Equation(2). The probability of isolation is the same as in Equation (3) after replacing g with g_d. The probability of false detection and false isolation is exactly the same as those in Equations (2) and (3) respectively (without any replacements).

The probability of framing detection is the same as that in Equation(5). The probability of framing isolation is the same as that in Equation(6) after replacing g with g_f.

6.2.2 SADEC

The expected number of comparators of any node is $N_c = \pi r^2 d$ the subset of comparators that can overhear M are those that lie within the dotted circle of Figure 3(c), we call these comparators the *Plus Comparators* C_p. The subset of comparators that cannot overhear the transmission of M are those that lie within the legitimate transmission range of M but out of the dotted circle, we call these the *Minus Comparators* C_m.

$$C_p = \pi y^2 d \text{ and } C_m = N_c - C_p = \pi(r^2 - y^2)d$$

In SADEC, for detection, a comparator node, say node A which is a comparator of node B, matches its count of the number of packets forwarded by B with the count announced by node B. If the count differs by more than FC_{th}, then node *A* will detect node *B*. Assume that μ is greater than FC_{th} since otherwise, the probability of detection is zero. A malicious node has one of two choices to announce for its forward counter value: either (i) zero to match with the counter values of the members of C_m or (ii) μ to match with the counter values of the members of C_p. Due to channel problems, a member of C_p could be in one of three states: (i) it overhears at least μ+1-FC_{th} packets. In this case, the forward counter of that member will match with the value μ that may be announced by the malicious node. (ii) It overhears less than FC_{th} packets. In this case, its forward counter will match with the value zero that may be announced by the malicious node. (iii) It overhears less than μ+1-FC_{th} and more than or equal to FC_{th}. In this case, its forward counter value will not match with any of the two possible announced values of the malicious node. The probability that a member of the C_p group overhears at least μ+1-FC_{th} packets (Case i) is given by,

$$P_{FC_{th}} = \sum_{i=\mu+1-FC_{th}}^{\mu} \left\{ \binom{\mu}{i} (1-Pc)^{i} \, P_c^{\mu-i} \right\} \qquad (9)$$

The probability that a member of the C_p group overhears less than FC_{th} packets (Case ii) is given by,

$$P_{\prec FC_{th}} = \sum_{i=0}^{FC_{th}-1} \left\{ \binom{FC_{th}-1}{i} (1-P_c)^{i} \, P_c^{FC_{th}-1-i} \right\} \qquad (10)$$

The probability that a member of the C_p group overhears less than μ+1-FC_{th} and more than or equal to FC_{th} (Case iii) is given by,

$$P_{\prec FC_{th}\prec} = 1 - P_{FC_{th}} - P_{\prec FC_{th}} \qquad (11)$$

The actual number of plus comparators (C_{pa}) is given by,

$$C_{pa} = C_p \cdot \left(P_{FC_{th}} + P_{\prec FC_{th}\prec} \right) = C_p \cdot \left(1 - P_{\prec FC_{th}} \right) \qquad (12)$$

The actual number of minus comparators (C_{ma}) is given by,

$$C_{ma} = C_m + C_p \cdot P_{\prec FC_{th}\prec} + C_p \cdot P_{\prec FC_{th}} \qquad (13)$$

A malicious node can successfully launch power control attack while avoiding isolation if min(C_{pa}, C_{ma}) < γ, since the intelligent adversary broadcasts a message count that satisfies the larger of the two sets. Therefore, the probability of isolation is given by,

$$P_{isolate} = Prob(\min(C_{ma}, C_{pa}) \geq \gamma) \qquad (13)$$

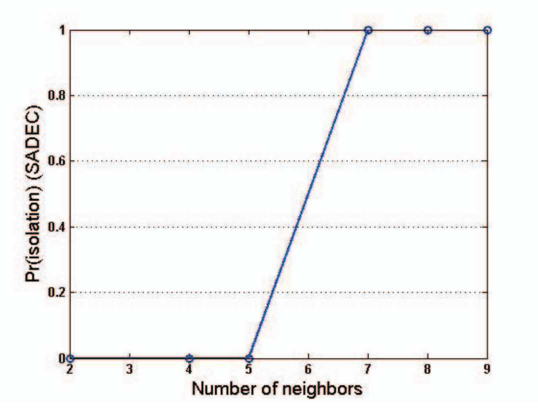

Figure 9: Prob. isolation of a malicious node through SADEC for the power control attack. γ=3, P_{mal}=0.7,ψ=10, β=2,r=30,P_c=0.01Nb/3,FC_{th}=3.5.

The probability of correctly isolating a malicious node involved in the transmission power control attack using SADEC is shown in Figure 9. We see that the isolation is deterministic under the given assumptions – if the number of comparators in the smaller class (between C_{ma} and C_{pa}) exceedsγ, then isolation always happens; otherwise it never happens.

A node, say *X*, may be falsely detected by its neighbor, say *Y*, if |FC(X,X) – FC(Y,X)| >= FC_{th}. This occurs when *Y* misses FC_{th} or more packets forwarded by *X*. Therefore, the probability of false detection is given by,

$$P_{fls\_detect} = \sum_{i=FC_{th}}^{\mu} \left\{ \binom{\mu}{i} P_c^{i} \left(1-Pc\right)^{\mu-i} \right\} \qquad (13)$$

And the probability of false isolation is given by,

$$P_{fls\_isolate} = \begin{cases} \displaystyle\sum_{i=\gamma}^{Nb} \binom{Nb}{i} \left(P_{fls\_detect}\right)^{i} \left(1-P_{fls\_detect}\right)^{Nb-i}, Nb \geq \gamma \\ \left(P_{fls\_detect}\right)^{Nb} \qquad ,Nb < \gamma \end{cases} \qquad (13)$$

The probability of framing detection and isolation is zero.

The probability of mistakenly isolating a legitimate node in SADEC is shown in Figure 10. This arises purely due to natural errors on the wireless channel. Expectedly as the density increases, the natural errors increases, leading to a higher likelihood for a legitimate node to be isolated. However, it remains low even with 40 neighbors (< 0.17).

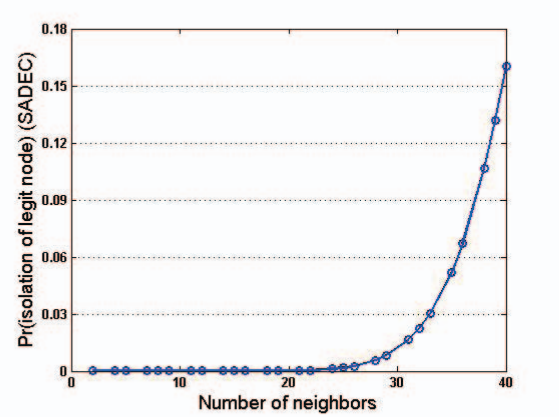

Figure 10: Probability of mistakenly isolating a legitimate node in SADEC with the power control attack due to natural collisions. γ=3, P_{mal}=0.7,ψ=10,□=2,r=30,P_c=0.01Nb/3,FC_{th}=3.5.

7 Overhead Analysis and Testbed Results

The memory cost of a technique like SADEC may be of concern since overheard packets has to be maintained in memory. However, the common case behavior is that nodes behave legitimately. Therefore, the packets are forwarded quickly and do not have to be kept in memory for long. Energy overhead of monitoring involves: (i) the energy spent by the CPU running the specific details of the monitoring algorithm such as searching the buffers, reading and writing in the serial flash, (ii) the energy spent in sending/receiving packets related to monitoring such as neighbor discovery and malicious node detection announcements, and (iii) the energy spent in idle listening. The last ingredient depends on whether the network is implementing sleeping and on which sleeping technique is being used. For the detailed mathematical analysis of energy overhead in both cases (with and without sleeping), we refer the reader to our work in [6] and the energy conserving addition in [24][6].

The additional resource requirements of SADEC over BLM include (a) state maintenance, which includes the next-hop information of every active route within the transmission range of the guard and the forward counters that are maintained by each node for each neighbor and for the node itself. Each node requires N_b+1 forward counters. The first grows linearly with the number of routes while the latter grows linearly with the number of neighbors (N_b), (b) broadcast of the forward counters in an on-demand basis, triggered by a relatively rare event (when a node hears from another node that is not in its neighbor list), or periodically every T_{win} time units, and (c) two node identifiers in each *REQ* and *REP*.

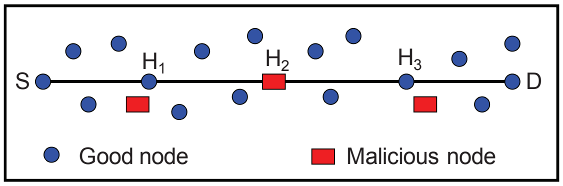

Figure 11: Test-bed setup

To evaluate and compare the energy overhead of SADEC and BLM, we implemented the two schemes separately on a test-bed consisting of Crossbow Mica2 motes [32]. The test-bed (Figure 11) includes one sender node (*S*) and one receiver node (*D*) separated by 3 nodes (i.e., 4 hops). Each node in the route has 8 neighbors. The distance between the nodes in the route is 30 meters. We use indirect measurements of energy consumption, namely, the time the CPU spends in the algorithm, the number of flash memory writes, and the time a node needs to be on just for monitoring purposes, which includes both the receive and the transmit time. Then we calculate the energy consumption using these measured parameters and the Mica2 data sheet values for the current draw [32]: CPU active = 8 mA, idle 3.3mA, sleep 8µA, Serial flash write 15mA, serial flash read = 4 mA, serial flash sleep 2µA, Radio Rx 10mA, Tx (max power) = 27mA. The watch buffer and the forward counters are maintained in the serial flash (512 Kbytes). The experiments are conducted on 50 nodes with N_M = 3, γ = 3, T_{win} = 0.2 seconds, *and* ψ *varies from 2 to 10 packets in* T_{win}. We use for each node two Alkaline Long-life AA batteries (1.225 average voltages, each provide a total of 9360 joules). The experiment time is 60 minutes.

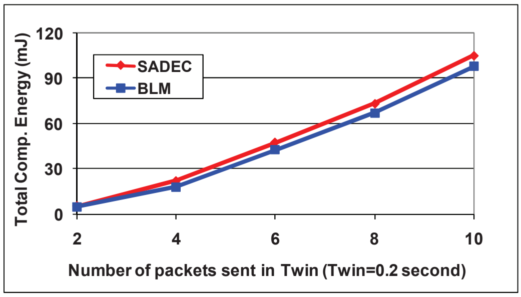

Figure 12: Average total computational energy consumed per node

Figure 12 shows the results of the experiment conducted to measure the computational energy overhead (items (i) and (a) above). The figure shows the average total energy cost per node during the experiment. For this experiment, we implement the algorithm for storing packets in the watch buffer and searching in it through a linear search. The algorithm takes the size of the watch buffer as input. For the experiment, the maximum size of the watch buffer over all the guard nodes is used. The algorithm is executed to search for a random number between 0 and 0.2 million. Since the size of the watch buffer is much smaller, most of the searches are unsuccessful mimicking a guard node overseeing a malicious node which is dropping packets (in the case of misrouting attack). Since unsuccessful searches take longer than successful ones, this results in an overestimate of the execution time. Since a lower packet rate results in smaller watch buffer sizes and fewer number of searches as well as fewer number of accesses to the forward counters, the overhead at the highest packet rate is about 18 times the overhead at the lowest packet rate. Most importantly, note that the difference in the computational energy overhead between SADEC and BLM is small since incrementing the forward counters is a light operation.

Figure 13 shows the results of the experiment conducted to measure the total monitoring energy overhead, i.e., the sum of the factors (i), (ii), (iii), (a), (b), and (c) mentioned earlier in this sub-section under the two attacks. Expectedly, the figure shows that the total energy overhead increases as the packet rate increases. Note that the worst case total energy overhead (when the number of packets is 10) over a 1-hour period is less than 1.5 Joule which represents only 0.008 % of the total energy that the AA batteries can provide. This provides strong evidence that SADEC is parsimonious in its energy overhead and is, therefore, suitable for energy-constrained sensor networks. Moreover, note that the computational energy overhead (Figure 12), when *number of packets sent in* T_{win} = 10, is less than 7% of the total energy overhead. Finally, note that SADEC overhead energy is slightly higher than that of BLM due to the additional forward counter announcements.

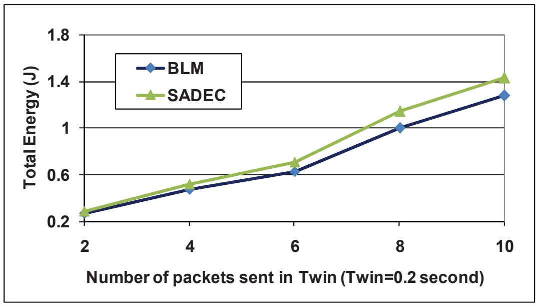

Figure 13: The average total overhead energy per-node due to monitoring for both BLM and SADEC over the experiment time (1 hour)

8 SIMULATION RESULTS

We use the *ns-2* simulation environment [20] to simulate a data exchange protocol, individually with BLM and with SADEC. We distribute the nodes randomly over a square field (1500m×1500m) with a fixed average node density. We use a generic on-demand shortest path routing protocol that floods route requests and unicasts route replies in the reverse direction. A route, once established, is not used forever but is evicted from the cache after an idle period $TOut_{Route}$ if no other packet has been forwarded to the particular destination. We simulate the misrouting attack and the power control attack as a representative of the second class of stealthy packet dropping attacks. A malicious node does not generate any data of its own. The simulation also accounts for losses due to natural collisions. The guards inform all the neighbors of the detected malicious node through multiple unicasts. For each simulation run; malicious nodes are chosen at random.

Input parameters: Each node acts as a data source and generates data using an exponential random variable with inter-arrival rate ϕ. The destination is chosen at random and used for a random time following an exponential distribution with rate ξ. We use N_M for the number of malicious nodes and N for the total number of nodes. The input parameters with the experimental values are given in Table 2, we use the same settings as in [6] so that the results are comparable.

Output parameters: The output parameters include (i) the fraction of data packets received (delivery ratio) calculated as the total number of packets successfully received by final destinations over the total number of packets sent, (ii) the framing isolation ratio, which is defined as the fraction of good nodes that have been incorrectly isolated due to the attack over the total number of good nodes, (iii) the false isolation ratio, which is defined as the fraction of good nodes that have been isolated due to natural causes (collisions and losses on the wireless channel) over the total number of good nodes, (iv) the isolation ratio, which is defined as the number of malicious nodes isolated to the total number of malicious nodes,(v) average isolation time which is the time taken between starting the first attack incidence by a malicious node to the time of isolation of that node averaged over all the malicious nodes (vi) the average end-to-end delay of data packets, which is the time a packet takes after leaving the source until it reaches its final destination. Note that here we only consider framing as a result of the attacks being simulated and we do not consider the kind of framing where enough number of malicious nodes in a neighborhood frames a legitimate neighbor. The latter kind of framing is identical to that in BLM and has already been analyzed [6].

Table 2: Input parameters for SADEC simulation

Param.	**Value**	**Param.**	**Value**
Tx Range (r)	250 m	ξ ,	1/280
MalC increment	50	f_{mal}	1.0
$\text{TOut}_{\text{Route}}$, T_{win}	50 s, 100	N_M	0-20
C_t, γ	150, 3	τ	0.5 sec
# nodes (N)	100	BW	40 kbps
FC_{th}	3	ϕ	1/70

The output parameters are measured at the end of the simulation time (2000 seconds). The output parameters are obtained by averaging over 30 runs. The reasoning provided for some experimental results was arrived at by careful examination of the simulation logs. When a claim is made of difference between SADEC and the BLM, the difference is significant at the 95% confidence level. We are interested in the different output parameters shown above. However, some of the plots show identical trends in the two attacks and therefore, we choose to show them for only one of the attacks. Additionally, in the interest of space, we omit some plots whose results appeared obvious. These include: (I) for the misrouting attack – (1) isolation time vs. N_M, (2) delivery ratio vs. γ, and (3) probability of false isolation vs. γ. (II) For the power control attack – probability of isolation versus. γ.

8.1 Misrouting Attack

This attack is implemented by letting any malicious node that is involved in a routing path to relay incoming packets to an incorrect next-hop with a probability of f_{mal}.

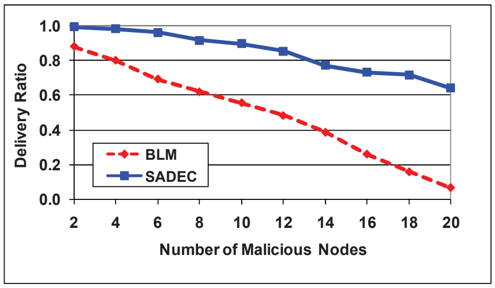

Figure 14: Effect of N_M on delivery ratio, γ=3

Figure 14 shows that the delivery ratio decreases as N_M increases. This is due to the packets dropped before the malicious nodes are isolated. As N_M increases, this initial drop increases and thus the delivery ratio decreases. Moreover, as N_M increases, the true isolation decreases. Therefore, the malicious nodes that could not be detected continue to drop packets which decreases the delivery ratio. The delivery ratio in BLM is much less than in SADEC and the difference increases as N_M increases. This is due to two main reasons. The first is that BLM fails to detect any of the malicious nodes and thus they continue to drop packets constantly. The second is that some of the good nodes in BLM get framed by the adversary and thus become isolated which reduces the overall throughput.

Figure 15 shows the variations in isolation probability as N_M varies. We observe that BLM has a poor performance while SADEC achieves over 80% isolation ratio with up to 12% compromised nodes. The figure also shows that the true isolation decreases as we increase N_M. This is because the number of available guards and comparators in the network decreases as more and more nodes get compromised. Furthermore, as N_M increases, local isolation becomes less effective since the number of legitimate neighbors decreases and if this goes below γ, then local isolation has to wait for direct isolation individually by each legitimate neighbor. Moreover, as N_M increases the data traffic in the network decreases (in the simulation malicious nodes do not send data) which results in a decrease in the number of packets that a malicious node drops. This in turn results in decreasing the likelihood that the malicious node is detected.

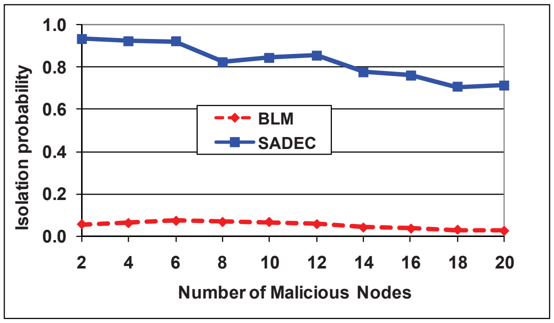

Figure 15: Effect of N_M on isolation probability, γ=3

Figure 16 shows the variations in false isolation as N_M varies. The figure shows that the false isolation probability in SADEC is slightly higher than that in BLM. This is due to the fact that the traffic load in SADEC is higher than that in BLM. SADEC detects and isolates more malicious nods than BLM, which reduces the number of dropped packets in SADEC as compared to BLM. Moreover, framing (Figure 18) in BLM is higher than that in SADEC, which further reduces the traffic in BLM. The figure shows that false isolation decreases as we increase N_M. As N_M increases the traffic load decreases due to two reasons. First, malicious nodes do not generate traffic by simulation setup. Second, the number of packet drop increases as N_M increases. Moreover, the number of good nodes decreases as we increase N_M. This in turn results in a decrease of the indirect false isolation since a node may not have more than γ legitimate nodes to agree on falsely isolating a neighbor.

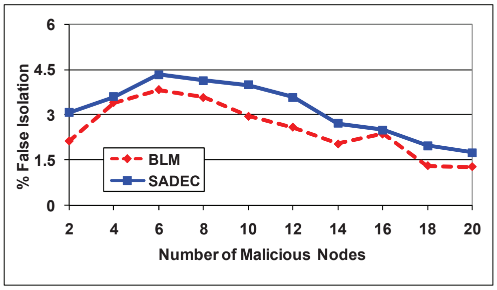

Figure 16: Effect of N_M on false isolation, γ=3

Figure 17 shows the variations in end-to-end delay as N_M varies (the average number of hops between source-destination pairs is 5.6 and the standard deviation is 1.9). The figure shows that the end-to-end delay initially increases as N_M increases and then starts to decrease. As N_M increases, the route reestablishment frequency increases. This is due to the fact that a route remains active for a time $\text{TOut}_{\text{Route}}$ and this timer is reset with every packet forwarded using that route. Consequently cutting the flow of packets (by maliciously dropping the packet) causes the route entries to stale. Therefore, additional traffic is generated to reestablish the route which increases traffic load. The opposing pull comes from the fact that as N_M increases the traffic decreases. As N_M increases beyond a point, the latter factor dominates and the overall result is a decrease in the end-to-end delay. The end-to-end delay in SADEC is slightly higher than that in BLM. This is due to the fact that the traffic load in SADEC is higher than that in BLM. SADEC detects and isolates more malicious nods than BLM, which reduces the number of dropped packets in SADEC as compared to BLM.

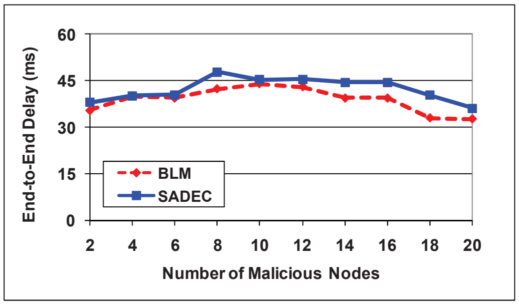

Figure 17: Effect of N_M on end-to-end delay, γ=3

Figure 18 shows the variations in framing ratio as N_M varies. The figure shows that the framing ratio increases as N_M increases for both BLM and SADEC. However, the framing ratio in BLM is much higher than in SADEC. The framing in BLM occurs as a consequence of successful and continuous misrouting attack (BLM fails to detect and isolate the malicious nodes). As N_M increases, the framing ratio increases due to the increase in the number of attack occurrences. As N_M increases, the framing ratio starts to level off since the traffic in the network becomes low and no more good nodes can be framed. On the other hand, the little framing in SADEC is due to imperfect true isolation of malicious nodes due to collisions, channel conditions, or insufficient number of guards (from Figure 15, we see that the coverage is not 100%). As N_M increases, the fraction of malicious nodes not isolated increases and thus framing increases.

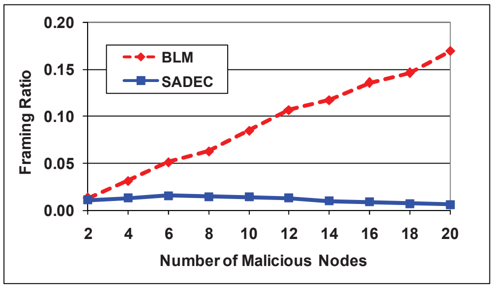

Figure 18: Effect of N_M on framing, γ=3

8.2 Power Control Attack

This attack is implemented by letting any malicious node that is involved in a routing path to control its power transmission to a distance less than that between the malicious node and the next-hop. We are considering a sophisticated adversary that has perfect control over its transmission power. Therefore, the results shown here are pessimistic for both BLM and SADEC.

7.2.1 Effect of N_M

Figure 19 shows the variations in delivery ratio as the number of malicious nodes varies. The figure shows that the delivery ratio decreases as N_M increases for the same reason as in Figure 14. The delivery ratio in BLM is less than that in SADEC and the difference increases as the number of malicious nodes increases. This is due to two main reasons. The first is that BLM fails to detect more malicious nodes than SADEC and thus they continue to drop packets constantly. The second is that more good nodes in BLM get falsely isolated which reduces the overall throughput.

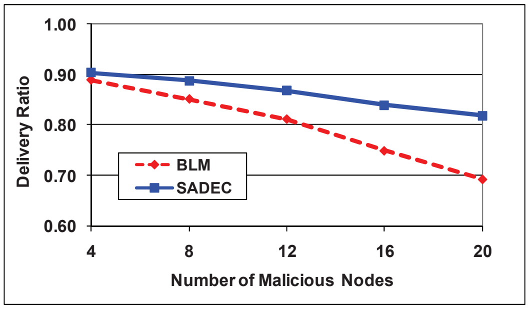

Figure 19: Effect of N_M on delivery ratio, γ=3

Figure 20 shows the variations in isolation probability as N_M varies. Most significantly, we observe that SADEC has significantly better performance than BLM. This is due to the availability of more comparators in SADEC compared to the number of guards in BLM. Moreover, the figure shows that the isolation probability decreases as we increase N_M for the same reasoning as in Figure 15.

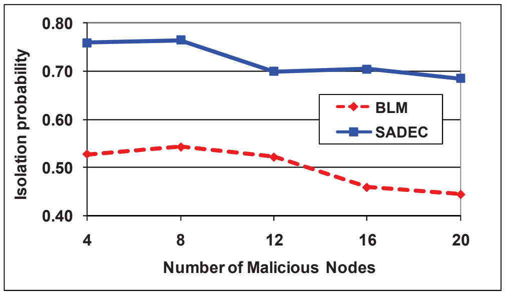

Figure 20: Effect of N_M on isolation probability, γ=3

Figure 21 shows the variations in framing ratio as N_M varies. The figure shows that framing is almost zero in SADEC. In BLM, framing is much larger and increases as we increase N_M, which can be explained as in Figure 18. Moreover, as N_M continues to increase, the false isolation probability starts to level off since the traffic in the network becomes low and no more good node can be framed. As N_M increases the traffic becomes low due to two factors (i) malicious nodes do not generate traffic in our simulation model and (ii) the isolation of many of the malicious nodes and good nodes creates some unreachable nodes.

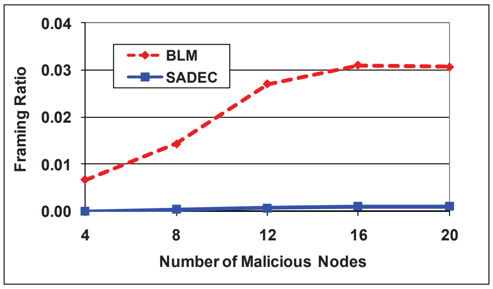

Figure 21: Effect of N_M on framing ratio, γ=3

Figure 22 shows the variations in average isolation time as N_M varies. It shows that SADEC has lower average isolation time. This is due to the ability of SADEC to isolate malicious nodes earlier than BLM due to the availability of more comparators. The figure also shows that the average isolation time increases as the number of malicious nodes increases in both SADEC and BLM. This is due to the reduction in the number of available comparators and guards respectively as the number of malicious nodes increases.

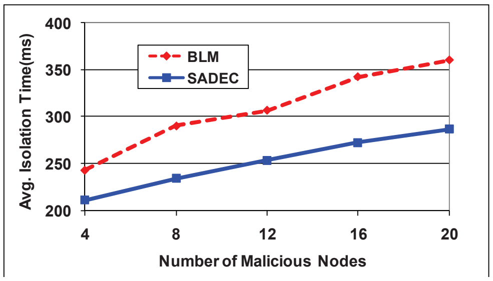

Figure 22: Effect of N_M on average isolation time, γ=3

7.2.2 Effect of γ

Figure 23 shows the variations in delivery ratio as γ varies. The figure shows that SADEC performs better than BLM and the advantage increases as γ increases. This is due to availability of more comparators in SADEC, which enables it to continue to isolate malicious nodes even with higher values of γ. In BLM the delivery ratio decreases sharply as γ increases since enough guard nodes are not available in many neighborhoods to successfully isolate the malicious nodes.

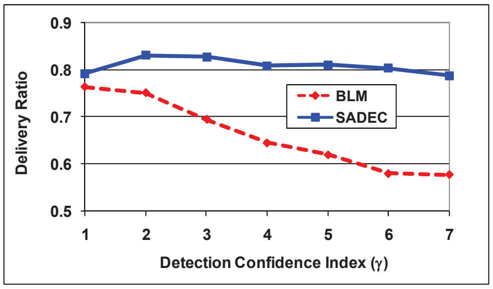

Figure 23: Effect of γ on delivery ratio, N_M=20

Figure 24 shows the variations in isolation probability as *γ* varies. The figure shows that SADEC always performs better than BLM due the availability of larger number of comparators in SADEC compared to the number of guards in BLM. Also the figure intuitively shows that the isolation probability decreases as the value of *γ* increases since *indirect detection* [6] becomes harder as *γ* increases. Indirect detection occurs in a node that could not directly oversee the malicious activities of a malicious neighbor but rather builds its decision based on alerts from a threshold number (*γ*) of other neighbors to the malicious node.

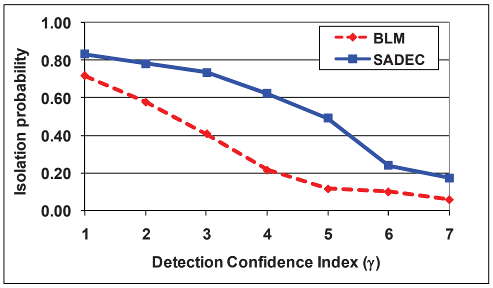

Figure 24: Effect of γ on isolation probability, N_M=20

Figure 25 shows the variations on framing ratio as *γ* varies. The figure shows that the framing ratio drops sharply as the value of *γ* increases. As *γ* increases, it becomes more and more unlikely to have more than *γ* nodes in a neighborhood that accuse a good node. Importantly, the figure shows that the framing ratio in SADEC decreases as *γ* increases much faster compared to BLM. This is due to the better detection capabilities in SADEC that enables it to detect more malicious nodes faster than BLM.

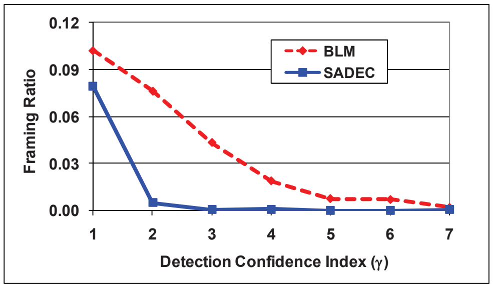

Figure 25: Effect of γ on framing ratio, N_M=20

Figure 26 shows the variations in average isolation time as γ varies. The figure shows that isolation time generally increases as γ increases since it becomes harder to isolate as γ increases as explained in Figure 24. Moreover, note that the isolation time goes down as γ increases further and it will go to zero for higher γ values. This is due to the decrease in isolation probability which affects the average isolation time. When the isolation probability goes to zero, the number of isolated nodes is zero and therefore average isolation time is zero. Finally and most importantly, note that SADEC has much lower isolation time compare to BLM due to the efficiency of the detection in SADEC.

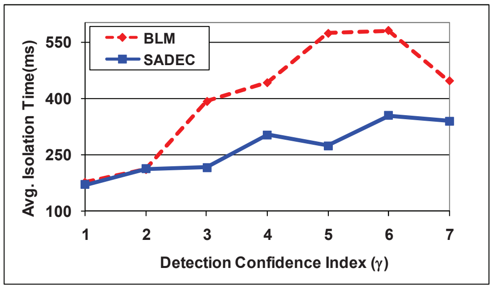

Figure 26: Effect of γ on average isolation time, N_M=20

The three most take away results that can be collected from the results are: (i) SADEC has much better malicious node isolation performance than BLM. For example, in Figure 15, the isolation probability of SADEC is 8 times more than that of BLM with up to 15% malicious nodes. (ii) SADEC has much lower framing probability than that of BLM. For example, in Figure 18, framing in BLM with 20 malicious nodes is almost 20 times more than that in SADEC. Framing is very dangerous since it takes legitimate nodes off the network. (iii) The delivery ration in SADEC is much better than that in BLM. For example, in Figure 14, the delivery ration in SADEC with 20 malicious nodes is almost 6 times more than that in BLM.

9 DISCUSSION

Here we have described the design of SADEC, which fundamentally relies on the ability of some guard nodes to overhear the behavior of neighboring nodes. This basic feature of wireless networks has been leveraged by many researchers, for almost a decade now starting from [28]. Any technique that relies on this has the shortcoming that it can be bypassed by a powerful adversary that can accurately place malicious nodes or precisely control their transmission power. Intrinsically, the placement or the transmission power control can be used to hide the behavior from the requisite number of guard nodes or comparator nodes, e.g., the next hop node does not get the packet but the guards see it. In that case, no detection will occur. SADEC suffers from this shortcoming as does *all* the work that relies on the feature. However, it is less susceptible to this than prior work since it increases the number of nodes that are performing the verification.

10 CONCLUSION

We have introduced a new class of attacks called *stealthy packet dropping* which disrupts a packet from reaching the destination by malicious behavior at an intermediate node. This can be achieved through misrouting, controlling transmission power, malicious jamming at an opportune time, or identity sharing among malicious nodes. However, the malicious behavior *cannot be detected by any behavior-based detection scheme presented to date*. Specifically, we showed that basic local monitoring (BLM) based detection cannot detect these attacks. Additionally, it will cause a legitimate node to be accused. We then presented a protocol called SADEC that successfully mitigates all the presented attack. SADEC builds on local monitoring and requires nodes to maintain additional routing path information and adds some checking responsibility to each neighbor. Additionally, SADEC's new detection approach expands the set of neighbors that are capable of monitoring in a neighborhood, thereby making it more suitable than BLM in sparse networks. We showed through analysis and simulation that BLM fails to mitigate most of the presented attacks while SADEC successfully mitigates them. The improvement is seen in terms of increase in the probability of isolation of malicious nodes and decrease in the probability of isolation of legitimate nodes.

In future work, we are considering detection techniques for multi-channel multi-radio wireless networks. The listening activity for detecting malicious behavior is more complicated due to the presence of multiple channels and multiple radios. We also plan to analyze the impact of the detection technique on the network throughput under different adversary models.

11 REFERENCES

[1] A. A. Pirzada and C. McDonald, "Establishing Trust In Pure Ad-hoc Networks," in ACSC 04, 26(1), pp. 47-54, 2004.

[2] S. Buchegger, J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes-Fairness in Distributed Ad-hoc NeTworks," in MOBIHOC'02, pp. 80-91, 2002.

[3] Y. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks," in Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), pp. 135-147, 2003.

[4] Y. C. Hu, A. Perrig, and D. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols," ACM Workshop on Wireless Security (WiSe'03), pp. 30-40, 2003.

[5] Y. C. Hu, A. Perrig, and D.B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in INFOCOM, pp. 1976-986, 2003.

[6] I. Khalil, S. Bagchi, and N. Shroff, "LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks," in DSN'05, pp. 612-621, 2005.

[7] I. Khalil, S. Bagchi, and N. B. Shroff, "MOBIWORP: Mitigation of the Wormhole Attack in Mobile Multihop Wireless Networks," Elsevier Ad hoc Networks Journal, V. 6, Issue 3, pp. 344-362, May 2008.

[8] I. Khalil, S. Bagchi, C. Nina-Rotaru, and N. Shroff, "UNMASK: Utilizing Neighbor Monitoring for Attack Mitigation in Multihop Wireless Sensor Networks," in Elsevier Ad Hoc Networks Journal, V. 8, I. 2, pp. 148-164, 2010.

[9] S.J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," in IEEE International Conference on Communications (ICC), pp. 3201-3205, 2001.

[10] Q. Zhang, P. Wang, D. Reeves, and P. Ning, "Defending against Sybil Attacks in Sensor Networks," SDCS 2005, pp. 185-191, 2005.

[11] C. Basile, Z. Kalbarczyk, and R. K. Iyer, Neutralization of Errors and Attacks in Wireless Ad Hoc Networks, DSN'05, pp. 518-527, 2005.

[12] B. Carbunar, I. Ioannidis and C. Nita-Rotaru, JANUS: Towards Robust and Malicious Resilient Routing in Hybrid Wireless Networks, WiSe'04, pp. 11-20, 2004.

[13] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru and H. Rubens, "ODSBR: An On-Demand Secure Byzantine Resilient Routing Protocol for Wireless Ad Hoc Networks," ACM Transactions on Information and System Security (TISSEC), V. 10, Issue. 4, 2008.

[14] R. Molva, G. Tsudik, and D. Westhoff (Eds.), "Statistical Wormhole Detection in Sensor Networks," ESAS'05, LNCS 3813, pp. 128–141, 2005.

[15] D. Liu and P. Ning, "Establishing Pair-wise Keys in Distributed Sensor Networks," in the ACM Conference on Computer and Communications Security (CCS), pp. 52-61, 2003.

[16] D. Johnson, D. Maltz, and J. Broch, The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, Ad Hoc Networking, Addison-Wesley, 2001.

[17] C. E. Perkins and E. M. Royer, Ad-Hoc On-Demand Distance Vector Routing, in Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pp. 90-100, 1999.

[18] D. Ganesan, B. Krishnamurthy, A.Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical Report IntelIRP-TR-02-003, Intel Research, March 2002.

[19] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Computer Communication Rev.'94, pp. 234–44, 1994.

[20] "The Network Simulator- ns-2," www.isi.edu/nsnam/ns/

[21] S. Bagchi, S. Hariharan, N. B. Shroff, "Secure Neighbor Discovery in Wireless Sensor Networks," Purdue University TR ECE 07-19. At "http://docs.lib.purdue.edu/ecetr/360/".

[22] R. Muraleedharan and L. A. Osadciw, "Jamming attack detection and countermeasures in wireless sensor network using ant system" in Wireless Sensing and Processing, proceedings of the SPIE, V. 6248, pp.62480G, 2006.

[23] S. Buchegger and J. L. Boudec, "Robust reputation system for p2p and mobile ad-hoc networks," in Economics of Peer-to-Peer Systems, 2004.

[24] I. Khalil, S. Bagchi, and N. Shroff, "SLAM: Sleep-Wake Aware Local Monitoring in Sensor Networks," in the 37th IEEE Dependable Systems and Networks Conference (DSN'07), p.p. 565-574, June 2007.

[25] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in ACM Workshop on Wireless Security (WiSe), pp. 1-10, 2003.

[26] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole attacks," in NDSS, pp. 131-141, 2004.

[27] S. Hariharan, N. Shroff, and S. Bagchi, "Secure Neighbor Discovery in Wireless Sensor Networks," Purdue Technical Report, TR ECE 07-19, 2007.

[28] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proceedings of the 6th Annual International Conference on Mobile Computing and Networks, pp. 255-265, 2000.

[29] R. de Oliveira and T. Braun, "A Dynamic Adaptive Acknowledgment Strategy for TCP over Multihop Wireless Networks," in the IEEE Computer Communication Conference (INFOCOM'05), pp. 1863-1874, 2005.

[30] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing Exposed Terminals in Wireless Networks," in USENIX Symposium on Networked Systems Design & Implementation (NSDI '08), pp. 59-72, 2008.

[31] I. Khalil and S. Bagchi, "MISPAR: Mitigating Stealthy Packet Dropping in Locally-Monitored Multi-hop Wireless Ad Hoc Networks," in ACM SecureComm'08 , doi: http://doi.acm.org/10.1145/1460877.1460913, 2008.

[32] http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf

[33] I. Stojmenovic, "Handbook of Sensor Networks: Algorithms and Architecture", 2005.

[34] F. Ye, H.Luo, J.Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor network," in the Proceedings of 8th ACM Annual Conference on Mobile Computing and Networking, pp. 148-159, 2002.

[35] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments, in MobiSys '06, pp. 28-41, 2006.

[36] C. Hartung, J. Balasalle, R. Han, "Node Compromise in Sensor Networks: The Need for Secure Systems", Technical Report CU-CS-990-05, Department of Computer Science, University of Colorado, January 2005.

[37] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in ACM Trans. Sen. Netw. 4, 3 (May. 2008), pp. 1-37. DOI= http://doi.acm.org/10.1145/1362542.1362546.

[38] G. Shafer, A Mathematical Theory of Evidence: Princeton University Press, 1976.

[39] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs" in IEEE Transaction on Mobile Computing, Vol. 6, no. 5, pp. 536-550, 2007.

Issa Khalil received the B.Sc. and the M.S. degrees from Jordan University of Science and Technology in 1994 and 1996 and the PhD degree from Purdue University, USA in 2006, all in Computer Engineering. Immediately thereafter he worked as a post-doctoral researcher at the Dependable Computing Systems Lab of Purdue University until he joined the College of Information Technology (CIT) of the United Arab Emirates University (UAEU) as an assistant professor in 2007. Khalil's research interests span the areas of wireless and wireline communication networks. He is especially interested in security, routing, and performance of wireless Sensor, Ad Hoc and Mesh networks. His current research is funded by grants from National Research Foundation, Emirates Foundation, and United Arab Emirates University. Dr. Khalil is the technical program co-chair of the 6th International Conference on Innovations in Information Technology and was appointed as a Technical Program Committee member and reviewer for many international conferences and journals. He is a member of IEEE since 2006.

Saurabh Bagchi joined the School of Electrical and Computer Engineering at Purdue University in West Lafayette, Indiana as an Assistant Professor in August 2002, where he is now an Associate Professor. Before that, he did his Ph.D. from the Computer Science department of the University of Illinois at Urbana-Champaign with Prof. Ravishankar Iyer at the Coordinated Science Laboratory. His Ph.D. dissertation was on error detection protocols in distributed systems and was implemented in a fault-tolerant middleware system called Chameleon. His research on wireless sensor networks is being supported by the National Science Foundation CISE Directorate and the Indiana 21st Century Research and Technology Fund.