Security for Sensor Networks

Jeffery Undercoffer, Sasikanth Avancha, Anupam Joshi and John Pinkston
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250
{junder2, savanc1, joshi, pinkston}
@cs.umbc.edu

Abstract

Sensor networks have been identified as being useful in a variety of domains to include the battlefield and perimeter defense. We motivate the security problems that sensor networks face by developing a scenario representative of a large application class where these networks would be used in the future. We identify threats to this application class and propose a new lightweight security protocol that operates in the base station mode of sensor communication, where the security protocol is mindful of the resource constraints of sensor networks. Our application class requires mitigation against traffic analysis, hence we do not use any routing mechanisms, relying solely on broadcasts of end-to-end encrypted packets. Our protocol extends the broadcast range of the base station model by utilizing nodes adjacent to the base station as an intermediary hop. Additionally, our protocol detects and corrects some classes of aberrant node behavior. We have simulated our protocol and present simulation results.

1 INTRODUCTION

Improvements in wireless networking and micro-electro-mechanical systems (MEMS) are contributing to the formation of a new computing domain – distributed sensor networks. These ad-hoc networks of small, fully programmable sensors will be used in a variety of applications: on the battlefield, as medical devices, in equipment maintenance and in perimeter security systems [4, 6]. These distributed sensor networks are characterized by limited power supplies, low bandwidth, small memory sizes and a different traffic model. The traffic model of mobile ad-hoc networks is typically many-to-many whereas the traffic model of a sensor network is more of a hierarchical model and/or many to one. Generally, MEMS are significantly more resource constrained than typical "mobile" or "handheld" devices. A node in a sensor network may or may not have computing requirements. In the case where computations are required, if the cost of a communication is less than the cost of the computation the computation may be replaced by a request to a computationally robust central location. The threat to a sensor network is different from the threat to a mobile ad-hoc network. As such, existing network security mechanisms, including those developed for Mobile Ad-Hoc Networks, are a poor fit for this domain. Research into authentication and confidentiality mechanisms designed specifically for sensor data and network control protocols is needed. Given the fact that little prior work ([12] being the exception) exists in this space, there is a need both to identify the problems and challenges and propose solution techniques.

In this paper, we motivate the security problems that sensor networks face by developing a scenario representative of a large application class where microsensor networks would be used in the (near) future. We identify the threats and vulnerabilities to this class of applications, starting from the radio layer and progressing to the application layer. This paper details why security mechanisms that are presently used in mobile ad-hoc environments are inadequate or not appropriate for sensor networks. We then describe a new security protocol that serves as a countermeasure to the identified threats. Our model of sensor network utilizes the central base station model and is mindful of the resource constraints of sensor networks. We have implemented our protocol, using SensorSim [11] and present our simulation results.

Section 2 details related research in the area of security for sensor networks. Section 3 details our class of application, stating how sensor networks are a solution. Section 4 details our security protocol for sensor networks. Section 5 details our implementation and simulation results. We state our plans for future work in Section 6.

2 RELATED WORK

There is relatively little work in the area of securing sensor networks. Like their mobile ad-hoc counterparts sensor networks lack a fixed infrastructure and the topology is dynamically deployed. Addressing the security of mobile ad-hoc networks, Yi et al. [7] point out that if the routing protocol can be subverted and messages altered in transit, then no amount of security on the data packets can mitigate a security threat at the application layer. Consequently, they introduce "*Security Aware Ad-hoc Routing*" (SAR). SAR characterizes and explicitly represents the trust values and relationships associated with ad-hoc nodes and use these values to make secure routing decisions. They address two problems: Ensuring that data is routed through a secure route composed of trusted nodes and the security of the information in the routing protocol. To motivate their scenario, they use the example of two military generals wishing to communicate via an ad hoc network using a generic form of the *Ad-Hoc On Demand Distance Vector Routing* (AODV) protocol. They employ a route discovery protocol where only nodes with a security metric equivalent to the sender and receiver participate in the routing process. Their work appears to be based on the *Bell-La Padula Confidentiality Model*. Their model, however, is dependent on self-enforcement where nodes with a lower than required security level voluntarily opt out of participating in the hop-by-hop routing process.

Perrig et al.[12] introduce "*SPINS: Security Protocols for Sensor Networks*" comprised of *Sensor Network Encryption Protocol* (SNEP) and μTESLA. The function of SNEP is to provide confidentiality (privacy), two-party data authentication, integrity and freshness. μTESLA is to provide authentication to data broadcasts. SPINS presents an architecture where the base station accesses nodes using source routing.

In SNEP each $node_j$ shares a unique master key K_j with the base station. This master key is used to derive all other keys. For data encryption SNEP employs a one time encryption key produced by using a key dervied from K_j and an incremental counter (message indicator) as inputs to the RC5 cryptographic algorithm. The RC5 algorithm outputs a binary string that is used as the one time key. The message is XORed with the one time key, transmitted and the counter is incremented in preparation for the next message. The base station, aware of the node's counter value and the derived key, produces the identical one time key, XORs the encrypted message with the one time key to produce the clear text.

Our protocol differs from SPINS in two fundamental and essential ways.

1. SPINS uses source routing, making the network vulnerable to traffic analysis. Our protocol relies upon broadcasts where the entire communication is end-to-end encrypted in order to mitigate against the threat posed by traffic analysis.

2. We provide a mechanism for detecting certain types of aberrant behavior, behavior that may be due to either a compromise or malfunction of an individual node. In either case we are able to remove the node from the network.

3 PERIMETER PROTECTION AS A CLASS OF SENSOR APPLICATIONS

We motivate our application class of perimeter security by considering the following scenario:

In the current political climate the threat posed to high-level U.S. Government officials is overwhelming, particularly when they are engaged in official business outside of the continental United States. Consider the unique case of the Secretary of State while representing U.S. interests in the Middle East. The political situation in the Middle East is constantly in flux, often necessitating last-minute schedule changes that require the Secretary to travel to, or remain overnight, in a city where his security detail has not had time to conduct a detailed advance and install the requisite security controls.

The Secretary faces a number of threats. Physical threats include poisoning from radiation, chemical or biological toxins. These threats are in addition to threats from explosives and individuals utilizing small arms or other military ordinance. More insidious threats are posed by collection efforts aimed at both the substance of the Secretary's agenda and those aimed at analyzing security controls in order to compromise them in order to harm the Secretary at some later time.

Such perimeter security applications represent a vast class of "monitoring and responding" type applications for which sensor networks will be used. We believe that a solution designed to mitigate the aforementioned threats will also cover the broad spectrum of all threat models that such applications face.

Our application, and its attendant security protocol, may be abstracted and applied to other scenarios where concentric circles of perimeter protection need to be temporarily established. Examples of other applications are placing alarm fields on the U.S. border and at U.S. ports of entry in order to prevent the smuggling of hazardous materials and munitions and to prevent illegal entry. For example, suppose information is developed leading to the suspicion that some person or persons unknown will be attempting to move fissionable material across a remote section of the U.S. border. Furthermore, suppose that this material produces a signature but the signature is only discernible at distances ≤ 20 meters. Such a scenario is within the realm of possibility. Moreover, it is conceivable that those behind such an operation could be leaking spurious information so as to cause the authorities to take precautionary measures, only to study, analyze and circumvent future preventive measures, in effect a "dry run" in preparation for the actual event. A sensor network could be rapidly deployed in order to prevent and apprehend those involved in such an event. However, the underlying architecture of the sensor network must be able to withstand probes and analyses in order to remain effective over time.

3.1 Sensor Technology as a Solution

Sensors are still some time away from actual mass fabrication and use. Current smartsensor prototypes, such as the Berkeley Renee Mote, have a larger footprint and are generally restricted to the following sensor types: accelerometers, microphones, light, motion and magnetometers. We envision a design inclusive of these sensors types; however, our proposed application assumes a (not so) futuristic scenario that include sensors that test for nitrates (explosives), chemical toxins, biological toxins and radiation. For example, sensors testing for motion and sound would be placed on rooftops, which provide an assailant(s) with a vantage point. Sensors testing for radiation would be placed in areas frequented by the protectee, while sensors testing for chemical and biological toxins would be placed in airways that feed into areas frequented by the protectee. Our colleagues at UMBC in the Department of Chemical and Biochemical Engineering are deploying such sensors [7], and we feel that such sensors will eventually be integrated with sensor prototypes.

Our contribution in this area is the creation of lightweight techniques for securing existing sensor network routing and data movement approaches, such as directed diffusion [5], spin [8] and data dissemination [9, 2]. We assume the computational capability and memory requirements typical of those provided by current generation of sensors. We note that in this work we make no attempt to counter the threat from a widespread denial of service attack against the RF layer. As pointed out in [12], such attacks are fairly straightforward to mount against fixed frequency RF communication links that are found in the sensors. Defending against them requires changes to the RF layer of the sensor; such as the use of spread spectrum techniques, which can mitigate against an RF level DoS attack. In our scenario, and many others, a denial of service is in itself an alarm.

The typical security problems that we might expect in the above scenario include:

i. Passive Information Gathering: If communications between sensors, or between sensors and base stations, are in the clear, then an intruder with an appropriately powerful receiver and well designed antenna can easily pick off the data stream. If information has to be encrypted, then "how" is the open question. In other words, which cryptographic approaches will be best, given the resource constraints and the routing paradigm employed by the sensor?

ii. Subversion of a Node: A particular sensor might be "captured", and information stored on it (such as the key) might be obtained by an adversary. If a node has been compromised then how to exclude that node, and that node only, from the sensor network is at issue. The smart card community has done significant work to address such situations that may be adapted to the security solutions for sensor networks.

iii. False Node: An intruder might “add” a node to the system that feeds false data or prevents the passage of true data. While such problems with malicious hosts have been studied in distributed systems, as well as ad-hoc networking, the solutions proposed there (group key agreements, quorums and per hop authentication) are in general too computationally demanding to work for sensors.

iv. Legitimate Addition of a Node to an Existing Sensor Network: If a node needed to be replaced or another node needed to be added to an existing sensor network, securely integrating the new node into the existing network is at issue.

3.2 Assumptions

We make the following assumptions when applying sensor technology as a solution to our application class of perimeter security:

- The base station is computationally robust, having the requisite processor speed, memory and power to support the cryptographic and routing requirements of the sensor network. The base station is part of a trusted computing environment.
- The communication paradigm is either base station to sensor or sensor to base station.
- The radio range of a sensor is 15 meters.
- Given the radio range of a sensor, the single hop area of coverage with the base station at the center is: 706 square meters, πr^2.
- The sensing range of a sensor is 1 meter, providing an area of coverage of 3.141 square meters.
- Given the single hop area of coverage provided by the base station and a sensor’s area of coverage, it will require 224 sensors to saturate the one hop area.
- The physical security protocol of our application class follows the traditional model of concentric circles of protection, where the the inner circle is resource rich. Conversely, as the distance from the inner circle increases the controls and mechanism become more general and are deployed in fewer numbers.

Given the above assumptions, it would require 224 sensors to saturate the one hop radius of the base station and 337 sensors to cover the two hop area of the base station with a saturation rate of 0.5. It is, however, unlikely that this many nodes would be deployed in our application. Accordingly, a sensor network would require an address space of 10 bits to accommodate the network.

4 SECURITY PROTOCOL

Security is a broadly used term encompassing the characteristics of authentication, integrity, privacy, non-repudiation, and anti-playback. In the case of our sensor network the security requirements are comprised of authentication, integrity, privacy (or confidentiality) and anti-playback. The recipient of a message needs to be able to be unequivocally assured that the message came from its stated source. Similarly, the recipient needs to be assured that the message was not altered in transit and that it is not an earlier message being re-played in order to veil the current environment. Finally, all communications need to be kept private so that eavesdroppers cannot intercept, study and analyze, and devise counter measures in order to circumvent the purposes of the sensor network.

Our approach to defining a security protocol for sensor networks is resource driven and factors in the trade offs between levels of security and the requisite power and computational resources. Primarily, we envision a scenario where a protected perimeter based on sensors is dynamically deployed. However, similar scenarios could be envisioned in an environment where the topology is well known in advance and the sensor network is pre-configured. Our operating paradigm is where data is reported to a computationally robust central location such as a base station or network controller.

4.1 Single Collection and Authentication Point (Base Station) Model

Consider the family of sensor routing protocols where each sensor communicates either directly or indirectly with a base station. In turn the base station correlates and aggregates information from each sensor. Accordingly, the base station will need to verify the authenticity of the sensor, the integrity of the communication and ascertain that it is not a replay of an earlier communication. Recall the assumption that the base station is computationally robust and secure. In our protocol each sensor $\boldsymbol{j}$ shares a unique 64 bit Key $\boldsymbol{Key}_j$, with the base station. Our protocol provides for a multi-hop scenario where the range of a base station is extended employing nodes that are adjacent to the base station to serve as intermediaries for non-adjacent nodes. Figure 1 depicts an example of such a sensor network topology.

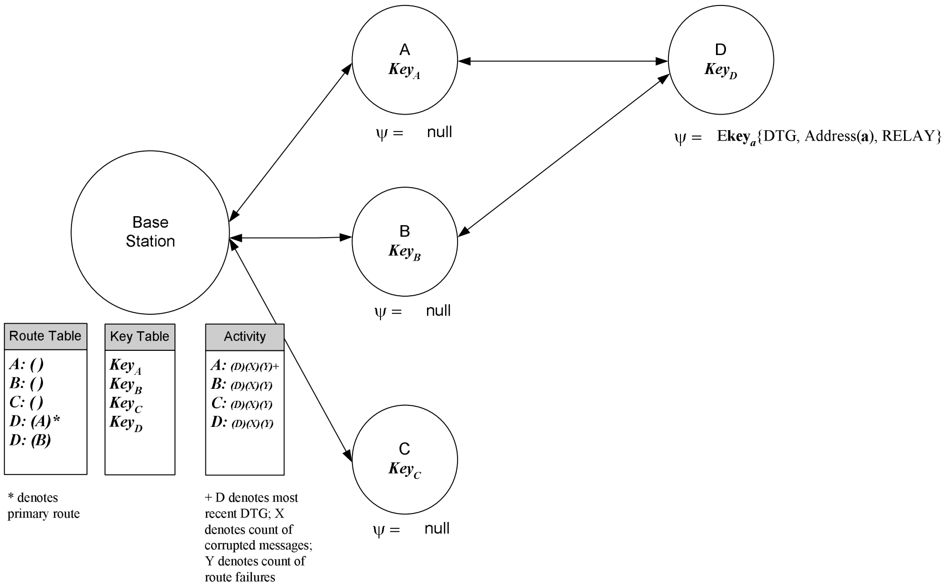


Figure 1. Example Network Topology

Our goal is to provide privacy and integrity to the data, to authenticate the sender, to prevent replay attacks and to prevent traffic analysis; consequently, the entire communication is encrypted (with one exception which will be explained). All communications consist of a preamble, header and payload. The format of all communications (sensor nodes and the base station) consist of a preamble, header and payload. The preamble is empty if the communication originates from the base station and is directed to a sensor, otherwise it contains the address of the sending node. The header contains the recipient's address, nonce and a command and is encrypted under key $\boldsymbol{K}_j$, which is shared between the base station and node j. The payload contains data exchanged between the node and the base station. As will be explained, the payload is encrypted under the shared key of the destination node, which may be different from the key used to encrypt the header. This difference comes into play when the communication needs to be relayed by an intermediate node. Figure 2 depicts the communication format.

Where:

$$\overbrace{<\text{Addr\_1( ),}}^{\text{Preamble}}\quad \overbrace{\textbf{\textit{Key}}_{j}\{\text{Addr\_2(}\mathbf{j}\text{)}, \textit{DTG}, \textit{COMMAND}\}}^{\text{Header}}\quad, \overbrace{\text{E}\,\textbf{\textit{Key}}_{j}\{\text{data}\}>}^{\text{Payload}}$$

Figure 2. Message Format

- Addr_1 is empty if the communication is from the base station to a sensor.
- Addr_1 contains the address of the transmitting node if the communication is directed to the base station. The inclusion of Addr_1 enables the base station to immediately select the correct key; instead of trying keys until it locates the correct one.
- Addr_2 contains the address of the destination node if the communication is from the base station to a node. If the communication is from a node to the base station Addr_2 will contain the address of the sending node.
- DTG is the date-time-group and is a nonce used to prevent replay attacks.
- COMMAND is a command to the sensor.

4.2 Topology Discovery and Network Setup

The base station is deployed with the unique ID and symmetric encryption key of each node in the micro sensor network. Similarly, each node is deployed with the unique key that it shares with the base station and, as in SPINS, its clock is synchronized with the base station's clock. Upon initialization of the sensor network the base station learns the network topology, creates and optimizes a routing table and provides a mechanism to non-adjacent (out of radio range) nodes that enables them to securely reach the base station.

At start up, the base station sends a *HELLO* message to each node. If the node replies with a *HELLO-REPLY*, then the node is adjacent to the base station and the base station adds that node to its route table. Those nodes that did not reply are assumed to be more than one hop away and non-adjacent to the base station. For these non-adjacent nodes, the base station sends a message containing the *RELAY* command and a payload, to be forwarded to the non-adjacent node, to each of the adjacent nodes. In a *RELAY* message, the header is encrypted under the adjacent node's key and the payload, which encapsulates a header and payload intended for the non-adjacent node, is encrypted under the non-adjacent node's key. The relaying (adjacent) node prepends the original preamble to the payload and transmits the new message. The header of the message received by the non-adjacent node contains a *HELLO* command and the payload contains a mechanism that will be used by the non-adjacent node to reach the base station through the intermediate (adjacent) node.

This mechanism, referred to as Ψ, is a pre-built header containing the *RELAY* command encrypted under the adjacent node's key. To respond to the *HELLO* message, the non-adjacent node constructs a *HELLO-REPLY* message encrypting it under the key it shares with the base station and places it in the payload. The preamble containing the base station's address and Ψ are prepended to the payload and the message is transmitted. In turn, the adjacent node receives the transmission, decrypts the header and upon seeing the *RELAY* command, prepends the preamble to the payload and transmits it to the base station. Once the base station discovers which nodes are adjacent to it and all of the paths by which the non-adjacent nodes are reachable, it optimizes its route table so as to not overburden an adjacent node with the task of relaying messages. If the optimization process results in a different route, the base station sends the affected non-adjacent node an updated Ψ. The secure topology discovery and network setup algorithm is presented in Figure 3:

Table 1 lists the commands that we have implemented.

As depicted in Figure 1, the base station maintains three tables. Their purpose and function follows:

The **Route Table** contains the primary route, indicated by an *, and alternate routes to a node. An entry of the form A:() indicates that the node is directly connected to the base station whereas an entry such as D:(A) indicates

```
C $\leftarrow$ all sensors in Sensor Network
Route Table $\leftarrow \phi$
Temp Route Table $\leftarrow \phi$
$\forall$ j $\epsilon$ C do
    Base Station $\rightarrow$ j : <Addr_1(), E$\mathbf{Key}_j$ {Addr_2(j), DTG, HELLO}, null >
    if ( j $\rightarrow$ Base Station : <Addr_1(j), E$\mathbf{Key}_j$ {Addr_2(j), DTG, HELLO-REPLY}, null >) then
        Route Table $\leftarrow$ Route Table + j()
        C $\leftarrow$ C - j
$\forall$ k $\epsilon$ C do
    $\forall$ j $\epsilon$ Route Table do
     Base Station $\rightarrow$ j : <Addr_1(), E$\mathbf{Key}_j$ { Addr_2(j), Null, RELAY}, E$\mathbf{Key}_k${ Addr_2(k),
     DTG, HELLO} >
     j $\rightarrow$ k : <Addr_1( ),E$\mathbf{k}_k$ { Addr_2(k), DTG, HELLO}, $\psi$} >
     if k $\rightarrow$ j : Addr_1( ), header, payload > where:
         header = $\psi$ = E$\mathbf{Key}_j$ {Addr_2(j), null, RELAY}
         payload = Addr_1(k), E$\mathbf{Key}_j${DTG, Addr_2(k),HELLO-REPLY }, null >

     j $\rightarrow$ Base Station: < Addr_1(k), E$\mathbf{Key}_k$ { DTG, Addr_2(k), HELLO-REPLY}, null >) then
      Temp Route Table $\leftarrow$ Temp Route Table + k(j)
Optimize(Temp Route Table)
Route Table $\leftarrow$ Route Table + Temp Route Table
$\star$ Note: The DTG is only verified by the final destination consequently it is null for intermediate
nodes.
```

Figure 3. Secure Topology Discovery and Network Setup Protocol

Command Name	**Function**
HELLO	Polling message to learn the network topology.
HELLO-REPLY	Response to the network topology polling message.
RELAY	Prepend Addr_1 to the payload and transmit.
UPDATE-PSI	Replace the stored Ψ with the new Ψ.

Table 1. Protocol Command Set

that A is an intermediate node between the base station and node D.

The **Key Table** contains the unique key shared by node j and the base station.

The **Activity Table** contains the most recent Date Time Group (DTG) received by the base station from a particular node, a count (X) of corrupted messages sent by the node, and a count (Y) of other nodes dependent upon this node to relay messages. The values of X and Y are used to detect aberrant behavior on the part of an individual node.

We use a cipher text auto-key system employing a 64-bit key for data encryption. This type of cryptosystem is detailed in [3]. Accordingly, the strength of a cryptosystem is dependent upon both its key length and the soundness of the encryption algorithm. Current cryptographic doctrine recommends using keys of 128 bits, however this requirement is predicated upon the notion that the encrypted communication remain secure against cryptanalysis and brute force attacks for a 30 year period. In contrast, we require that the sensor network's communications withstand a brute force attack for the life of the network and a short period thereafter.

To prevent traffic analysis, the entire communication is encrypted (with the exception of the preamble which is null except for traffic intended for the base station). Accordingly, a node will need to decrypt all communications that it “hears”. This adds very little overhead because when the node decrypts the first 64 bits of the the message, the recipient’s address (Addr_2) is revealed. If a valid address is present then the node will continue to decrypt the message, otherwise it will discard it. It

As previously stated, authentication is achieved through the use of a shared secret, which is the 64 bit key $\mathbf{K}_j$, shared between the base station and node ***j***. Message integrity is achieved through the selection of an encryption algorithm that exhibits strong properties of diffusion and confusion. Accordingly, an attack aimed at altering the message will only be against the form of the message and not its substance. Anti-Play back is achieved by the use the *Date-Time-Group*. Finally, privacy is achieved as a result of encrypting all communications.

4.3 Inserting Additional Nodes into the Network

The insertion of an additional node into the existing sensor network is easily accomplished. In our model the unique identity and the key ***K*** of the node to be added are loaded into the base station, the new node’s clock is synchronized with that of the existing network and the base station repeats the topology discovery algorithm.

4.4 Isolating Aberrant Nodes

An aberrant node is one that is not functioning as specified. Identifying and isolating aberrant nodes that are serving as intermediate nodes is important to the continued operation of the sensor network.

A node may cease to function as expected for several reasons, to include:

- It has exhausted its source of power.
- It was damaged.
- It is dependent upon an intermediate node and is being blocked because the intermediate node has fallen victim to 1 and 2 above.
- It is dependent upon an intermediate node and is being deliberately blocked because the intermediate node has been compromised.
- An intermediate node has been compromised and it is corrupting the communication by modifying data before forwarding it.
- A node has been compromised and it communicates fictitious information to the base station.

Our protocol effectively mitigates against the class of attack/failure where an intermediate node is involved. The protocol for the detection of aberrant nodes is presented in Figure 4:

In order to mitigate against an intermediate node that corrupts the data relaying to the base station, the base station keeps a counter of corrupted packets. Such a tactic constitutes a denial of service against the sending node because the base station will most likely request a retransmission, consequently depleting the node of power.

Periodically, the base station checks the activity table associated with a node, testing for a prolonged period of inactivity and for a high incidence of corrupted messages originating from the node. If the node is directly connected (i.e.: it does not rely upon an intermediate node) this could be evidence of aberrant behavior on the part of the node. If the node relies upon an intermediate node this could be evidence of aberrant behavior on either the part of the node or the intermediate node.

In either case, the base station polls the node. If the base station does not receive a poll-reply within the time out period it will re-poll the node via an alternative path, if it exists. If it receives a reply it will transmit a new ψ to the node which reflects the alternate route and increments the intermediate nodes counter (Y) of route failures.

```
$\forall$ j $\epsilon$ { (Current Time $T$ - DTG) $> \Delta$ } OR $Activity_{\mathbf{X}} > Threshold$ do
    Base Station $\rightarrow k_{primary} \rightarrow$ j : $POLL$
    if j $\not\rightarrow$ Base Station : POLL-REPLY then
        Base Station $\rightarrow k_{alternate} \rightarrow$ j : $POLL$
        if j $\rightarrow$ Base Station : POLL-REPLY then
            Base Station $\rightarrow k_{alternate} \rightarrow$ j : UPDATE-PSI $+\psi_{k_{alternate}}$
            $k_{primary}$ $Activity_{Y}$ $++$
        else
            Base Station $\rightarrow k_{primary}$ : $POLL$
            if $k_{primary} \rightarrow$ Base Station : POLL-REPLY then
                Route Table = Route Table - $j$

$\forall$ j $\epsilon$ Route Table do
    if $Activity_{\mathbf{Y}} > Threshold$ then
        Route Table = Route Table - $j$
```

Figure 4. Network Repair Algorithm

5 IMPLEMENTATION DETAILS and SIMULATION RESULTS

We have implemented the topology discovery and network setup components of our protocol using the SensorSim [11] extension of the NS network simulator [10]. All of our protocol, to include cryptographic functionality is implemented at the routing layer.

5.1 Simulation

We used the simple radio and battery models of the simulator. This model assumes a current draw of 12 mA to transmit a message, 1.8 mA to receive a message and 2.9 mA for the CPU to process a message. We also assume a data rate of 19,200 kbps and message length of either 24 or 48 bytes.

We conducted experiments to measure energy expenditure of each sensor function (*Tx*, *Rx* and *CPU*) during network setup time for four different network topologies. We simulated a geographic environment measuring 5654 square meters. We divided this environment into two concentric circles, the inner circle with a radius of 15 meters and the outer circle with a radius of 30 meters. The base station was located at the center. The sensor placement within our experimental topologies is as follows:

1. 30 nodes randomly placed in the inner circle and 70 nodes randomly placed in the outer circle.

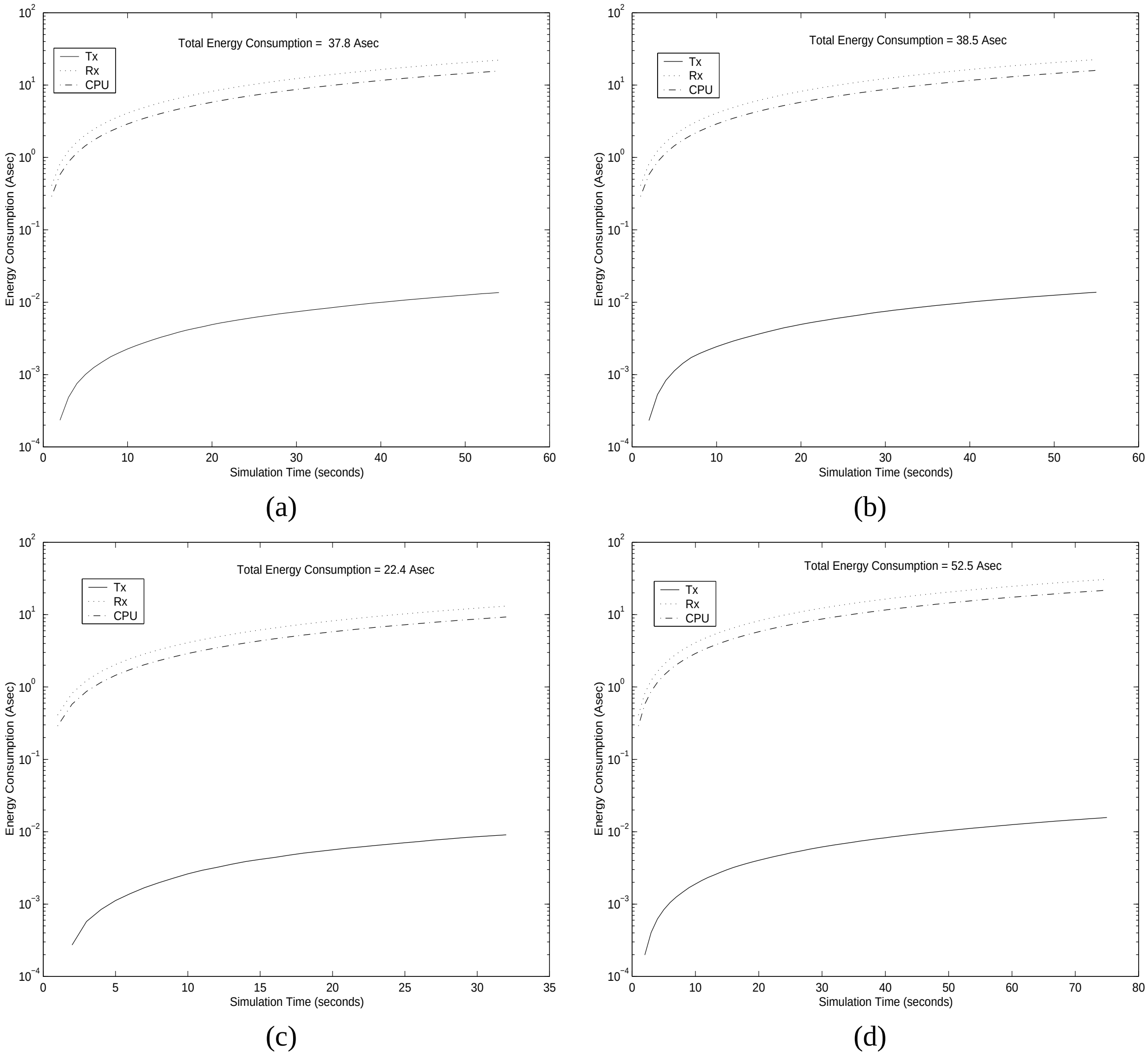
2. 50 nodes randomly placed in the inner circle and 50 nodes randomly placed in the outer circle.

3. 70 nodes randomly placed in the inner circle and 30 nodes randomly placed in the outer circle.

4. 100 nodes randomly placed across the entire area.

The results of our experiments are illustrated in the graphs contained in Figure 5.1. We measured the energy consumed (*Y axis*) by each component of the sensor: the transmitter, receiver and CPU, for the entire network of 1base station an 99 sensors, plotting it against the time taken (*X axis*) for the particular network topology to converge. We used a log plot so that that small values would be discernible.

Accordingly, Figure 5.1 (a) shows the results for Topology #1. Topology discovery and network setup occurred in 54 seconds of simulation time with a total energy expenditure of 37.8 $Asec$ for all nodes in the sensor network. Figure 5.1 (b) shows the results for Topology #2. It took 55 seconds of simulation time for topology discovery and network setup and the total network energy consumption was 38.5 $Asec$. The energy expenditure and time taken



(a)

(b)

(c)

(d)

Figure 5. (a) 30 Adjacent and 70 Non-adjacent Nodes. (b) 50 Adjacent and 50 Non-adjacent Nodes. (c) 70 Adjacent and 30 Non-adjacent Nodes. (d) Random Distribution of Nodes

for Topology #3 is illustrated in Figure 5.1 (c). Here energy consumption was 22.4 $Asec$ and it took 32 seconds. The random distribution of sensors in Topology #4 took 75 seconds of simulation time and energy consumption was 52.5 $Asec$.

In all cases, and as anticipated, the receiver (*Rx*)component consumed the highest amount energy, closely followed by the CPU. The transmitter (*Tx*) component consumed the least amount of energy. This is intuitive, as the number of messages received greatly outweighs those transmitted.

The topology with the densest inner circle and the sparsest outer circle consumed the least amount of energy and converged the quickest. The topology scenario that was most representative of the methods used for physical protection (30 inner nodes and 70 outer nodes) was near the median for time and energy consumption. Our results indicate that as the ratio of adjacent to non-adjacent nodes increases in favor of adjacent nodes, energy consumption for topology discovery and network setup decreases.

The cost of network setup, in terms of energy consumption, is the most expensive period due to the volume of messages. However, energy consumption decreases from this point forward for the life of the sensor network. To put the energy requirements into perspective, suppose that a sensor network using our security protocol were to maintain its peak rate for a protracted period. If this were the case then each sensor equipped with a battery similar to the Eveready *X91* with a capacity of 3,135 mAh would sustain the for approximately 435 hours (Note:

the Berkeley Renee Mote uses two of these batteries [1]). As our network would have a required lifespan of a few days, this time period is well within the bounds of the requirements for our application class.

6 CONCLUSIONS and FUTURE WORK

A novel scenario defining perimeter protection as an application class of sensor networks was presented. We identify threats to this application class and proposed and implemented a new security protocol that operates in the base station mode of sensor communication. We simulated our protocol using SensorSim and the results indicate that the protocol is viable and well suited for our application class.

We have not implemented the protocol to repair the network from the effects of an aberrant node and are continuing work to do so. Once implemented we will conduct experiments to determine the overall cost of the network as well as the effects of aberrant sensors on the efficacy of the network.

References

[1] Mote. http://kingkong.me.berkeley.edu/ nota/RunningMan/Mote.htm. Page from the Smart Dust program giving an overview of the Berkeley Renee Mote.

[2] Philippe Bonnet, J. E. Gehrke, and Praveen Seshadri. Towards sensor database systems. In *Second International Conference on Mobile Data Management*, 2001.

[3] Dorothy Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[4] Ball Semiconductor Inc. *Medical Applications - Benefits of Spherical Geometry*, 1997.

[5] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, 2000.

[6] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile networking for smart dust. In *ACM/IEEE Intl. Conf. on Mobile Computing and Networking*, Seattle, WA, August 1999. Mobicom 99.

[7] Yordan Kostov and Govind Rao. Low cost optical instrumentation for biomedical measurement. *J. Review of Scientific Instruments*, 2000.

[8] Joanna Kulik, Wendi Rabiner, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *5th ACM/IEEE Mobicom Conference*, 1999.

[9] Sam Madden, Michael J. Franklin, and Fjording. The stream: An architecture for queries over streaming sensor data. In *ICDE Conference*, 2002.

[10] Netowrk simulator. http://www-mash.berkeley.edu/ns, 1996.

[11] S. Park, A. Savvides, and M.B. Srivastava. Sensorsim: A simulation framework for sensor networks. In *The Third ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2000.

[12] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J.D.Tygar. Spins: Security protocols for sensor networks. *Wireless Networks*, 8:521 – 534, 2002.