Enhancing Base Station Security in Wireless Sensor Networks

Jing Deng, Richard Han, and Shivakant Mishra

jing.deng@colorado.edu, {rhan,mishras}@cs.colorado.edu

Technical Report CU-CS-951-03

April 2003

University of Colorado

Department of Computer Science

Enhancing Base Station Security in Wireless Sensor Networks

Jing Deng, Richard Han, Shivakant Mishra

Department of Computer Science

University of Colorado, Boulder, CO 80309-0430.

Contact: {rhan, mishras}@cs.colorado.edu

Abstract

Wireless sensor networks that are deployed in applications such as battlefield monitoring and home sentry systems face acute security concerns, including eavesdropping, forgery of sensor data, denial of service attacks, and the physical compromise of sensor nodes. Sensor networks are often organized hierarchically, with a base station serving as a gateway for collecting data from a multi-hop network of resource-constrained sensor nodes. Prior work that has focused on securing the routing between sensor nodes has assumed that the base station is sufficiently powerful to defend itself against security threats. This paper considers strategies for securing the sensor network against a variety of threats that can lead to the failure of the base station, which represents a central point of failure. First, multipath routing to multiple destination base stations is analyzed as a strategy to provide tolerance against individual base station attacks and/or compromise. Second, confusion of address and identification fields in packet headers via hashing functions is explored as a technique to help disguise the location of the base station from eavesdroppers. Third, relocation of the base station in the network topology is studied as a means of enhancing resiliency and mitigating the scope of damage.

1 INTRODUCTION

Wireless sensor networks are rapidly growing in popularity. Applications of sensor networks that have emerged include habitat monitoring [1], robotic toys [2], and battlefield monitoring [3]. A wide range of applications are emerging, including location aware sensor networks in the home and office, assistive technology for biomedical sensing, and outdoor deployments of sensor networks to monitor storms, oceans, and weather events. For military deployments, security is essential to protect the routing infrastructure and packet data from threats such as eavesdropping, tampering, denial-of-service (DOS) attacks, and the physical compromise of sensor nodes deployed into enemy territory.

The research challenge is to secure the routing infrastructure against such threats given the severe resource constraints imposed by wireless sensor networks. Wireless sensor networks consist of individual sensor nodes that are highly resource-constrained in terms of their limited energy lifetime, modest CPU, and scant memory [2, 8]. While it has been demonstrated that symmetric key cryptography can be implemented on today's wireless sensor platforms [5,12], initial results indicate that public key cryptography remains out of reach for today's sensor networks due the compute-intensive nature of public key methods [12]. Prior work in securing wireless sensor networks therefore focuses on exploiting symmetric key-based techniques for achieving authentication, data integrity, and confidentiality. As a result, a key focus of this paper concerns security obtained through symmetric key cryptography.

A notable feature of the architecture of a wireless sensor network is its hierarchy, rooted in a base station. As shown in Figure 1, a wireless sensor network often collects and relays data to a back-end server via a gateway or base station. The base station is typically resource-rich in terms of its computational ability, storage capacity, and energy lifetime compared to individual sensor nodes. A base station will have

capabilities on the order of a laptop or laptop-equivalent and will be capable of both wired connectivity to the Internet as well as wireless connectivity to the sensor network. In some cases, the base station may be mobile, situated on top of a roving van or command vehicle, or may have limited mobility enough to be guided to an opportune location in the sensor network topology. A fundamental assumption of this paper is that the sensor network architecture conforms to the base station-rooted hierarchy shown in Figure 1. Prior work in securing sensor networks given a base station-rooted topology includes the SPINS suite of security building blocks [5], fault tolerant routing [9], securing of TinyOS routing as well as directed diffusion [10], and the INSENS secure routing system [12]. Prior work in securing ad hoc networks given peer-to-peer routing includes SEAD [6] and Ariadne [7], which both utilize symmetric key schemes, as well as a number of public key techniques that are too costly for today's sensor networks [15, 16, 17, 18].

Given a base station architecture and symmetric key cryptography, this paper considers strategies for securing the sensor network against a variety of threats that can lead to the failure of the base station, which represents a central point of failure. Prior work that has focused on securing the routing between sensor nodes has assumed that the base station is sufficiently powerful to defend itself against security threats. In contrast, this paper considers that the base station itself may be vulnerable. As a result strategies must be implemented throughout the sensor network to withstand attacks that can lead directly or indirectly to the failure of the base station.

As shown in Figure 1, we consider three strategies for securing the sensor network against base station failure. First, as shown in Figure 1a), multipath routing to *multiple* destination base stations is analyzed as a strategy to provide tolerance against individual base station attacks. This strategy is considered both for the route discovery phase as well as the data routing phase. We also analyze the extent to which the number of base stations enhances the resilience of the network. Second, Figure 1b) illustrates confusion of address and identification fields in packet headers via hashing functions. This approach is designed to disguise the location of the base station and thereby counter threats from a passive observer who would eavesdrop on packet headers, especially the source, destination and type fields, in order to infer and trace back the location of the base station. Third, Figure 1c) depicts the relocation of the base station in the network topology. We analyze the extent to which base station mobility and placement can affect the resiliency of the network and mitigate the scope of the damage inflicted by a malicious sensor node. The strategies studied in this paper are limited to the particular kinds of threat models outlined above. Our objective is not to claim that these strategies withstand all manner of attacks, e.g. wormhole [23] attacks, or apply to all sensor networks, e.g. mobile sensor networks in which all sensor nodes move, not just the base station.

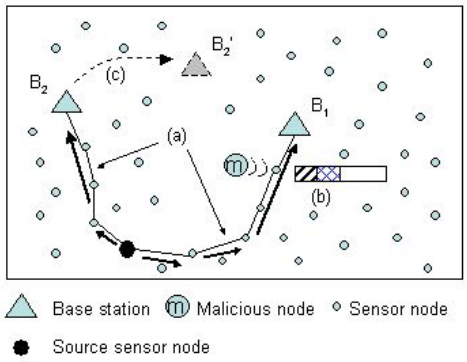

Figure 1. a) Multiple paths to multiple base stations b) disguising header address fields c) relocation of base station.

In Sections 2 and 3, the strategy of multipath routing to *multiple* destination base stations has been considered. Section 2 describes a route discovery protocol in a wireless sensor network in the presence of multiple base stations. Route discovery protocol ascertains the topology of a wireless sensor network after the sensor nodes are deployed. This section describes the design of this protocol, analyzes its resilience against different type of security attacks, and presents performance measurements from a simulated prototype to illustrate the power of multiple base stations during route discovery. Section 3 describes a secure and intrusion-tolerant data routing protocol that exploits multiple redundant routes to different base stations. This section illustrates the resilience of a wireless sensor network comprising of multiple base stations against sensor node compromises and base station failures via performance measurements from a simulated prototype. Protocols described in these two sections are based on INSENS secure routing mechanism [12].

In Section 4, the strategy of confusion of address and identification fields in packet headers via hashing functions has been considered. This section details how the location of the base station is disguised via confusion of identification fields as well as relocation of the base station. In Section 5, the strategy of relocating base stations has been considered. Different base station placement strategies, so as to improve the resilience of the sensor network against attacks on base stations and sensor nodes. Section 6 discusses the related work, Section 7 provides a discussion and future research directions, and finally, Section 8 concludes the paper.

To test the performance of the three strategies, we have simulated wireless sensor networks in ns2[19] simulator. We use the following parameters in most of the experiments described in this paper. For a random network topology, we generate 200 nodes, and put them in a 2500 X 2500 m^2 square area. For grid network topology, we generate 14 X 14 nodes, and put them in a 2860 X 2860 m^2 square area. For each experiment, we randomly generate 30 to 50 network topologies. The results shown in various graphs in the paper are average values of each test.

2 MULTIPLE BASE STATIONS: ROUTE DISCOVERY

A route discovery protocol ascertains the topology of the sensor network. Our route discovery protocol is based on INSENS route discovery protocol [12]. INSENS provides support for intrusion-tolerant routing in wireless sensor network. It builds multiple redundant paths between sensor nodes and a base station to bypass intermediate compromised nodes. In addition, INSENS limits DOS-style flooding attacks, prevents false advertisement of routing and other control information, and is designed for *resource-constrained* wireless sensor network. In particular, INSENS ensures that a single compromised node can only disrupt a localized portion in the network, and cannot bring down the entire sensor network.

While INSENS has several important useful features, it suffers for a serious drawback. It assumes that the base station cannot fail or be isolated from the network by malicious compromised nodes. This assumption may not hold under several scenarios. For example, if an adversary discovers the location of a base station, it can isolate it from the rest of the network by simply jamming the communication medium in its neighborhood. In this paper, we overcome this drawback by accommodating multiple base stations that cooperate with one another to build a robust wireless sensor network. In particular, we consider a redundant base stations model of wireless sensor network, and design protocols to build redundant routing mechanisms in such a network. These protocols preserve all the good features of INSENS, and in addition provide support for tolerating failure of one or more base stations.

In particular, our route discovery protocol adheres to the following design principles. First, to prevent DOS-style flooding attacks, individual nodes are not allowed to broadcast to the entire network. Only the base stations are allowed to broadcast. Base stations act as gateways to the wired world, e.g. a satellite uplink connecting to terrestrial networks. Authentication of the base stations is achieved via one-way hashes, so that individual nodes cannot spoof the base station and thereby flood the network. Unicast packets must first traverse through a base station, thereby preventing DOS/DDOS broadcast attacks. Second, to prevent advertisement of false routing data, control routing information must be authenticated. A key consequence of this approach is that the base stations always receive knowledge of the topology that is correct, though it may only represent a partial picture due to malicious packet dropping. Third, to address resource constraints, 1) symmetric key cryptography is chosen for confidentiality and authentication between the base stations and each resource-constrained sensor nodes, since it is considerably less compute-intensive than public key cryptography, and 2) the resource-rich base station is chosen as the central point for computation and dissemination of the routing tables. Fourth, to address the notion of compromised nodes, redundant multipath routing is built to achieve secure routing. The goal is to have disjoint paths, preferably to different base stations so that even if an intruder takes down a single node or path, secondary paths will exist to forward the packet to the correct destination.

Route discovery is subdivided into two rounds. In the first round, the base stations flood (limited flooding) a *request message* to all the reachable sensor nodes in the network. In the second round, each sensor node sends its neighborhood topology information back to two different base stations using a *feedback message*.

2.1 Route Discovery: Route Request

Whenever there is a need to construct the forwarding tables of all sensor nodes, a central node directs all base stations to initiate the first round of the route discovery protocol. Each base station initiates a *request message* by broadcasting it to all its neighbors. When a sensor node receives a request message initiated by base station *b* for the first time, it forwards (broadcasts) this request message. This request message includes a path from the base station *b* to *x*. As this request message is forwarded downstream in the network, each node appends its identity in the path. On receiving a request message, a node *x* also records the identity of the sender of this message in its neighbor set. A node may receive a request message initiated by base station *b* many times. However, it forwards this request message at most once. When a node receives another request message initiated by *b*, the identity of the sender is added to its neighbor set, but the request is not rebroadcast. This implies that if there are *n* base stations, a node may forward up to *n* request messages, each initiated by a different base station. To further limit the scope of flooding of request messages, we include a protocol parameter (an integer) that dictates the maximum number of request messages a node may forward. For example, Figure 2(a) shows the forwarding of request messages when the value of this parameter is 3.

A malicious node in the network can attempt to launch several attacks in this round. First, it can attempt to spoof the base station by sending a spurious request message. Second, it can include a fake path in the request message it forwards. Third, it may not forward a request message, or launch a DOS attack by repeatedly sending several request messages. We adopt the security mechanisms of INSENS to counter these attacks. They require sensor nodes to be pre-configured with appropriate values.

First, each base station *b* generates a sequence of numbers $K_{b0}, K_{b1}, \ldots, K_{bn}$ such that $K_{bj+1} = F(K_{bj})$, where F is a one-way function, $0 < j < n$, and K_{b0} is chosen randomly. All nodes are pre-configured with function F, and final sequence values $K_{an}, K_{bn}, \ldots, K_{mn}$ of each base station, *a, b, ..., m* respectively. A base station *b* transmits K_{bn-1} in the first request message it initiates. Each sensor node can authenticate that this message originated from base station *b* by verifying $K_{bn-1} = F(K_{bn})$. In general, a base station *b* uses K_{bn-i} in the i^{th} route discovery phase. As shown in [12], this mechanism allows a sensor node to authenticate that a request message it received indeed originated from a legitimate base station. This mechanism ensures that a malicious node cannot spoof a base station, and cannot launch DOS attacks by replaying earlier (legitimate) request messages. However, it remains possible that a malicious node could flood a modified request message using the *current* sequence number from a valid request message just sent out by the base station. In such an attack, called a rushing attack [14], an attacker tries to propagate a spurious message before the base station can propagate its own valid message. This attack is confined to the local subtree of nodes below the malicious node. Damages inflicted due to this attack are further reduced by deploying multiple base stations. A node that receives a spurious request message first is still likely to get a valid request message initiated by some other base station. As we will see later, this will enable this node to eventually communicate with at least one base station.

The second mechanism that we use to defend against intrusions is a keyed MAC algorithm. Each sensor node is configured with a separate secret key that is shared only with the base station. This keyed MAC is used to preserve the integrity of control information included in a request message. The overall effect of these security mechanisms is that a malicious node can attack in the first round only by localized flooding, by not forwarding a request message, and by sending fake path in the request which is later on detected in the second round. The latter two attacks will result in some of the nodes downstream from the malicious node not getting a request message or not being able to forward their feedback message to the base station in the second round. Again, a malicious node may be able to compromise a small number of nodes in its vicinity by employing these types of attacks, but cannot jeopardize the security of the complete network.

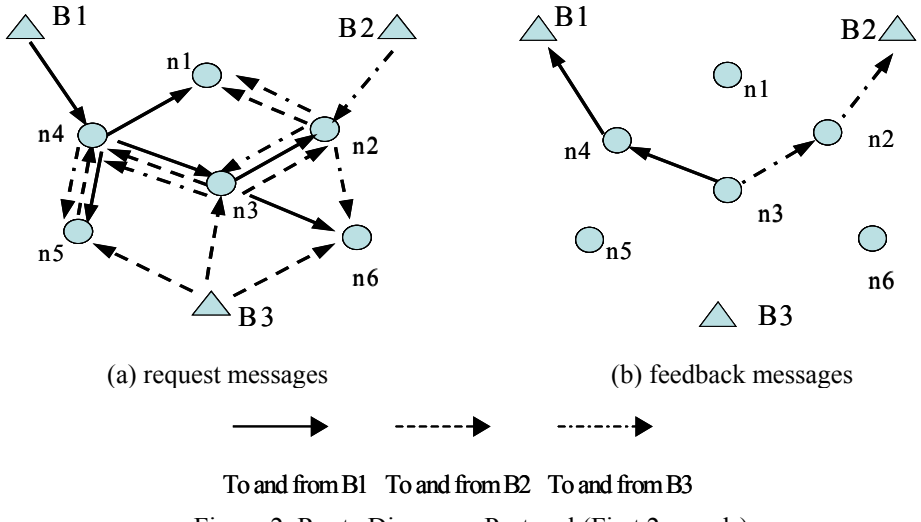

Figure 2. Route Discovery Protocol (First 2 rounds)

2.2 Route Discovery: Route Feedback

In the second round, each sensor node sends its local connectivity information (a set of identities of its neighbor nodes as well as the path to itself from a base station *b*) back to the base station *b* using a *feedback* message. A separate feedback message is sent to every base station whose request message was forwarded in the first round. The mechanism used to send feedback messages to different base stations is same. So, for simplicity, we will concentrate on sending a feedback message to just one base station in the following discussion. After a node has forwarded its request message in round one, it waits a certain timeout interval before generating a feedback message. This interval allows a node to listen to the local broadcasts of its neighbors, who will also be forwarding the same request message. A node will hear the request messages from its upstream, peer and downstream neighbors. A feedback message containing neighbor list and path to *b* is propagated to *b* using the reverse path taken by the request message initiated by *b*.

The integrity of the topology data returned to a base station by each node in its feedback message is protected by a keyed MAC applied over neighbor list, path, and some other control information. This MAC ensures that a base station will construct a correct topology, though it may be incomplete due to malicious nodes that may drop or tamper with feedback messages. The messages that reach the base station are guaranteed after verification to be correct and secure from tampering.

A malicious intruder could still launch several attacks. First, an intruder could launch a DOS-style attack and send multiple feedback messages to each of its upstream neighbors. Second, an intruder could eavesdrop and learn topology information. To address the first DOS-style attack, we employ two defense mechanisms: (1) To prevent repetitive transmissions of a feedback packet from the same originating node, all nodes follow the policy of not forwarding duplicate feedback messages; and (2) use rate control to prevent transmissions of feedback packets from many thousands of phantom originating nodes. To provide confidentiality against eavesdropping by a malicious node, the path and neighbor information is encrypted using the originating node *x*'s secret key, with the caveat that the identity field of the originating node in the path is left unencrypted.

The overall effect of these security mechanisms is that a malicious node is limited in the damage it can inflict, whether attacking by DOS attack, by not forwarding a feedback message or by modifying the neighborhood information of nodes, which can be detected at the base station. The rate-controlled DOS attack will affect upstream nodes, but only in a limited way. The latter two attacks will result in some of the nodes downstream from the malicious node not being able to provide their correct connectivity information to the base station. Though a malicious node could launch a battery-drain attack by persistently sending spurious feedback messages at the rate-controlled limit, such an attack would still affect a limited

number of upstream nodes. In summary, a malicious node may be able compromise only a small number of nodes in its vicinity using these attacks.

2.3 Performance Evaluation of Route Discovery

As mentioned earlier, a malicious node may be able to compromise a small set of nodes in its vicinity during route discovery. We have performed a set of experiments to measure the extent of damage a malicious node can cause during route discovery. We have simulated two types of attacks a malicious node may launch. In the *passive attack*, a malicious node either drops feedback messages or modifies the neighbor information in the feedback message before forwarding (recall that this tampering is later on detected by a base station). The effect of passive attack is that some of the nodes may not be able to convey their connectivity information to the base station and hence will not be included in the network topology constructed by the base station.

In the *active attack*, a malicious node launches a DOS attack during the second round of route discovery protocol. Figure 3 shows the result of launching active and passive attacks. The x-axis in this graph records the maximum number of nodes that may be compromised by a single malicious node, and the y-axis records the number of such (malicious) nodes. For example, about 34% of the sensor nodes can disable only 5% of the nodes in the network by launching a DOS attack when three base stations are deployed.

The numbers reported in this figure are averaged over 100 different randomly generated topologies of 100 nodes distributed over a 2000 X 2000 m^2 space. In case of active attack, we have calculated this damage by counting all the nodes downstream from the malicious node, its neighbors, and the neighbors' downstream nodes that do cannot reach any base station. In case of passive attack, we have calculated this damage by counting all the nodes downstream from the malicious node that cannot reach any base station.

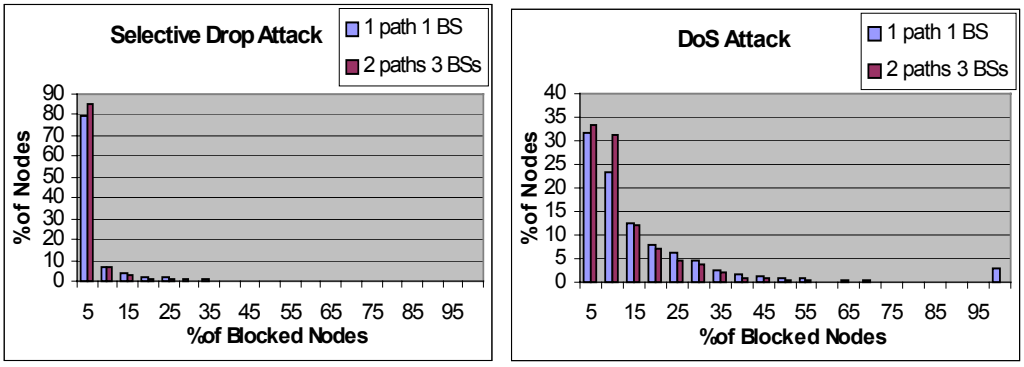

Figure 3 Passive and active attacks during route discovery.

We make three observations from this figure. First, as expected, an active attack compromises more nodes than the passive attack. Second, multiple base stations improve resiliency of the protocol from both passive and active attacks during route discovery. The main reason for this is that a single malicious node can successfully block a set of nodes from a base station if there is only one base station. However, if there are multiple base stations, it becomes extremely difficult for this malicious node to block nodes from all base stations. Finally, there is a catastrophic scenario when there is a single base station. There exists a set of nodes that can bring down the entire network. The reason for this is that the nodes in the vicinity of the lone base station can isolate it from the network by simply launching a DOS attack. Presence of multiple base stations eliminates this catastrophic scenario.

3 MULTIPLE BASE STATIONS: MULTI-PATH DATA ROUTING

A common technique to tolerate failures and security compromises of intermediate nodes in a computer network is to build multiple redundant routing paths between source and destination nodes. These paths are independent of one another in the sense that they share as few common nodes/links as possible; ideally, only source and destination nodes are shared among different redundant paths. Each message sent from a source to a destination is sent multiple times, once along each redundant path. The presence of a few failed

or compromised nodes along some of these paths can jeopardize the delivery of some of the copies of a message. However, as long as there is at least one path that does not contain a failed or compromised node, the destination is guaranteed to receive at least one copy of the message that has not been tampered with. An important advantage of this technique is that it does not require any need for detecting failures or intrusions, i.e. it works despite the presence of (undetected) intrusions. We exploit this technique to build multiple redundant paths in a wireless sensor network.

After sending request messages in the first round, base stations wait for a fixed period of time to collect all the connectivity information received via feedback messages. A very important consequence of the security mechanisms used in the first two rounds is that the base stations can glean out all the connectivity information that has not been tampered with. After computing the correct connectivity information from the feedback messages they have received, base stations share their connectivity information with one another to compute the global topology of the network. From this global topology, multiple redundant routes for each sensor node are computed, and forwarding tables are built. Finally, these forwarding tables are disseminated to the respective sensor nodes.

An important goal in constructing multiple redundant routes is to minimize the damage a malicious node may inflict. In particular, a malicious node has a greater chance of inflicting damage on nearby nodes, for example by launching a DOS attack. So, we choose two independent paths in such a way that the nodes in the two paths are far apart. For a sensor node A, this is done as follows. The first path from A to the closest base station is chosen using the breadth-first search shortest path algorithm. To determine the second path, three sets of nodes, S_1, S_2, and S_3 are first constructed. S_1 is the set of nodes belonging to the first path, S_2 is the set of nodes belonging to S_1 and any neighbor nodes of the nodes in S_1, and S_3 is the set of nodes belonging S_2 and any neighbor nodes of the nodes in S_2. All three sets exclude A or the base station. The second path is then computed as follows.

1. Remove all nodes in S_3 from the network, and find the shortest path from A to a base station. If such a path is found, terminate the computation. The path found it is the second path.
2. Remove all nodes in S_2 from the original network. Find the shortest path from A to a base station. If such a path is found, terminate the computation. The path found it is the second path.
3. Remove all nodes in S_1 from the original network. Find the shortest path from A to a base station. If such a path is found, it is the second path. Otherwise, there is no second path from A to the base station.

An interesting question is which base station should be chosen for the second path? There are at least two different strategies possible here. In the first strategy, the second path is chosen based on the method described above, irrespective of which base station it leads to. In particular, the second redundant path may lead to the same base station as the first path. In the second strategy, the second path is chosen based on the method described above, but with an additional constraint that this path must lead to a *different* base station than the base station in the first path. If such a second path cannot be found, then we revert back to the first strategy. Thus, if the second strategy is used, some sensor nodes may have redundant paths leading to different bases stations, while others may have redundant paths leading to the same base station. Finally, depending on the network topology, it is indeed possible that no second redundant path is found for some sensor nodes. In that case, the current implementation maintains only a single path.

After computing the redundant paths and forwarding tables for each node, respective base stations propagate these tables to the respective nodes in a breadth-first manner. A base station first sends the forwarding tables of all nodes that are its immediate neighbors. It then sends the forwarding tables of nodes that are at a distance of two hops from it, and so on. This mechanism cleverly uses the redundant routing mechanism just built to distribute the forwarding tables. Standard security techniques such as those proposed in [5] can be used to distribute these forwarding tables in a secure manner.

3.1 Performance Evaluation of Multiple Paths

Passive Attack by compromised sensor nodes

With two independent routes available between every node and one of the base stations, our protocol's goal is to route messages correctly in the presence of a single malicious node. Interestingly, our protocol deals

quite well with multiple malicious nodes as well. We have performed a set of experiments to measure the number of nodes that can be blocked when a set of nodes turn malicious and (simply) drop data packets. This can be termed as a *passive attack* as it does not involve any attempt to actively interfere with the routing mechanism or functioning of other nodes. Figure 4 shows the average number of nodes that can be blocked as a function of the number of malicious nodes.

These results are based on a network of 200 nodes randomly distributed over a 1500 X 1500 m^2 space. The numbers reported in this figure are averaged over 50 different combinations of nodes randomly selected to be malicious. For example, for 10 malicious nodes, we measured the number of blocked nodes for 50 different combinations selected randomly of 10 nodes turning malicious. For each test, 20 random topologies were chosen.

This graph shows four different computing scenarios: (1) a single base station with a single path, (2) a single base station with two redundant paths, (3) three base stations with a single path, and (4) three base stations with two redundant paths (computed using the second strategy). There are three observations we make from this graph. First, multiple redundant paths with multiple base stations clearly provides the highest resilience from passive attacks. For example, even when the number of malicious nodes is as high as 10, the number of blocked nodes is only about 15. Under the same conditions, the number of blocked nodes exceeds 40 in other computing scenarios. The second observation is that multiple base stations with single paths is more resilient than single base station with multiple paths. The main reason for this result is that when there is a single base station, the set of nodes in the vicinity of the base station can block a significantly large number of nodes by being able to bring down both the redundant paths. This is less likely when there are multiple base stations. In other words, multiple base stations reduce the number and severity of resilience bottlenecks in the network. Finally, we observe that as the number of failed nodes increase, the effect of maintaining multiple redundant paths diminishes. The number of blocked nodes under single path and two paths scenarios get closer to each other. The main reason for this is that as the number of failed nodes increase, there are fewer correct sensor nodes remaining to be blocked. Thus, the difference between single path and 2-path diminishes.

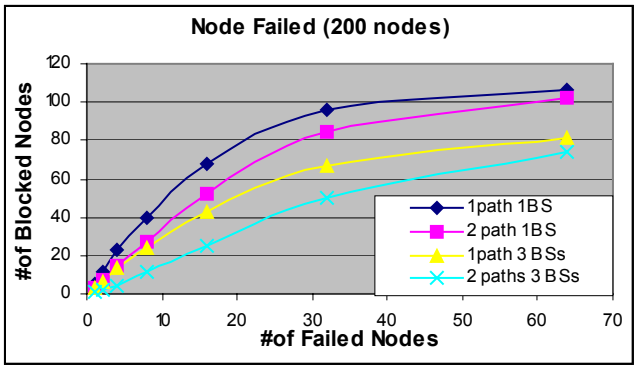

Figure 4 Multi-node passive attack on a sensor network

Active attack by compromised sensor nodes

To analyze the effect of an *active* attack, we have performed a set of experiments to analyze the effect of DOS attacks that a malicious node may launch. The DOS attack we have simulated in these experiments is comprised of repeatedly sending data packets to the base stations to block the wireless medium and not allow other nodes to send their data packets. DOS attacks are difficult to address completely at the network level. In our opinion, these attacks must be addressed at multiple levels. In our analysis, we have assumed that sensor nodes use an appropriate rate-based control mechanism while forwarding data packets. This implies that a malicious node that repeatedly sends data packets will be able to block its neighbors, but not other (upstream) nodes. However, a node in the vicinity of a base station can isolate that base station from the rest of the network by simply launching this DOS attack.

Figure 5 shows the damage a malicious node may cause by launching this type of DOS attack. In this experiment, two topologies (random and grid) are tested. In random generated topology, the position of

each node is randomly selected, while in the grid topology, each node is placed on a square grid. To accommodate the simulator, it was necessary to perturb each position in grid topology to a small region around each vertex in a square grid graph. In this way, random topologies could be generated even for a nearly uniform square grid. The total number of nodes is 200. In the case of single base station, the base station was placed in the center, while in the case of three base stations, the base stations were placed in the middle forming an equilateral triangle.

This figure shows five different computing scenarios: (1) a single base station with a single path, (2) a single base station with two redundant paths, (3) three base stations with a single path, (4) three base stations with two redundant paths (computed using the first strategy), and (5) three base stations with two redundant paths (computed using the second strategy). The x-axis records the percentage of nodes that may be blocked by a DOS attack launched by a single malicious node, and y-axis records the percentage of such (malicious) nodes in the network.

There are several observations we make from this graph. First, the computing scenario involving multiple base stations with multiple redundant paths computed using the second strategy provides the highest resilience against this type of DOS attack. For example, 110 nodes in this computing scenario under random topology can block only 10 nodes by (individually) launching a DOS attack. The second observation is that the multiple redundant paths provide better protection against DOS attacks than the single path approach. The third observation is that the multiples paths approach performs far better than single path for the grid topology. This is because a grid nearly always offers a valid redundant second path. Finally, in the worst case, there are some nodes in the single base station scenarios that can bring down the entire sensor network. These are the nodes that lie in the vicinity of the base station, and if they are compromised, they can launch a DOS attack to isolate the base station. There are no such nodes in the presence of multiple base stations.

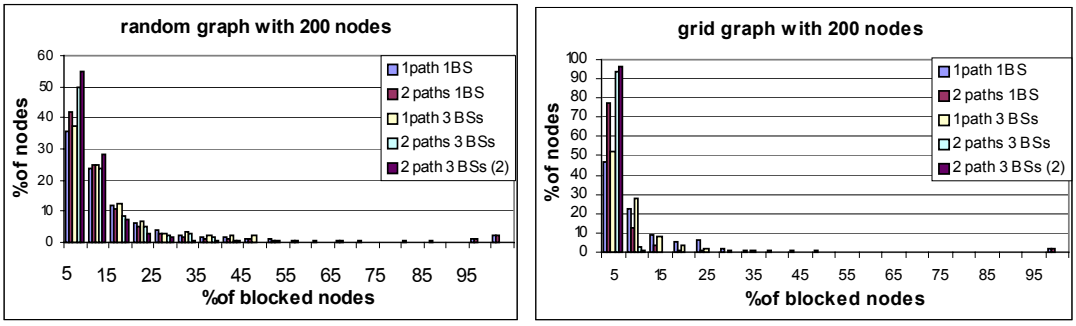

Figure 5 DOS attack on a sensor network

Base station failure

A key advantage of deploying multiple base stations in a wireless sensor network is that the network can tolerate failure of one or more base stations. This is supported by our experiments as shown in Figure 6. This figure shows the number of sensor nodes that cannot reach any base station as a result of one or more base station failures. This figure shows four different computing scenarios, comprising of two, three, four, and five base stations respectively. In all computing scenarios, two redundant paths were computed using the second strategy. There are two observations we make from this figure. First, multiple paths with multiple base stations provide a strong resilience from base station failures. For example, only 19 nodes are disconnected as a result of a base station failure when there are only two base stations in the network. Second, this figure shows that a larger number of base stations provide higher resilience from base station failures. For example, in the presence of five base stations, a single base station failure disconnects only one sensor node, and more than half of the sensor nodes are still connected even when four base stations fail. It is important to note that there is no need to detect base station failures here. Base station failures are automatically tolerated by having redundant base stations.

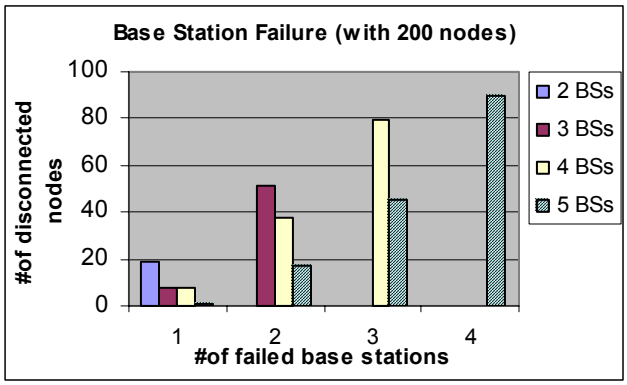

Figure 6 Effect of base station failure

Base Station Scalability

All our experiments have shown that the performance and security of a wireless sensor network are improved by deploying multiple base stations. Furthermore, a larger number of base stations typically results in further performance improvement or increased security. An important question is how many base stations should be deployed. We have performed some experiments to provide an answer to this question. Figure 7 shows the average number of nodes that can be blocked by a single malicious node for different number of base stations in a network. This average has been computed over 20 different random network layouts. We observe that as the number of base stations increase, the average number of blocked nodes decrease. However, the decrease in number of blocked nodes is quite insignificant beyond 4 base stations. This shows that for a random layout network with 100 nodes, four base stations seem to be sufficient.

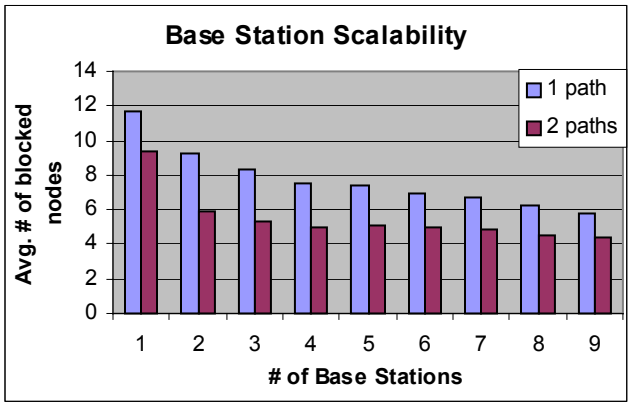

Figure 7 Base Station Scalability

4 DISGUISING BASE STATION LOCATION

Security-critical wireless sensor networks are typically deployed in highly insecure environment. For example, they may be deployed by military in a battlefield to detect enemy troop movements. Since base stations control security and operation of an entire WSN, they are natural targets for compromise by an adversary. Thus, it is important that the identity and location of a base station are not revealed to an adversary. Unfortunately, the nature of communication in a WSN makes it relatively easy for an active adversary to figure out a base station's identity and location. This is because all communication packets either originate from or destined to a base station. We propose two strategies designed to make it difficult for an adversary to locate a base station. These are confusion of IDs, and base station relocation.

4.1 Confusion of ID's

The basic idea behind confusion of IDs is to conceal the source and destination addresses in a packet transmitted over the network. The obvious solution is to have a different pair-wise secret key shared between each node and its neighbor. A node uses an appropriate pair-wise shared key to encrypt this information before sending a packet. Only the intended recipient neighbor node will be able to decrypt this packet. The main problem with this solution is that it is not known in advance who the neighbors of a node might be. As a result, it is extremely difficult to pre-configure nodes with appropriate pair-wise keys. We devise two different solutions, one applicable during the route discovery phase and the other applicable after the route discovery phase.

During the route discovery phase, if an adversary can successfully attack a base station, the base station may not be able to get the complete network topology information. This will not only be a serious hindrance in determining the topology of the network, but also may seriously jeopardize a correct operation of the entire network. We propose an ID confusion technique to make it difficult for an attacker to trace the location of a base station during route discovery. This technique can defend the network from a passive adversary who can only sniff the packets exchanged in a small area of the network, but cannot compromise a sensor node. Despite the limitation of being not able to defend against an active adversary, this technique is practical, because the route discovery phase typically lasts for a very short period of time. An adversary typically won't have enough time to compromise a sensor node during this short time period.

We use a reversible hash function H to do ID confusion, where an ID is the network address of a sensor node or base station. H is not a secure hash function because it is reversible. For each ID m, we compute a set $C_m = \{x : H(x) = m\}$. To do confusion, every node is pre-configured with a global shared key K_c before the network deployment. When a sensor node x sends a packet to destination y, it randomly selects an element x_x from C_x as the source address, and selects an element x_y from C_y as the destination address. The sensor node encrypts the whole packet with K_c, and sends it out. When a node receives this packet, it decrypts it, gets x_x and x_y, computes $H(x_x)$ and $H(x_y)$, and gets node IDs x and y. If node y forwards this packet to node z, it randomly select y_y from C_y, and y_z from C_z as new source and next hop addresses, encrypts packet with K_c, and sends it out. Because a sensor node can use $H^{-1}(x)$ to get an element in C_x, it doesn't need to save C_x. This way, the current and next hop addresses in each request or feedback messages change while being forwarded. Furthermore, since we are using CBC mode for encryption, the (encrypted) contents of the message also changes while being forwarded. A passive attacker cannot know the content of the packet, or the destination or source address of the packet, if he doesn't know K_c. The function H can be very simple, for example, $H(x) = x / 1000$, so it won't add too much computing burden on the sensor node. The main overhead of this technique is the need for a receiver to decrypt the header of a packet to check the source and next hop addresses.

After the route discovery phase, base stations know the topology of the network, and can communicate with all sensor nodes. They create a pair-wise key for each pair of neighbor nodes that are on some route. The pair-wise keys are sent to sensor nodes along with their forwarding table. After receiving a pair-wise key for each neighbor with whom a sensor node needs to communicate, it doesn't need to use K_c to do ID confusion. Instead, it can use the appropriate pair-wise key to simply encrypt the messages it sends. At this stage, even if an adversary can compromise a sensor node, it can only decrypt the message sent to that node. In particular, it won't be able to decrypt messages exchanged between other nodes.

To evaluate the performance of ID confusion technique during route discovery phase, we have implemented the above algorithm in Berkeley MICA sensor motes [8]. The program runs on Atmel Atmega128 microcontroller. Resource constraints are: a 4MHZ processor with 128K Bytes code memory and 4K Bytes internal data memory, an RFM Monolithics TR 1000 radio at 19.2Kbps, two 8-bit Timer/Counters with Separate Prescalers and Compare Modes, and two conventional AA batteries for power. We use nesC [20], and choose RC5 (with 12 rounds) as block cipher. The standard random number generator LFSR is used for hash function reversing. We do ID confusion for a 32 bytes of data with 4 bytes as nodes addresses. Table 1 shows the results of the experiments.

	Speed (milliseconds)	ROM (Bytes)	RAM (Bytes)
RC5 encryption	23.45	520	6
RC5 decryption	23.52	540	104
Hashing	0.89	370	104
Total	47.78	1300	110

Table 1

Thus, it takes an extra 48 milliseconds to forward a packet. If the path from a sensor node to a base station has five intermediate nodes, a packet sent from this sensor node to the base station will take an extra 240 milliseconds. The memory requirements for implementing this algorithm in sensor nodes are also quite low.

4.2 Relocation of Base Stations

The wireless sensor network described so far has been static in nature. The location of sensor nodes and base stations do not change after the initial deployment. However, there are many instances when the sensor nodes themselves may be mobile, e.g. ocean sensor networks that float with water currents to monitor marine life or airborne sensor networks for deployment into storms. In some cases, sensor nodes are attached to the wildlife as the animal moves through its habitat [13]. Further, even though sensor networks are centered around one or more base station architecture, the base stations themselves may be mobile, e.g. a van with an antenna or an armored vehicle command center.

In fact, a very powerful technique to conceal the location of a base station is to simply change its location from time to time. If the location of a base station is frequently changed, it will become extremely difficult for an adversary to figure out the current location of that base station. However, if a base station changes its location, the forwarding tables in some or all sensor nodes will have to be updated.

An important feature of the secure routing framework that we have developed is that it supports relocation of base stations with minimal overhead. Since the base stations know the complete topology of the sensor network after the route discovery phase, they need only determine who their new nearest neighbors are when they relocate to a new location. This is done by a relocated base station flooding a new type of *request message* that the neighbors do not forward. Instead, on receiving this message, they simply send a new type of *feedback message* that contains a MAC and their identity. On receiving the new feedback messages, base stations reconstruct the new topology, compute new routes, and download new forwarding tables as discussed earlier. Notice that there is no need for invoking the route discovery protocol whenever a base station relocates.

Figure 8 shows the total number of packets exchanged when a base station relocates in a single-base station network and a three-base station network. For comparison, we have also included the number of packets exchanged, if the route discovery protocol was used to compute the new routes when a base station relocates. A packet here is defined as a single-hop message, e.g. if a message sent from a base station to a sensor node makes five hops, it is counted as five packets.

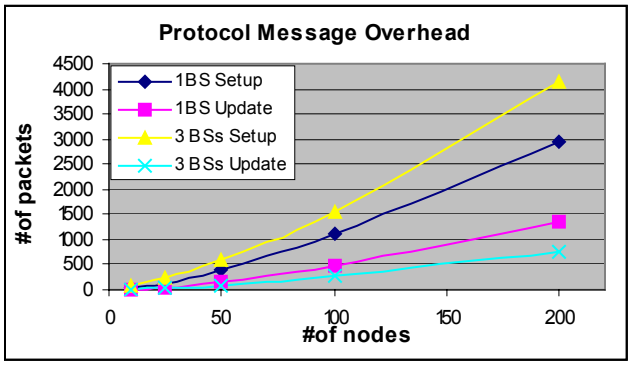

Figure 8 Number of packets exchanged during a base station relocation

We make three important observations from this figure. First, the number of packets exchanged to build the routing framework after a base station is relocated is significantly less that the number of packets exchanged if the route discovery protocol is used. The second observation is that this number is significantly less when there are multiple base stations. The main reason for this is that the number of sensor nodes whose forwarding tables are changed due to a base station relocation is much less if there are multiple base stations. As a result, fewer packets are sent. Finally, it is interesting to note that although the number of packets exchanged during base station relocation is less for multiple base station network than for single base station network, the number of packets exchanged in the initial setup is more for multiple base station network than for single base station network. The reason for larger number of packets exchanged during initial setup in multiple base station network is that multiple request messages (one for each base station) and corresponding feedback messages are propagated in this case.

5 BASE STATION PLACEMENT STRATEGIES

The possibility of relocating base station(s) at a minimal cost opens up a very interesting opportunity in building secure and wireless sensor networks. Since a base station has complete knowledge of the network topology, it may be possible for it to relocate itself to an *optimal* location so as to maximize its security advantage. For example, a base station could be moved from its initial position to a location in the sensor network that provides the densest connectivity, so as to maximize the prospect of multipath routing to deliver its packets. Another security metric could be placing the base station in the center of the network, so as to minimize the maximum exposure to eavesdropping that any given packet would face in traversing the multi-hop network. A third security metric would be to place the base station where it is least susceptible to DOS and DDOS attacks.

Determining optimal locations of a set of base stations is non-trivial and computationally intensive for most cases. We have addressed the issue of optimal placement of base stations to minimize the maximum damage a compromised node can inflict by launching a DOS attack. In particular, we have experimented with several different network layouts (random as well as uniform), and computed optimal locations for two base stations. As an example, Figure 9 shows two such layouts with optimal locations of two base stations. These layouts also show the location of the node that can cause the maximum damage by launching a DOS attack. In the first layout (left), sensor nodes are deployed randomly with the same probability of a sensor node occupying any location. In the second layout (right), a large number of sensor nodes are deployed on the left side (dense part of the network) and only a few nodes or deployed in the remaining area.

Based on these experiments, we have arrived at the following (preliminary) heuristics. For a uniform random deployment of a sensor network, it seems that the optimal location of the two base stations is on the two outer opposite edges of the network. The intuition behind this placement strategy is that an adversary may attempt to cause the maximum damage by compromising a node in the vicinity of a base station, so as to isolate that base station from the rest of the network. By placing the two base stations on the outer opposite edges of the network, the remaining base station can then continue to provide connectivity to most of the sensor nodes. In fact, as the left figure shows, the best place for an adversary to launch attack is towards the middle of the network under this optimal placement. For a dense-sparse network, it seems that the optimal location of one base station is along the edge between the dense and sparse parts of the network, and the optimal location of the other base station is on the opposite outer edge. The intuition behind this placement strategy is that an adversary can naturally cause maximum damage by compromising a node in the middle of the dense part of the network. By placing a base station on an outer edge of the dense part, this base station can reach most of the nodes in the dense part. Note that if the base station is placed in the middle of the dense part, it may become a prime target for attack. Also, by placing the other base station on the opposite sides of the dense part, other nodes that lose connectivity to the first base station due to the attack can stay connected with this base station. Finally, by placing this base station on the edge between dense and sparse parts of the network, all the nodes in the sparse part of the network can stay connected with it.

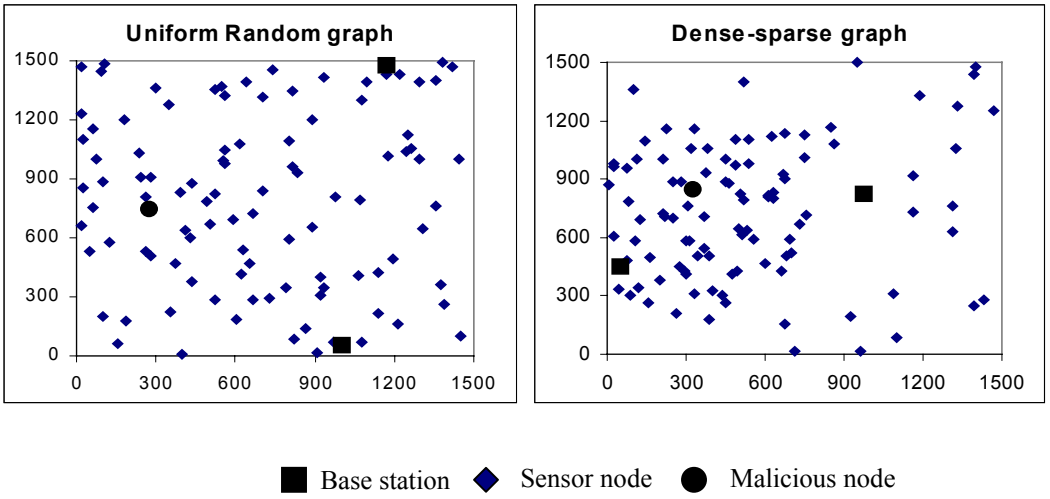

Figure 9

These heuristics are quite preliminary at present. They are based on a set of only a few (10s) experiments. Naturally, optimal placement will depend on many factors such as number of nodes, layout of the nodes, number of base stations, attack model, and so on. We recognize that this is an important research area and our future research plans include a detailed investigation of this issue.

6 RELATED WORK

Sensor network security is a critical issue in sensor network research. Ganesan et al propose a redundant multipath routing approach for a sensor network [4] in order to provide fault tolerance and reliable data dissemination. Deng et al. [12] propose to use multipath in sensor network to tolerate intrusions to sensor network. Chan et al. [21] propose to use multipath for the two nodes communication in sensor network. We have proposed a routing scheme that uses multiple paths and multiple base stations to tolerate the attacks to sensor node and base station.

In the field of ad hoc wireless networking, previous work on secure routing employs public key cryptography to perform authentication [15, 16, 17, 18]. SEADS [6] and Ariadne [7] use symmetric cryptography, a one-way hash function, TESLA, and MACs to build secure wireless network routing. The base station centralized routing differs from traditional ad hoc wireless routing in that the former focuses on an asymmetric or hierarchical architecture with a base station and sensors, rather than on peer-to-peer routing.

SPINS [5] addresses secure communication in resource-constrained sensor networks, introducing two low-level secure building blocks, SNEP and µTESLA. The one-way hash chain in our paper is inspired by µTESLA, and we widely use it in this paper. Different from µTESLA, the one-way hash chain doesn't need time synchronization or late release, because it is used to authenticate the *request* event itself, not the content of *request* message.

Slijepcevic et. al [22], Wood et al. [11] Karlof et al. [10] provide survey papers for secure sensor network routing, and discuss many attacks on a sensor network. [10] point out that a multiple base station solution and dynamic placement of base stations. However, the authors haven't explored these topics in detail. This paper also proposes using a shared key for the whole network to guard against the intruders. Our mechanism on the other hand can provide pair-wise keys to each pair of neighboring sensor nodes. This can provide much more secure protection.

Staddon et. al [9] propose a protocol for base station to get neighborhood information of sensor nodes. This protocol doesn't consider the case of sensor node compromises. Deng et. al [12] use a secure network setup algorithm similar to our route discovery protocol, to find the network topology. Our routing protocol also

differs from [12] in that it supports multiple base stations. [12] proposes multiple path routing to povide intrusion tolerance in sensor networks. However, [12] is based on single base station.

7 CONCLUSION

In this paper, we have addressed the issue securing a wireless sensor network against a variety of threats that can lead to the failure of the base station, which represents a central point of failure. First, multipath routing to multiple destination base stations is analyzed as a strategy to provide tolerance against individual base station attacks or sensor node compromises. Second, confusion of address and identification fields in packet headers via hashing functions is explored as a technique to help disguise the location of the base station from eavesdroppers. Third, relocation of the base station in the network topology is studied as a means of enhancing resiliency and mitigating the scope of damage. We have done extensive experimentation with all three strategies both via simulation in ns2 and implementation on Berkeley MICA sensor motes. The results from these experiments show that a wireless sensor network can be secured quite well against attacks on base stations and compromises of sensor nodes.

REFERENCES

[1] A. Mainwaring, J. Polastre, R. Szewczyk D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", First ACM Workshop on Wireless Sensor Networks and Applications (WSNA) 2002, pp. 88-97.

[2] F. Martin, B. Mikhak, and B. Silverman, "MetaCricket: A designer's kit for making computational devices," IBM Systems Journal, vol. 39, nos. 3 & 4, 2000.

[3] ARGUS Advanced Remote Ground Unattended Sensor Systems, Department of Defense, U.S. Air Force, http://www.globalsecurity.org/intell/systems/arguss.htm.

[4] D. Ganesan, R. Govindan, S. Shenker and D. Estrin, "Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks," *Mobile Computing and Communication Review (MC2R) Vol 1., No.2. 2002*.

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar, "SPINS: Security Protocols for Sensor Networks," *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM 2001*, July 2001.

[6] Y. Hu, D. Johnson, A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," Fourth *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02).*

[7] Y. Hu, A. Perrig, D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002).*

[8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, "System architecture directions for network sensors", *ASPLOS 2000*.

[9] J. Staddon, D. Balfanz, G. Durfee, "Efficient Tracing of Failed Nodes in Sensor Networks", *First Workshop on Sensor Networks and Applications, WSNA'02*, Atlanta, Georgia, USA.

[10] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[11] A. Wood, J. A. Stankovic, "Denial of Service in Sensor Networks," *IEEE Computer*, 35(10):54-62, October 2002.

[12] J. Deng, R. Han and S. Mishra, "The Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks", *to appear in IEEE 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, Palo Alto, CA, USA, April, 2003.

[13] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet, *In ASPLOS 2002*, 2002.

[14] Y. Hu, A. Perrig, D. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols", Technical Report TR01-384, Department of Computer Science, Rice University, June 2002.

[15] J. J. Kong, P. Zerfos, H. Luo, S. Lu, L.X. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks", *International Conference on Network Protocols (ICNP 2001*).

[16] P. Papadimitratos, Z. Haas, "Secure Routing for Mobile Ad hoc Networks", *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simula-tion Conference (CNDS 2002).*

[17] L. Zhou, Z. Haas, "Securing Ad Hoc Networks", *IEEE Network Magazine, vol. 13, no.6, November/December 1999.*

[18] K. Zhang, "Efficient protocols for signing routing messages", *In Proceedings of the Symposium on Network and Distributed Systems Security (NDSS '98)*, San Diego, Cali-fornia, March 1998.

[19] NS2 Web Site, http://www.isi.edu/nsnam/ns.

[20] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler, "The nesC Language: A Holistic Approach to Networked Embedded Systems", *To appear in Proceedings of Programming Language Design and Implementation (PLDI) 2003*, June 2003.

[21] H. Chan, A. Perrig, D. Song, "Random Key Predistribution Schemes for Sensor Networks", *Appears in IEEE Symposium on Security and Privacy 2003*.

[22] S. Slijepcevic, V. Tsiatsis, S. Zimbeck, "On Communication Security in Wireless Ad-Hoc Sensor Networks", *Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02),* June 2002, Pittsburgh, Pennsylvania, USA.

[23] Y. C Hu, A. Perrig, D. B. Johnson, "Packet Leashes: A Defense against ormhole Attacks in wireless Networks", *Appears in IEEE Infocom 2003*, 2003.