A FRAMEWORK FOR IMPLEMENTING SECURITY IN WIRELESS SENSOR NETWORKS

by

KAMRAN JAMSHAID

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2002

MAJOR: COMPUTER SCIENCE

Approved by:

_____

Advisor Date

Acknowledgments

The work would not have been possible without the valuable advice and guidance of my advisor Dr. Loren Schwiebert. He has been a wonderful source of inspiration and ideas, and I dare not think what it would have been like without his feedback and help. Supervising nine graduate students and simultaneously following a full teaching load can never be easy, and sometimes I wonder how he still manages to greet everyone with a smile!!!

I do not think it will ever be possible for me to accomplish anything without the love and support of my parents, for their kind words of encouragement keep prodding me along...

Finally I would like to thank my fellow lab members in NeWS Lab for putting up with me. I know it would have been hard at times, but I guess they did a pretty good job! :-)

Contents

Chapter 1

Introduction

1.1 What are Wireless Sensor Networks

Over the last few years, technological advances in the design of processors, memory, and radio technology have propelled an active interest in the area of distributed micro-sensing, in which a number of independent, self-sustainable nodes collaborate to perform a large sensing task. A network of these devices can be large in scale, changing dynamically yet still maintaining robust communication connectivity. More commonly referred to as Wireless Sensor Networks (WSNs), this particular aspect of distributed networks has been the subject of extensive recent research with their use being advocated for a wide variety of applications. Researchers are contemplating their widespread deployment in challenging scenarios where wired networks are infeasible or impractical. For example, military interest in the network of smart sensors is motivated by many problems that can be safely and effectively solved by smart sensors (Schwiebert, Gupta, and Weinmann 2001). Such a network of smart sensors could be deployed in combat scenarios to track troop movements. Sensors placed on small robots could conduct landmine detection. Smart sensors could detect the use of biological or chemical weapons and, via network communication, report their presence in time to protect troops.

Besides military usage, many useful and varied applications of sensor networks are also being developed for our everyday lives. Biomedical sensors are being developed

for a retinal prosthesis to aid the visually impaired (Schwiebert, Gupta, and Weinmann 2001). Another example, on which the current research is based, is for pollution detection along beaches, with smart sensors distributed along the shoreline. During times of heavy rains, for example, overloaded combined sewer systems may discharge a mixture of raw sewage, polluted runoff, and litter from streets and, in some cases, industrial waste waters, into local waterways where it can contaminate downstream beaches making them unsuitable for swimming and surfing (The Beach Program). The detection of the presence of viruses, bacteria, and other pathogens early enough can significantly lower the public's risk of illness. The sensors implanted along the beach continually analyze water samples over regular periods of time and can notify a central control facility in case substantial deviations from an acceptable range of values are observed.

1.2 Communication Primitives in Wireless Sensor Networks

We envisage that a typical wireless sensor network will comprise of a number of sensor nodes scattered over a large area. Each of the sensor nodes will be a self-sustained, independent identity. (Intanagonwiwat, Govindan, and Estrin 2000) predict that a typical sensor node in the future will have the following set of features: a matchbox-sized device, powered by a battery source, with a power-conserving processor clocking at several hundred megahertz, program and data memory amounting to tens of megabytes, and a radio modem with an energy efficient MAC. These "bare-bone" devices would be capable of being fitted with one or more sensing devices, e.g. photoelectric diodes to measure light intensity, motion detectors for troop movements, etc.

On detecting an event of interest, these sensor nodes will collect the relevant parameters associated with the event into a data packet and then transmit it to a base station. Because of a number of reasons discussed in the following sub-section,

the message will only be broadcast to the neighboring sensor node, which then further relays the packet to its neighbor in the direction of the base station as indicated by the routing protocol. Traversing hop by hop, the message finally reaches the base station, which then analyzes the sensed data readings to take further action as appropriate.

This form of reverse multicast (data readings from multiple sensors arriving at a single base station) will primarily be "simplex in nature". The base station will have powerful transmitters which enable it to reach a node in the network directly, while a message from the same node traveling upstream to the base station could be transported only hop by hop with the help of other nodes in the network.

1.2.1 Why multiple hops?

In wireless communication where obstacles in the line of sight between the source and the receiver can result in a drastic drop in signal strength, multihop communication patterns can be helpful in ensuring access to all areas of the network. In addition, while dealing with an unreliable, high bit-error-rate channel like the wireless medium, there are some special communication considerations which should be taken into account. One way of ensuring more reliable and efficient communication in these channels is through the use of multihopping. Multihopping communication facilitates the reuse of resources in both spatial and temporal domains, provided that the nodes which participate in the network are reasonably well distributed in space (Frodigh, Johansson, and Larsson 2000). In contrast, single-hop networks mainly share the channel resources in the temporal domain only. This sharing of channel resources enables the multihop network to provide greater spectral efficiencies, resulting in better bandwidth utilization with a lower probability of collisions.

Multihopping also enables us to derive maximal energy savings in the network by controlling the transmission power and limiting the broadcast over a short distance.

The power received, P_R, at a receiving antenna with a gain G_R is

$$P_R = \frac{P_T G_T}{4\pi d^2} A = \frac{\lambda^2}{(4\pi d)^2} G_T P_T G_R$$

where A is the effective area or aperture of the antenna, $G_R = 4\pi A/\lambda^2$, and the wavelength $\lambda = c/f_c$ where c is the velocity of light and f_c is the carrier frequency.

The equation shows that the minimum received power for a given signal to noise ratio is inversely proportional to the square of the distance between the sender and the receiver. For a given P_T, the received power, P_R, will decrease by an order of 4 if the distance between the sender and the receiver increases by an order of 2. Thus a transmission over a short hop distance is more energy conserving than a direct communication between two end points in the network.

Another direct benefit of controlling power over a short range is that it can reduce the total interference level in a homogeneous multihop network with multiple communicating nodes and fixed traffic (Frodigh, Johansson, and Larsson 2000). This lesser interference can result in higher throughput and improved Quality of Service (QoS) in the network. In summary, multihopping is beneficial because it:

- conserves transmit energy resources
- reduces interference
- increases overall network throughout
- allows access to all nodes in the network

1.3 An Example: Environmental Sensor Network

Our work is based on an example of environmental sensors that have been deployed to check the pollution contents along an aquatic recreational area. Small sensor nodes, implanted along the shoreline, sample water specimens and conduct internal tests to measure bacteria levels. Significant deteriorations in water quality are

alerted to an on-shore processing station via radio communication. Such a network topology consists of a linear node placement in which the actual distance between each node has been predetermined to yield the most effective system performance. These factors include, among other things, the terrain characteristics affecting the signal propagation, the maximum range of radio transceivers, and fault tolerance of the network (the number of faulty or corrupted nodes that might need to be "hopped over" in order to reach out to a secure node). In addition, short spacing between the sensor nodes provides a low probability of interception (LPI) (Carman, Kruss, and Matt 2000). Typically we assume that the spacing between the sensor nodes will be in the range of 100-150 m to ensure a reasonable trade-off between the various system requirements. A typical network topology is shown in figure 1.1.

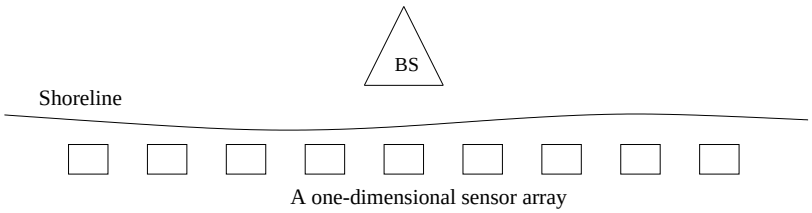

Figure 1.1: Network topology for environmental sensor networks

Placed in this topology, each of the sensor nodes will have two immediate neighbors, one on each side. A sensor node depends on its two immediate neighbors to help carry its messages to the required destination, i.e, each sensor node acts as a repeater for its neighbor, receiving data and helping propagate this data further in the network.

1.4 Securing Wireless Sensor Networks

Though the focus of recent research on WSNs has been on making such networks feasible by extending their lives using energy-conserving communication mod-

els (Lindsey and Raghavendra 2002) (Heinzelman, Chandrakasan, and Balakrishnan 2000) (Salhieh, Weinmann, Kochhal, and Schwiebert 2001), little effort has yet been extended to determine how these networks would actually survive the tough rigors of real world challenges. Security is of paramount importance in these types of devices, especially in applications where strategic decisions are expected to be based on information received from these sensor nodes. It would be impractical to assume that these devices are not foolproof. A group of sensors deployed on a battle ground may be compromised by an adversary so as not to send any information, or even worse, to send misleading information. An employer, while conducting interviews "hacks" into the personal network of prospective employees and gets confidential information about their health-related issues. These issues bring to light some of those aspects of WSNs which will help determine whether such devices do actually gain widespread utility. Such networks can be of practical use only if the readings from these devices can actually be trusted so that the decisions based on them are sound and not suffering because of concocted or invalid information.

Though computer and network security has been an extensively researched field, now having established standards and protocols, scaling many of the existing protocols down to the environment of sensor networks introduces new problem areas. Because of a number of similar characteristics, many of these issues are easiest to relate with the constraints of implementing security protocols over ad-hoc networks.

In the present work, our effort is to identify why the existing security solutions will not work in the environment of sensor networks. Our work presents various approaches to scale down the security overhead by suggesting efficient implementations of key set-up, point-to-point and broadcast communication, and routing protocols. The rest of this thesis document is organized as follows. Chapter 2 outlines some of the operational requirements of sensor networks that force us to look beyond the solutions available for traditional computer networks. Chapter 3 focuses on bootstrapping

a device with a number of keys that will be used when the device joins the network. The proposed protocol, SEKEN, enables a sensor node to establish a key with the base station and its neighbors in an energy efficient manner. The chapter also evaluates the performance comparison of SEKEN with some other key setup protocols in terms of energy efficiency. Chapter 4 provides the communication paradigms which should be adopted to achieve secure communication semantics. It describes secure mechanisms for secure unicast and broadcast, the two communication primitives that we believe will be used most often in wireless sensor networks. In addition, the chapter also discusses threats to a distributed routing protocol and suggests a mechanism for ensuring secure route selection between the devices. Chapter 5 outlines some of the related work in the area and suggests why some of the proposed solutions do not meet our application requirements.

Chapter 2

Security Issues in Wireless Sensor Networks

In this chapter, we identify some of the issues which present interesting challenges for security implementation in sensor networks. The chapter begins with a brief explanation of the some of the objectives required in a secure communication protocol. This leads to a discussion of why the existing solutions would fail to scale in the unique application realm of wireless sensor devices.

2.1 Secure Communication Requirements

Security is an important factor for the kind of sensor network applications we outlined in chapter 1. In this section, we formalize the secure communication requirements expected in a sensor network. A communication model allowing provisions for implementing these properties within its various protocols assures protection against the kind of attacks to which these types of networks are vulnerable.

1. *Data Confidentiality* Data confidentiality implies protection from disclosure to authorized entities. In our context of sensor networks, this means that we should adopt suitable mechanisms to prevent intermediate or non-trusted nodes from understanding the contents of the packets as they are relayed from one node to the other (Yi, Naldurg, and Kravets 2001). Confidentiality is generally achieved by encrypting the data packets with a secret shared between the two end points of communication.

2. *Data Integrity* In a network there can be multiple sources of message corruption, including benign failures such as radio propagation impairment, or malicious attacks by adversaries (Zhou and Haas 1999). Message integrity helps us maintain data consistency by guaranteeing that a message being transferred was never corrupted. Integrity is usually achieved using Message Digest functions and appending the digest to the data message. The receiver verifies the integrity of the message by recomputing the hash on the data packets and comparing it with the value received.

3. *Data Authentication* Data authentication allows a receiver to verify that the data really was sent by the claimed sender. This is essential in cases where the receiver needs to ensure that all data used in any decision making process originated from the correct source (Perrig, Szewczyk, Wen, Culler, and Tygar 2001). Without authentication, an adversary can masquerade as another node and inject malicious messages into the network or gain unauthorized access to resources or some other sensitive information.

4. *Data Freshness* Informally, data freshness implies that the data is recent and it ensures that no adversary replayed old messages. This is required for time-varying data like sensor readings, routing update messages, or other critical updates where the adversary can gain by replaying old messages. Timestamping data packets when they are generated at the source is perhaps one of the most primitive yet effective mechanisms to check the freshness of the data. Weak data freshness (which only assures the ordering of the packets) can be achieved by keeping track of the sequence number of the packets.

Though there are other security goals (validation, anonymity, authorization, etc.) which might be of concern to some other applications, yet we expect the majority of the sensor network security requirements can be achieved through the properties

described above.

2.2 Security in Wireless Networks

To counter the security threats in a wireless network, certain actions may be taken which are broadly classified into two categories (Jiang, Vaidya, and Zhao 2000):

- Prevent the transmission signals from being detected through the use of spread-spectrum modulation techniques, effective power control, and directional antennas.
- Protect the information transferred in the network through cryptographic methods.

Completely eliminating the possibility of signal detection is a difficult task. While the main goal of spread spectrum communication is to guard against an unauthorized reception or interference of signals, it still requires messages to be transferred hop by hop between the nodes and hence could sacrifice some of the semantics of the required security policies (e.g, confidentiality of data). So, encryption plays a pivotal role in ensuring communication security by providing appropriate mechanisms at higher layers in the protocol stack. However, in order to yield a more secure protocol, we believe that encryption could take advantage of similar services being provided by other layers (e.g, spread spectrum communication at the physical layer).

2.3 Research Challenges

Sensor networks have stringent constraints bound by their cost and performance factors. The following unique characteristics of sensor networks provide us with an impetus to think beyond the current solutions available for securing computers and other communication peripherals (Carman, Kruss, and Matt 2000).

2.3.1 Physical security of the devices

Most of the sensor networks being envisaged for future deployment are small, inexpensive devices deployed in challenging scenarios where the possibility of the devices being compromised cannot be ruled out. Lying unattended in the open, the device could be an easy target for powerful adversaries who can "pry open" the sensor node using significantly stronger cracking tools. Not only are these adversaries capable of physically damaging the device, rendering it nonfunctional, but it is not improbable that they can even alter device characteristics/mechanisms to send out data readings of their choice.

The issue is further complicated by the difficulty of differentiating trustworthy nodes from compromised ones. A compromised node is perhaps still capable of generating otherwise valid test information and distributing it around in order to appear functionally stable. This prevents nodes from taking punitive measures against their corrupt neighbors so that they continue rely on the fake information being fed to them. Left undetected, a compromised node continues inflicting damage in the network for a longer duration of time.

In some of the related papers, tamper-resistant nodes have been suggested as a possible solution to this problem. However, proposing such a property is different from its actual implementation, especially when we consider the cost and the performance characteristics of these devices (Stajano and Anderson 1999). Tamper-proofing (e.g, detecting a broken seal) is another suggested solution, though it does not ensure that the compromised node will be noticed in time to prevent any damage. Also it needs to be confirmed that appropriate effective actions will be taken to maintain the secrecy of the previously agreed keys and the cached data, if any.

2.3.2 Scalability

Network scalability is another important factor while designing the security protocols for the wireless sensor devices. It is envisaged that the sensor networks will have hundreds of nodes spread over a wide area. The security implementation should be such that it does not add a significant overhead to the overall working of such a large network.

Similarly, changes in the network membership need to be supported in an equally efficient manner. A device (sensor node) joining/leaving the network should be transparent to the network as a whole and a minimum amount of information should have to be reconfigured. Contributory key establishment protocols might not be directly applicable in our network scenario where having such a large number of network nodes might actually slow down this process. Advantage can, however, be taken of a trusted third party, e.g, the local base station, which is assigned the responsibilities of generating a random session key and securely distributing it.

2.3.3 Limited computational and communication resources

Most of the sensor nodes deployed in the open will be battery-powered devices. Depending on their role within the network, the duration of usage, and the sensitivity of operation, some or all of these nodes might have some power recharging mechanisms (e.g. solar powered cells). So in order to ensure long and efficient device operation, battery consumption will have to be reduced to a minimum. This suggests that drastic power-conserving methodologies will have to be adopted at all levels of device operation.

In terms of power consumption, radio communication is more expensive than computation. Pottie *et al.* (Pottie and Kaiser 2000) have deduced that the energy cost of transmitting 1Kb over a 100m distance is the same as the energy required by a general-purpose 100MIPS/W processor to efficiently execute 3 million instructions.

Our protocol will have to minimize exchange of security-related setup messages in order to enhance efficiency. As an example, some of the key agreement algorithms suggested in the literature involve frequent exchange of set-up messages before a key is mutually agreed upon (Steiner, Tsudik, and Waidner 1996). Many of these "handshake" messages involve transmission of large data packets, comprising of large random numbers in order to agree on a reasonably secure session key. Keeping our stringent energy limitations in mind, such protocols cannot be used in the kind of sensor networks we have described. Similarly, the choice of cryptographic ciphers employed for encryption should reflect the need to save on both computation and communication resources.

The sensor nodes might need to perform aggressive data aggregation and compression to cut down on some of these costs. Similar data generated from multiple nodes need not be conveyed to the base station individually. Also perhaps not all of the sensor node communication needs to be encrypted. Only information regarded as critical for the network functionality or mission success needs to be sent securely. This might include routing information or other critical data warranting immediate action from the mission control (e.g, a base station).

2.3.4 Changing network topology

The security implementation chosen for these devices will also have to take into consideration changing network topology. Consider a scenario in which a group of environmental sensors have been placed along a shoreline. These sensors are tossed about their position by the incoming tide at different times of the day, leading to loss of line of sight and resulting in the network getting partitioned (Steere, Baptista, McNamee, Pu, and Walpole 2000). Similarly, with the passage of time, some of the sensor nodes might drain out their battery resources or simply develop a fault that renders them useless. Also there is a possibility that some of these nodes are detected

to be compromised and hence should not be allowed to be a part of the communication network.

This places a new challenging constraint on our network model. The network as a whole assumes a changing topology, where some of the nodes become unavailable for some time duration. Maintaining network connectivity under these circumstances introduces a new set of problems. A sensor node with a compromised, faulty, or unavailable neighbor needs to be able to discover nodes beyond its immediate neighbors in order to get its messages across. It also needs to develop a new secure relationship with the set of nodes it discovers. Checks certifying the authenticity of the new node need to be carried out before the nodes agree to negotiate a session key. Routing information would need to be updated to reflect the new topology. This dynamically changing topology introduces new problem areas which have not been investigated before in the context of secure sensor devices.

2.3.5 Device Constraints

Most of the sensor nodes will be low cost, small devices with limited computational and memory resources (Schwiebert, Gupta, and Weinmann 2001). This places a stringent constraint on our cryptographic primitives. Storing long cryptographic keys (to ensure realistic security) and performing operations with them are not only resource exhausting but perhaps even impossible to perform with such devices. A typical sensor node will have its memory shared, among other things, by the device operating system (including device testing and trouble shooting routines) and sensor application software (Perrig, Szewczyk, Wen, Culler, and Tygar 2001). This leaves the sensor node with little memory for the implementation of many of the commonly available security routines and primitives.

Under these constraining circumstances, we rule out the use of asymmetric cryptography to encrypt regular data. Symmetric cryptography uses a smaller key size,

is orders of magnitude faster, and is not susceptible to chosen-ciphertext attack (Schneier 1996). This allows the two parties to exchange larger amounts of data in a shorter span of time with a lower drain on the device resources.

It has to be remembered that security is only an auxiliary operation: it exists to strengthen our communication model. This obviously makes the security implementation take a back seat if a contention for device resources develops. Depending on the system requirements, a compromise might need to be reached on satisfactory resource utilization.

Chapter 3

SEKEN: A Key Exchange Protocol for Sensor Networks

In this chapter, we propose a key set-up protocol with a view toward optimizing energy utilization for a key set-up procedure in a typical sensor network. SEKEN (Secure and Efficient Key Exchange for sensor Networks) has been developed keeping in mind the stringent resource and communication requirements of wireless sensor networks. The protocol has minimum resource consumption for securing key exchange between two neighboring sensor nodes. We compare SEKEN against two other contemporary key set-up solutions and show that SEKEN provides considerable power savings without compromising security or system scalability.

3.1 SEKEN

3.1.1 Assumptions

Before describing the protocol, let us identify the assumptions underlying the working of our model. We assume that the radio model is symmetric (Lindsey and Raghavendra 2002). In other words, for a given signal-to-noise ratio, the energy required to transmit an m bit message from node A to node B is the same as the energy required to transmit the same m bit message from node B to A. In addition, we assume that the base station has significantly larger resource availability compared to a regular sensor node. Specifically, since the base would be housed on-shore, it can run on utility electricity and use more powerful computers with larger memory

and processing power (Steere, Baptista, McNamee, Pu, and Walpole 2000). The base station can keep a record of all the keys it shares with each of the sensor nodes and can use these keys to send confidential messages to a sensor node. Because of readily available electric power, the base station can also make long range transmissions to reach individual nodes anywhere within the sensor network. However, in order for messages to travel upstream (i.e, from a sensor node to the base station), the message has to hop from node to node in order to maximize the energy conservation.

We also make some assumptions about the general architecture and the trust requirements of our sensor nodes. Firstly, we assume that the sensor nodes are created with a unique device identifier (DId) which is known only to that particular sensor node. The device identifiers of all the nodes have to be manually fed into the base station and each DId acts as an initial shared secret between that device and the base station. Like MAC addresses in Network Interface Cards (NICs), this device identifier field should be sufficiently long so that every sensor node can be assigned a value which would be unique yet random enough to prevent an adversary from impersonating it by carrying out a brute force attack. This DId is only used during the bootstrapping process and is never exchanged in cleartext, hence ensuring that this identifier is never explicitly disclosed to any other sensor node. Device tamper resistance mechanisms might have to be employed in order to ensure that the memory is flushed out if any attempt is made to physically manipulate the device in order to retrieve this data. In addition, we assume that the public key of the base station has been pre-deployed within the sensors. Sensor nodes can conveniently be programmed with this key before their actual deployment in the field. This is essential because it makes the availability of an omnipresent Certification Authority (CA) redundant.

3.1.2 Notation

We will use the following notation to illustrate different operations in our cryptographic operations.

- A message M encrypted with key K is represented as $E_K(M)$.
- $E_{PUB}(M)$ is an encryption of message M with the base station's public key.
- $A, B1, B2$ are examples of node IDs. Node IDs are different from the Device IDs (DIds) in the sense that the former is only a temporary tag assigned by the base station while the latter is a more permanent identifier for the device.
- $N_1$and N_A are examples of a nonce (a random bit string).
- $H(M)$ is a one-way hash function computed over a message M.

3.1.3 The Protocol

We define three basic message types that help the nodes to set-up a key among themselves. Each of these message-types is identified by a unique message identity field in the message header. A node wishing to join the network sends a "join-network" message. Similarly, a node after successfully negotiating its authentication with the base station, authenticates itself with its neighbors using an "authenticate-me" message. Finally, a node which fails to receive a response from its previous neighbor sends an "update-neighbor" message to the base station. Each of these messages is identified by a unique header that prompts a suitable action at the appropriate network devices.

The protocol works in the following steps. The node closest to the base station initiates the key setup phase. It retrieves its device identifier from memory, appends the current timestamp to it, and encrypts the entire packet with the base station's public key. By this point in time, the sensor node has calculated a local copy of the

key, K_A, it will be sharing with the base station by computing K_A = MAC(DId, TS). It waits a random amount of time before transmitting the previously generated encrypted packet to the base station. The base station decrypts the received message with its corresponding private key and checks its database for a device with the same ID. On confirming the validity of the device, the base station computes MAC(DId, TS) to generate its own copy of the key proposed by the sensor node. It responds with the following housekeeping information that enables the sensor node to become a valid, working network entity: this includes the node ID, ID_A, and a counter, C_A, initialized to some random value. The node ID is a unique temporary device identification for this particular network only and helps with the routing of messages. Such an ID can be a geographical representation of the node's location within the sensor network. The counter value is used in a MAC to generate a session key between this node and its neighbor and is incremented at both the base station as well as at the sensor node after generating a key between this node and its neighbors. For data confidentiality, all this information is encrypted with K_A before transmission.

The first node that manages to complete the key set-up procedure with the base station acts as a gateway for all the other nodes in the network. Because of its proximity to the base, all messages destined toward the base station have first to be carried over to this node. In the next leg of the protocol, the remaining sensor nodes in the network go through a similar procedure to establish secure relationships among themselves. The new sensor node (assume $B1$) wishing to join the network appends the current timestamp to its DId, encrypts the result with the public key of the base station, and computes a local copy of K_{B1}. The encrypted packet is broadcast and received by the closest neighbor in line toward the base station with the help of the underlying routing protocol. The neighbor appends its own ID to the message (so as to help the base station estimate the rough location of the new node) and the message is finally transported to the base station. The base station performs the

routine validity checks on the node, computes the key proposed by the sensor node, and then sends it the information it will need to be a part of the network. In addition to ID_{B1} and C_{B1}, the base station also sends node $B1$ the key K_{A-B1} it will share with its neighbor A. For example, in figure 3.1, the broadcast message received by the node $B1$ is $E_{B1}(K_{A-B1}, ID_{B1}, C_{B1})$, where $K_{A-B1} = MAC(K_A, C_A)$.

Once this information is available at the node, it attempts to authenticate its neighbor using a challenge-response scheme. Node $B1$ generates a nonce N_1 encrypted with the key K_{A-B1} and transmits it to its neighboring node A. Node A, on detecting a "authenticate-me" message, computes its own copy of $K_{A-B1} = MAC(K_A, C_A)$ and responds with the original nonce N_1 and a new nonce N_2, both encrypted with the new agreed key, K_{A-B1}. To complete node A's authentication, $B1$ responds with the nonce N_2 encrypted with the shared key, K_{A-B1}. This completes the authentication process for both of the new sensor nodes $B1$ and A. The same process is then carried out for all the remaining sensor nodes as they join the network. For example, in response to node $C1$'s request, the base station responds with $\mathrm{E}_{K_{C1}}(K_{B1-C1}, ID_{C1}, C_{C1})$. This means that node $C1$ will eventually share $K_{B1-C1} = MAC(K_{B1}, C_{B1})$ as a key with node $B1$. The SEKEN protocol is illustrated in figure 3.1.

3.1.4 Node Addition and Removal

The SEKEN protocol has been designed to enable a cost-efficient network reconfiguration whenever a node enters or leaves the network. In this sub-section, we delve into some of the details which enable SEKEN to accomplish membership addition or removal in a network. Suppose that a network node wants to attach itself to this chain of sensor nodes by appearing in between two existing nodes. For example, a node $T1$ joins the network between the nodes A and $B1$ in figure 3.1. It issues a "join-network" message to which the node A appends its own ID and forwards it to the base station just like for any other node. The base station maintains the topo-

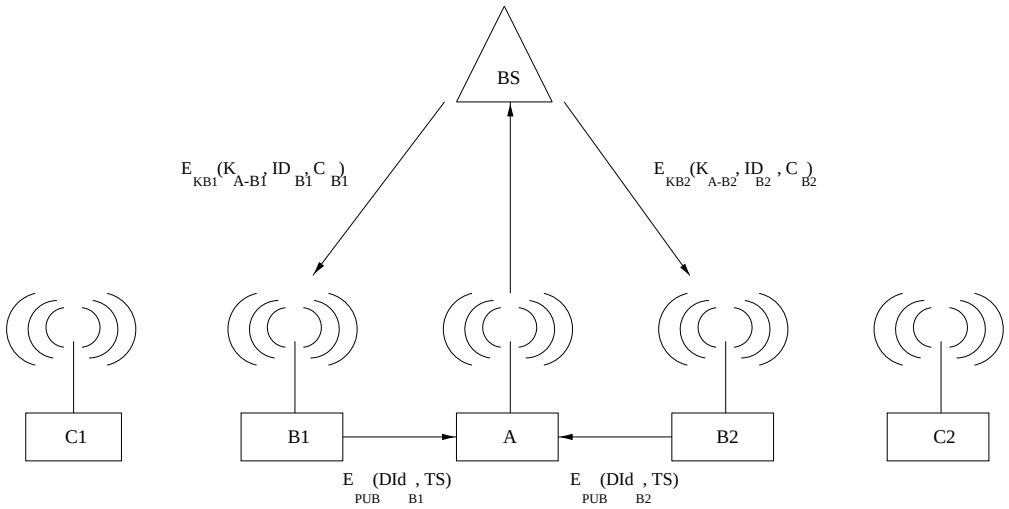

Figure 3.1: Message exchange for SEKEN protocol. Nodes B1 and B2 are setting up a secure key with the base station. The gateway node A has already received $E_{K_A}(C_A, ID_A)$ from the base station, BS

logical graph of the whole network, which helps it to discover that a new node has been appended between two existing nodes. Along with sending it the routine network configuration information (ID_{T1} and C_{T1}), the base station also sends the MAC values computed over both of its neighbor's keys and their current counter values to act as a shared key between them. After receiving this information, the new node authenticates itself to each of its two neighbors as explained in equations 3.13-3.15.

Now assume that the node $T1$ has been displaced and is no longer within the radio range of its neighbors A and $B1$. Assuming that the packet acknowledgment is done on a hop-by-hop basis, node $B1$ discovers that it has lost contact with its neighbor $T1$. It generates a discover-neighbor packet that is received by node A, which again appends its own ID to the request and sends it to the base station. The base station discovers that node $B1$ is already a part of the network. It simply calculates a new key between the two sensor nodes A and $B1$ and sends it to $B1$. Node $B1$ then authenticates itself to node A using the procedure outlined above.

In this section, we compare the efficiency of SEKEN against some other common key set up protocols and then compare their corresponding energy costs. Our results indicate that SEKEN shows good performance characteristics against some of the existing key set up protocols without affecting system scalability.

One of the simplest protocols which promises to provide the most energy-efficient solution to key set-up is pre-deployment of keys before the sensor networks are put into active operation (Carman, Kruss, and Matt 2000). In this case, the sensor nodes would already share the cryptographic keys, so the protocol involves only node authentication using a challenge response scheme. Though this protocol has a minimum overhead (as only a few short messages need to be exchanged), it raises scalability and security concerns especially for changing mission configurations. For example, new nodes being added to the network must be configured with the same key material as the nodes already deployed. Additionally, if a need arises for two different sensor networks to communicate with each other, the key material of one of these networks needs to be overwritten with that of the other. Secure methods would need to be developed to carry out this operation effectively with sensor nodes actually deployed in the field. The steps in authentication are shown as follows:

$$B \rightarrow A : E_{K_{AB}}(N_A) \tag{3.1}$$

$$A \rightarrow B : E_{K_{AB}}(N_A, N_B) \tag{3.2}$$

$$B \rightarrow A : E_{K_{AB}}(N_B) \tag{3.3}$$

We also compare the SEKEN protocol against a Kerberos key exchange set up between two parties. In this protocol, instead of pre-configuring each sensor node with an explicit key for its neighboring sensor nodes, each node shares only a long-term pairwise key with a trusted server *a priori* (Menezes, van Oorschot, and Vanstone

1996). We assume that the server plays the role of a KDC (Key Distribution Center) and itself proposes the session key. A Kerberos version 5 protocol simplified for the sensor network environment is shown below. Some of the fields have been omitted to enhance the efficiency of the protocol.

$$B \rightarrow T : B, A, N_B \tag{3.4}$$

$$T \rightarrow B : ticket_A, E_{K_{BT}}(K, N_B, A) \tag{3.5}$$

$$B \rightarrow A : ticket_A, authenticator \tag{3.6}$$

$$A \rightarrow B : E_K(T_B) \tag{3.7}$$

Ticket_A is defined by $E_{K_{AT}}$ (K, B), while authenticator = E_K(B,T_B), where K_{AT} or K_{BT} is the key shared between T and A or T and B respectively, K is the session-key chosen by T and T_B is a timestamp from B's local clock.

The SEKEN protocol is summarized as follows: Initially the gateway node A authenticates itself with the base station.

$$A \rightarrow BS : E_{PUB}(DId_A, TS) \tag{3.8}$$

$$BS \rightarrow A : E_{K_A}(C_A, ID_A) \tag{3.9}$$

In round 2, the rest of the sensor nodes authenticate themselves. Here we show the sequence of steps for setting up a key for sensor node $B1$.

$$B1 \rightarrow A : E_{PUB}(DId_{B1}, TS) \tag{3.10}$$

$$A \rightarrow BS : E_{PUB}(DId_{B1}, TS), E_{PUB}(ID_A) \tag{3.11}$$

$$BS \rightarrow B1 : E_{K_{B1}}(K_{A-B1}, ID_{B1}, C_{B1}) \tag{3.12}$$

This is followed by a mutual neighbor authentication carried out between the two sensor nodes.

$$B1 \rightarrow A : E_{K_{A-B1}}(N_A) \tag{3.13}$$

$$A \rightarrow B : E_{K_{A-B1}}(N_A, N_B) \tag{3.14}$$

$$B \rightarrow A : E_{K_{A-B1}}(N_B) \tag{3.15}$$

We use the first order radio model to compute the energy costs associated with transmission and reception of data packets in the sensor network. The model associates the following costs for k bits of data exchanged between the source and destination separated by a distance d. To transmit a message, a node consumes

$$\begin{aligned} E_{Tx}(k,d) &= E_{Tx-elec}(k) + E_{Tx-amp}(k,d) \\ &= E_{elec} * k + E_{amp} * k * d^2 \end{aligned}$$

To receive this message, the radio expends

$$\begin{aligned} E_{Rx}(k) &= E_{RX-elec}(k) \\ &= E_{elec} * k \end{aligned}$$

where $E_{elec} = 50$ nJ/bit is the energy required to run the transmitter or receiver circuitry, and $E_{amp} = 100$ pJ/bit/m^2 is the energy consumed for amplification of the transmitted signal.

We assume that all symmetric key sizes are 64 bits. We believe that this key length is strong enough to provide sufficient strength against a brute-force attack over the life period of the sensor network. All nonce, node IDs, and timestamps are assumed to be 32 bits in length.

Simulation programs were written to compute the amount of energy consumed for running each of these protocols over a linear sensor array under the set of conditions and assumptions identified earlier. For the context of discussion in this chapter, we compute and compare only the energy cost of communication between the different key set-up schemes. The energy consumption is the sum of the energy consumed by the sensor nodes involved in the key set-up and subsequent authentication process (i.e, transmission and reception costs for the base station are being ignored). The

Table 3.1: Energy consumed for one run of the protocol

Pre-deployed Keys	SEKEN protocol	Kerberos Key Exchange
140.8 μJ	424.0 μJ	428.8 μJ

Table 3.2: Energy consumption for larger sensor networks

Number of nodes	Pre-deployed Keys	SEKEN protocol	Kerberos Key Exchange
50	6.899 mJ	78.15 mJ	81.84 mJ
100	14.08 mJ	288.55 mJ	296.0 mJ
150	20.98 mJ	630.95 mJ	642.16 mJ
200	28.16 mJ	1.105 J	1.120 J

distance between each network entity is uniformly assumed to be 100 m. This means that the cost of transmission is computed to be 1050 nJ/pit while the cost of reception is 50 nJ/ bit under the radio model we have described above.

Table 3.1 shows the energy consumed by each protocol for authenticating a node (one node other than the gateway node).

Now consider the case for sensor networks of larger sizes. For a pre-deployed key mechanism (Carman, Kruss, and Matt 2000), each sensor node only has to authenticate its key with its neighbors. Application of SEKEN to such a sensor network has already been explained above. For the sake of consistency with our network architecture, Kerberos requires that a node requesting a secure key have its ticket request traverse node to node (hop-by-hop) until it reaches the base station. However, the response from the trusted server can reach the individual nodes directly.

The results of the protocol confirm that pre-deployment is indeed the most efficient method of authenticating two neighboring sensor nodes. These results represent the minimum amount of energy which would have to be expended in any protocol for

implementing security over such devices. However, the protocol is practically infeasible because deployment of such a network requires painstaking care and precision in which we have to ensure that the two sensors sharing a pre-defined key do eventually end up as neighbors in the field. For example, in the case of sensor devices being used in a military context, it would be much more convenient to just throw these devices from an aircraft flying over enemy territory, and leave it to the sensor nodes to organize themselves into an information sharing network when they settle on the ground.

The results indicate that the efficiency of SEKEN falls between that of Kerberos and the pre-deployed key mechanism. Though there is not a huge difference between the energy consumption in SEKEN and Kerberos, yet a Kerberos requirement that the server shares a long-term explicit master key with every sensor node is a potential drawback, specially for large size networks. No such assumption is made in the SEKEN protocol, in which all such keys are set-up within the execution of the protocol itself. SEKEN just requires all potential network nodes to share a one-time secret with the base and to pre-programmed with the base station's public key. Additionally, in SEKEN, the base station also assigns a node ID to all sensor nodes, while in our implementation of Kerberos we are assuming that there is a mechanism for a sensor node to obtain a reliable copy of the ID of the node with which it wants to establish a secure key. Thus despite communicating more useful information, the performance characteristics of SEKEN comfortably hover between an ideal protocol (pre-deployment of keys) and a modern day practical protocol (Kerberos).

Table 3.3 gives the energy consumed when a sensor node is added between two existing nodes that are already a part of the network. Pre-deployment of keys consumes a static energy of 0.2816 mJ irrespective of the node's location in the network. However, energy consumption for SEKEN and Kerberos is a function of the node's location in the network (as the membership addition request will have to traverse all

Table 3.3: Energy consumption for member addition between two existing nodes

Node position from gateway	SEKEN protocol	Kerberos Key Exchange
50	5.566 mJ	5.720 mJ
100	10.84 mJ	11.00 mJ
150	16.12 mJ	16.28 mJ
200	21.40 mJ	21.56 mJ

the way upto the base station) and are shown in this table. Once again, SEKEN shows superior performance characteristics over Kerberos.

While describing the challenges for sensor networks, we pointed out that security is only an auxiliary operation for the sensor networks. Hence, the protocols implementing the security primitives should consume minimal system resources or else they would lose their utility. A typical Duracell AA battery (e.g, MN 1500) with a rated capacity of 2.85 ampere hours operating at its nominal voltage of 1.5 V has an energy potential of 15.39kJ (Duracell)(Carman, Kruss, and Matt 2000). For the case of a sensor network with 200 nodes, SEKEN consumes 1.106 J which comes out to be just 0.0072% of the available battery power. These figures show encouraging results, suggesting that the final cost of implementing security, including cryptography and key management, would remain confined to minimal energy consumption, making security over sensor networks an efficient, desired operation.

Chapter 4

The Operational Framework

4.1 Introduction

In this chapter, we build upon securing the communication infrastructure within our sensor nodes. We will design a framework of messages that will demonstrate how the important secure communication requirements can be achieved using these message types. In particular, we will concentrate on securing the routing message exchange between the sensor nodes. Secure routing is all the more important in wireless sensor networks because of the inherent distributed nature of the sensor nodes, which implies that a certain level of trust must exist between two communicating parties in order to make any progress.

4.2 Secure Message Exchange between two Sensor Nodes

At this point in our experimental setup, we believe that more than 80% of the messages originating from the sensor nodes will be destined for the base station. Most often, these will be sampled data readings being reported to the base station for further analysis and action. In this section, we analyze various mechanisms to secure this kind of point-to-point communication. Specifically, the protocol we outline will not be confined to sensor node to base station communication, but can be used between any two parties sharing a secret symmetric key.

Traditionally, if message confidentiality is a desired communication characteristic,

then it is achieved using message encryption with a symmetric key the two parties shared with each other. However, because of the distributed nature of our problem scenario, secure message delivery requires much more than mere dependence on message confidentiality.

In (Marti, Giuli, Lai, and Baker 2000), the authors suggest a watchdog approach that enables the nodes to identify their misbehaving neighbor by passively listening when their own message is being forwarded by that neighbor. A neighbor which simply drops the packets or modifies the contents before forwarding them, is classified as a misbehaving node and further message exchange with that node is avoided. In such a scenario, simple message encryption takes care of both message confidentiality and integrity, because the previous hop node will raise an alarm if the current node modifies the contents of the packet.

However, it is not difficult to realize that such a scheme fails if we have two consecutive untrusted nodes in a routing path, as they can collude between themselves to achieve the desired results. In the sensor networks we have described, it will not be uncommon for two misbehaving nodes to strategically align their position in order to inflict maximum damage to the network. Designing against such adversaries assumes significant challenges, and the kind of simplistic approach outlined above fails to generalize to contiguous misbehaving nodes.

Diversity coding (Zhou and Haas 1999) has been suggested in the literature as a mechanism to ensure reliable message delivery in the face of network failures and other errors. By transmitting sufficient redundant information through additional routes, we enable the receiving node to be able to reconstruct the message even if some of the information is lost or corrupted along the way.

Though diversity coding can ensure both message availability and integrity, its effectiveness is hampered by the fact that the resource-constrained sensor nodes have to make redundant transmissions to deliver a message. After analyzing the pros and

cons, we conclude that message tampering can be much more damaging than simple message dropping. Inability to communicate with other fellow nodes will only force a sensor node to look for alternative data paths while relying on a modified message can have disastrous consequences on the decision making process. We propose the use of a Message Authentication Code (MAC) to guarantee against packet modification. The source computes the MAC over the original data packet and then transmits the encrypted data packet and the MAC. The destination node can verify the integrity of the packet by recomputing the MAC over it and comparing it with the received value.

$$B1 \rightarrow A1 : E_{K_{B1}}(M), MAC(M) \tag{4.1}$$

$$A1 \rightarrow BS : E_{K_{B1}}(M), MAC(M) \tag{4.2}$$

Sending a MAC with each packet results in extra transmission cost. Assuming that MD5 is used to compute the hash, this involves an extra transmission of 64 bits (8 bytes) overhead per data packet. (Perrig, Szewczyk, Wen, Culler, and Tygar 2001) suggest that since MAC also achieves integrity, we do not need to use other message integrity mechanisms such as 16-bit CRC, and thus encrypting and signing messages imposes an effective overhead of 6 bytes per message over an encrypted message with integrity checking.

4.3 Broadcast Communication

Broadcast communication is an effective way for the base station to update mission configuration parameters for the entire sensor network in a short amount of time. It is expected that the base station will request varying frequency and granularity of sensor data readings over the lifetime of the network. During periods of high alert, the base requests higher resolution data more frequently while the demand for

such data subsides during off-peak hours. A broadcast request would be an efficient mechanism to propagate time-varying network requirements in the entire network domain. The sensors can be programmed to listen to these instructive guidelines from the base station and to adjust their operation accordingly. Since these broadcasts will manipulate the operation of the sensor nodes, it is imperative that secure mechanisms be in place in order to ensure that an adversary cannot exploit it to suit his needs.

For a broadcast message, confidentiality is usually not a requirement as the data is intended for the entire network and not directed toward any specific recipient. Assurances, however, are required that both the data actually came from its reputed source (data source authentication) and that its state is unaltered (data integrity). In addition, strict timeliness guarantees have to be imposed in order to protect against message re-use or replay.

We propose the use of transaction authentication (Menezes, van Oorschot, and Vanstone 1996) for securing the broadcast primitive. It denotes message authentication augmented to additionally provide uniqueness and timeliness guarantees on data, thus preventing message replay. Digital Signatures, appended with timestamps provides the required features. Each of the sensor nodes is bootstrapped with the public key of the base station that was securely embedded into the node before it was put into operation. When the base station has a message to broadcast, it appends the current timestamp to the message and then digitally signs the copy using its private key. The message is then broadcast to all the nodes in the network. The receiving nodes verify the originator of the message with the public key and check the timestamp to rule out replay attacks.

The format of the messages exchanged is as follows:

$$BS \rightarrow Broadcast : [M, Time]_{PRI} \tag{4.3}$$

One of the assumptions during the design of this scheme is that data integrity and origin authentication cannot be separated and that presence of one implies the other.

Data which has been altered has a new source; and if a source cannot be determined, then the question of alteration cannot be settled. Integrity mechanisms thus implicitly provide data origin authentication, and vice versa (Menezes, van Oorschot, and Vanstone 1996).

4.4 Secure Routing

Routing plays a critical role in the operational stability of any network. Routing protocol packets carry control information that governs important network characteristics like network congestion, link usage, message latency, etc. In ad-hoc wireless sensor networks, routing assumes even greater significance because a node is dependent on the cooperative behavior of its neighbors to carry its messages forward. In other words, each device acts as a relay for its neighboring devices. In order to demonstrate the necessity of secure routing within the wireless sensor networks, we will introduce a number of attacks which can be carried out by an adversary to bring down the communication infrastructure (Lundberg 2000). Though most of these attacks can be successfully carried out by a single malicious node, the magnitude of loss will certainly exacerbate if a number of adversaries collude among themselves to carry out the attacks.

4.4.1 Routing Attacks

Traditional routing solutions rely on some form of distributed routing databases, being maintained either by the network nodes or by specialized management nodes (Hubaux, Gross, Boudec, and Vetterli 2001). However, in wireless sensor networks it is not possible to implicitly trust every node. In this section, we look at some of the possible attacks suggested in the literature against existing routing protocols. We analyze the loopholes these attacks exploit and then suggest remedial measures that can ensure the authenticity and the integrity of the routing information throughout

the network.

Black Hole Routing

In the black-hole routing attack, a sensor node converges all network traffic toward itself by constantly advertising shorter routes to all the other nodes in the network. This initiates routing table updates in the neighboring nodes and these updates are then further propagated to rest of the network. Because of the traffic redirection, the adversary gains valuable network information being generated by various sensor nodes and if necessary can either silently drop the packets or modify their contents before passing them toward their intended destination.

Because the routing tables have been updated "legally", the adversary does not necessarily have to switch over to a promiscuous mode and to consume resources by eavesdropping on every packet in the network. This could result in considerable resource-conservation for the adversary, which can then be directed toward carrying out other attacks against the network.

Looping in Routing

The malicious node carries out this attack by dispensing routing information that creates loops within the routing protocol. This keeps the messages circulating within the group of sensor nodes being attacked. Also known as the "sleep deprivation torture", the motive of the attacker is to drain out the battery resources of a particular set of sensor nodes that might constitute a critical path in the network connectivity. Over a period of time, a mobile adversary can carry out such attacks in different portions of the network, rendering the original homogeneous network into a group of sketchy, scarcely-populated disconnected networks.

Routing Table Overflow

In this attack, the purpose of the attacker is to create so many routes to non-existent nodes to prevent new valid routes from being created. The attack is more effective against proactive routing algorithms because they maintain routes to all nodes in the network, including those to which no packets are being sent. Traditionally, routing protocols have been proactive in nature, and hence react to any change in topology even if no traffic is affected by that change. Link status is updated by periodic transmission of control messages, and even this otherwise innocuous system primitive can be exploited by an adversary to carry out a Denial of Service (DoS) attack against nodes.

Disclosing the Network Topology

Dynamic Source Routing (DSR) is an ad-hoc network routing protocol. It is a reactive algorithm that, as its name suggests, uses source routing to deliver data packets (Frodigh, Johansson, and Larsson 2000). The headers of the data packet carry the addresses of the nodes through which the packet must pass. Though this routing mechanism eases the processing burden on the intermediate nodes on a particular routing path, it opens other avenues for exploits. An attacker eavesdropping on the network will find a DSR packet very helpful for mapping the network topology. In a military scenario, this could be equivalent to giving away the location of the network entities and hence jeopardizing the mission.

4.4.2 Secure Routing Requirements

An insight into the types of network attacks against the ad-hoc routing protocols enables us to narrow down our list of expectations in a secure routing protocol. We believe that a routing protocol with the following set of properties will be able to provide an effective guard against the kind of attacks we have outlined above.

1. *Strong Origin Authentication.* This is essential to ensure that a node does not inject invalid routing updates while impersonating another network device.

2. *Cryptographic Checksums.* We realize that we will have to protect against tampering of routing information messages. We will also need a mechanism to protect against generation of fraudulent routing information.

3. *Message Timeliness* will be required to protect against replay and message reuse attacks.

4. *Confidentiality* is generally **not** considered to be a primary requirement in routing security.

4.4.3 A Secure Routing Protocol

The purpose of our routing protocol is to enable the sensor nodes to identify paths leading to the base station. The protocol can be suitably modified to enable secure routing between any two nodes in the network. We assume that the base station periodically sends a hearbeat beacon message to the sensor nodes. The base transmits these messages with a reduced signal strength so that it is received by the closest sensor node only. The gateway node verifies the base station as the originator of the message (by the mechanism described in the next paragraph) and then rebroadcasts the packets to its neighbors further downstream. When a sensor node receives this heartbeat signal, it marks the source of its originating neighbor as the first upstream node in the hierarchy toward the base and then rebroadcasts the packet so that it trickles through the whole of the network.

In order to prevent other nodes from masquerading themselves as the base station, the heartbeat signal will have to be protected by the kind of guarantees that would enable a unique identification of its source. We employ a variation of the Guy-Fawkes protocol to authenticate the base station (Balfanz, Smetters, Stewart, and

Wong 2002). Each heartbeat message is composed of a random nonce, N_A, and a commitment to a future nonce, $H(N_{A+1})$. A sensor node authenticates the source of the current packet by computing the hash value over N_A and comparing it with the $H(N_A)$ it received in the previous heartbeat message $(N_{A-1}, H(N_A))$. A match in value confirms that both of these packets were created by the same node and no adversary could have spoofed the packet in transit. This chain of messages is self-authenticating as one message automatically authenticates the previous one. A receiver will need to be bootstrapped with one authentic key that leads on to a self-authenticating chain. One place where this can securely be conveyed to a sensor node is during the parameter-exchange process in the SEKEN protocol.

The heartbeat beacon will be a 64-bit message periodically exchanged between the sensor nodes. A more energy-efficient version of the protocol can be obtained if we can ensure some tight bounds on the message latency in the network. In this case, the nodes exchange only $H(N_A)$ among themselves that would be half the length of the previous version of the heartbeat message. The authenticating lead N_A is then broadcast from the base station at the end of a pre-defined upper bound on the message latency. Since all the network nodes would already have received $H(N_A)$ before the base station broadcasts N_A, it is trivial for a device to mark the node that first reported the correct $H(N_A)$ as its priority link to the base station. This protocol has the additional advantage that if a node temporarily goes down, it can easily regain synchronization when it resurfaces, while in the earlier protocol the node would have had to undertake a tedious message exchange with a trusted neighbor.

In our simple topology, each of the sensor nodes has only one path to the base station and under normal circumstances, a node will hear the heartbeat message only once from its upstream neighbor. However, in some other network topologies, where a node will have more than one neighbor that can act as potential routers to the base station, the sensor node recognizes only the first node from which it hears

the message and all subsequent messages are ignored. In addition to help prevent a broadcast storm, this scheme also precludes malicious nodes from redirecting traffic by first passively listening to the beacon and then transmitting it later. The beacon message from the adversary will be ignored because by then the node would have already heard a valid signal from another valid device.

Chapter 5

Related Work

SPINS (Security Protocol for Sensor networks) by Adrian Perrig, *et al.* (Perrig, Szewczyk, Wen, Culler, and Tygar 2001) is a suite of security building blocks (SNEP and TESLA) optimized for wireless communication in a resource-constrained environment. SNEP provides data confidentiality and two party data authentication along with data freshness while TESLA provides an authenticated broadcast by introducing asymmetry through a delayed disclosure of symmetric keys. Every sensor node in the network trusts the base station and shares a master key with it. However, this requires the protocol to assume that all sensor nodes have the capability to directly communicate with the base station, which might not always be true as illustrated in our application scenario. Also, the paper does not mention how the sensor node is bootstrapped with the master key it shares with the base station. Such a process could be carried out in advance over a secure medium (Perrig 2001).

In (Wong and Chan 2001), the authors introduce two Mutual Authentication and Key Exchange Protocols (MAKEPs) for establishing secure communication between a low power wireless device (a battery operated Palm Pilot) and a powerful base station (a server). Server-specific MAKEP assumes that the client has access to a certificate specific to the server from a third party. This trusted third party maintains a list of long-lived symmetric keys for all the devices and provides them with server-specific certificates on request. However, a security issue can arise because once knowing a

node A's long-lived symmetric key, the server can impersonate itself as node A to other servers. In Linear MAKEP, node A randomly chooses a sequence of integers as its secret key, generates a corresponding sequence of public keys, and obtains a signature from a trusted authority (TA) for each pair of them. A pair of these keys together with a nonce input from the server determines the new key. One potential issue with this particular technique is that the number of servers a client can access is limited by the number of key pairs it initially creates. The authors claim that this can be used to restrict resources being requested by a client in order to ensure a more fair distribution of available services.

In (Steiner, Tsudik, and Waidner 1996), the authors describe "natural" extensions to the two party Diffie-Hellman key exchange protocol. In the upflow stage of the protocol, they collect contributions from all group members (each group member performs one exponentiation). The final node in the upflow run of the protocol initiates the downflow stage by either broadcasting the intermediate values after raising them with its own exponent or transmitting the intermediate values to its downstream neighbor, which propagates the message further downstream. The broadcast version of the protocol is more efficient in terms of the number of rounds, although it assumes that one of the nodes has broadcast capability.

Carman *et al.* (Carman, Kruss, and Matt 2000) analyze a number of key set up protocols for sensor networks including Kerberos, Otway-Rees, Key Hierarchy, and other protocols. They compare these protocols based on the size of exchanged messages as well as the computational resources required for the key calculation on a number of different microprocessors. Tatebayashi *et al.* (Tatebayashi, Matsuzaki, and Newman 1989) present a Key Distribution Protocol for digital mobile communication systems. The protocol can be used in a star type network and employs public key cryptosystem for uplink channels and a secret key cryptosystem for downlink channels.

Jean-Pierre Hubaux *et al.* (Hubaux, Buttyan, and Capkun 2001) survey the

threats and possible solutions for the security mechanisms in mobile ad hoc networks. They propose a new public key distribution system suitable for these types of self-organized networks. In this new scheme, instead of relying on certificate directories for the distribution of certificates as in PGP, the authors propose to store and distribute these through individual users. Each user maintains a local certificate repository that contains a limited number of certificates selected by the user according to some algorithm. When user U wants to obtain the public key of user V, they merge their local certificate repositories, and U tries to find an appropriate certificate chain from U to V. In (Balfanz, Smetters, Stewart, and Wong 2002), the authors present a user-friendly authentication solution that relies on some prior exchange of public information over a privileged secure channel. For devices which are extremely limited in computational resources, a form of Guy Fawkes protocol is proposed which assures authentication for interactive message exchange by having A commit to (and later send) a meaningless random message to B whenever A is not in a position to know what to say next.

Zhou and Haas (Zhou and Haas 1999) exploit the inherent redundancies in ad hoc networks to improve on system security. Service availability is improved by distributing the trust over a set of servers and then allowing those servers to share and co-ordinate the key management responsibilities. The shares for each of these servers are updated over a period of time to counteract the effect of mobile adversaries. In addition, the threshold on the number of these servers required to perform a service can change over a period of time in order to adapt to changes in the network status. Some more of the security issues for wireless ad hoc networks have been identified in (Stajano and Anderson 1999). The authors envision embedded wireless connectivity in most of the future devices and discuss some means of achieving authentication, availability, and integrity in them. For key exchange, they suggest a physical contact in the pre-birth stage between the new and the master device transferring the secret

over electrical contacts.

In (Dahill, Levine, Royer, and Shields 2001), the authors demonstrate examples of specific exploits that are possible against existing ad hoc routing protocols by mobile adversaries. The authors classify the differing security needs of an ad hoc network into distinct environments (e.g, usage in military operations, emergency rescue missions, or simple provision of networking in a conference) and propose that security policies should be adopted on the basis of security requirements. The authors go on to propose a solution to a "managed-open" environment, in which nodes in an ad hoc network exchange initialization parameters in a secure infrastructure before their actual deployment. The protocol makes extensive use of asymmetric cryptography and digital signatures, with certificates being validated by a trusted certificate server. Though the authors manage to maintain most of the security requirements by adopting a tedious use of public key cryptography, it remains unclear whether the proposed algorithm will be able to yield any reasonable performance guarantees on resource-starved, low power wireless devices

S. Yi *et al.* (Yi, Naldurg, and Kravets 2001) analyze the security of ad hoc routing algorithms with respect to the protection associated with the transmission of routing messages. The authors introduce the notion of an integrated security metric that is a combination of the security attributes and trust levels, and then use this metric to influence the route selection process. The proposed Security Aware Ad Hoc Routing (SAR) protocol can be considered to be an extension to an on-demand protocol like AODV except that only those nodes that can provide the required security or the trust level can process the packet or forward it. The authors assume a simplistic approach in which a node unable to provide the required trust level simply drops the path setup request packets during the route discovery phase of the protocol. In a typical real world scenario, it would be hard to expect such an unstinted cooperation even from the nicest of the adversaries!! Because of the broadcast nature of the wireless

medium, an eavesdropper can easily impersonate fake trust levels and become a part of the "secure" path between the data source and the destination.

Another area of interesting research in ad hoc network security is Intrusion Detection Systems (IDS). Compared with wired networks where traffic monitoring is usually done at switches, routers and gateways, an ad-hoc network does not have such traffic concentration points where the IDS can collect audit data for the entire network (Zhang and Lee 2000). In addition, a high error rate over wireless links can also make it increasingly difficult to distinguish false alarms from real intrusions. The authors in (Zhang and Lee 2000) suggest the presence of individual IDS agents on each and every node, which collectively form the IDS system to defend the network. Data on a node is locally analyzed using statistical anomaly detection techniques and then a consensus on a decision can be reached using cooperative detection with neighboring nodes. Though a typical network intrusion detection system usually works on the network layer, the authors suggest multi-layered integration to take advantage of the richer semantic information available at higher layers and to be able to analyze the attack in its entirety. Though these mechanisms have been proposed for the more powerful ad hoc devices, it remains to be seen how these techniques scale in the constrained environment of sensor networks.

Chapter 6

Conclusions and Future Work

We will want our work to be considered as an attempt to encourage the scientific community in exploring the avenues for scalable yet practical and efficient mechanisms for securing the wireless sensor networks. In this work, we have made an effort to explicitly define the security requirements which they would have to contend against. Some of our proposed solutions to these challenges show encouraging results, yet we feel that advantage can still be taken of inherent system redundancies, like exploiting the levels of protection offered by the lower layers. Though some of the algorithms we propose in this work take advantage of a topology specific to our particular application, yet we feel that with adequate and proper modifications, these methods can easily be adapted to work for any network type.

Our future work will focus on improving the performance characteristics of our various proposed solutions. For example, though our key set-up protocol (SEKEN) makes minimal assumptions about the system requirements, yet it has to rely on an initial shared secret between the sensor node and the base station in order to bootstrap the sensor node. For a very large sensor network with hundreds of nodes, this can prove to be a tedious requirement where a human operator has to manually feed these numbers into the base station's data base. In addition to this problem, we are also exploring mechanisms to ensure the security of this initial secret so as to render it unforgeable.

Our secure routing protocol maintains only a single route to its neighbors, and if that route becomes unavailable for sometime, the sensor node will have to wait for a period of upto two routing beacons in order to resynchronize with the network. Depending on the frequency of these beacon messages, this forcibly penalizes the sensor node for a certain period of time. Recently, there has been active research in efficient routing protocols for these sensor networks. In order to prolong the life of different network nodes, these routing protocols transmit the data traffic along different routes so as to ensure an equitable energy consumption for every node in the network. However, our routing protocol only transmits the message to the closest neighbor of a node (which reports the beacon before the others), and this can quickly drain out the battery resources for that neighbor. We are in the process of updating our neighbor selection metric which will take into account the number of messages previously exchanged with this neighbor and its current energy level.

Intrusion detection for these sensor networks is another area which we would like to explore in the coming years. Robust intrusion detection systems can provide an additional line of defense and will tremendously enhance the survivability of the network against all sorts of adversaries.

Finally, we hope to have a working prototype for all these protocols on the sensor nodes being developed for the project.

Bibliography

Balfanz, D., D. K. Smetters, P. Stewart, and H. C. Wong (2002). Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Network and DIstributed Systems Security Symposium*.

Carman, D. W., P. S. Kruss, and B. J. Matt (2000, Sep). Constraints and Approaches for Distributed Sensor Network Security. In *NAI Labs Technical Report 00-010*.

Dahill, B., B. N. Levine, E. Royer, and C. Shields (2001). A Secure Routing Protocol for Ad Hoc Networks. In *University of Massachusetts, Technical Report*.

Duracell. Technical/OEM. http://www.duracell.com/oem/Primary/Alkaline/alkenergydens.asp.

Frodigh, M., P. Johansson, and P. Larsson (2000, Sep-Dec). Wireless Ad hoc networking - The art of networking without a network. In *Ericsson Review*.

Heinzelman, W. R., A. Chandrakasan, and H. Balakrishnan (2000, Jan). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Hawaii International Conference on System Sciences*.

Hubaux, J.-P., L. Buttyan, and S. Capkun (2001). The Quest for Security in Mobile Ad Hoc Networks. In *ACM Symposium on Mobile Ad Hoc Networking and Computing*.

Hubaux, J. P., T. Gross, J. Y. L. Boudec, and M. Vetterli (2001, Jan). Towards self-organized mobile ad hoc networks: the Terminodes project. In *IEEE Communications Magazine*.

Intanagonwiwat, C., R. Govindan, and D. Estrin (2000). Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*.

Jiang, S., N. H. Vaidya, and W. Zhao (2000, June). Routing in Packet Radio Networks to Prevent Traffic Analysis. In *IEEE Information Assurance and Security Workshop*.

Lindsey, S. and C. S. Raghavendra (2002, March). PEGASIS: Power Efficient GAthering in Sensor Information Systems. In *IEEE Aerospace Conference*.

Lundberg, J. (2000). Routing Security in Ad Hoc Networks.

Marti, S., T. Giuli, K. Lai, and M. Baker (2000). Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*.

Menezes, A., P. van Oorschot, and S. Vanstone (1996). *Handbook of Applied Cryptography*. CRC Press.

Perrig, A. (2001, July). Private communication.

Perrig, A., R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar (2001). SPINS: Security Protocols for Sensor Networks. In *Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 189–199.

Pottie, G. J. and W. J. Kaiser (2000, May). Wireless Integrated Network Sensors. In *Communication of ACM*.

Salhieh, A., J. Weinmann, M. Kochhal, and L. Schwiebert (2001, September). Power Efficient Topologies for Wireless Sensor Networks. In *International Conference on Parallel Processing*, pp. 156–163.

Schneier, B. (1996). *Applied Cryptography*. John Wiley & Sons.

Schwiebert, L., S. K. S. Gupta, and J. Weinmann (2001). Research Challenges

in Wireless Networks of Biomedical Sensors. In *Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 151–165.

Stajano, F. and R. Anderson (1999). The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Seventh International Workshop on Security Protocols*.

Steere, D. C., A. Baptista, D. McNamee, C. Pu, and J. Walpole (2000). Research Challenges in Environmental Observation and Forecasting Systems. In *International Conference on Mobile Computing and Networking*, pp. 292–299.

Steiner, M., G. Tsudik, and M. Waidner (1996, March). Diffie-Hellman Key Distribution Extended to Group Communication. In *Third ACM Conference on Computer and Communications Security*.

Tatebayashi, M., N. Matsuzaki, and D. B. Newman (1989, Aug). Key Distribution Protocol for Digital Comunication Systems. In *Advances in Cryptology - CRYPTO 89*, pp. 324–334.

The Beach Program, E. O. o. W. Primary Sources of Pollution. http://www.epa.gov/ost/beaches/2000/primary.html.

Wong, D. S. and A. H. Chan (2001). Mutual Authentication and Key Exchange for Low Power Wireless Communication. In *IEEE MILCOM*.

Yi, S., P. Naldurg, and R. Kravets (2001). Security-Aware Ad-Hoc Routing for Wireless Networks. In *University of Illinois at Urbana-Champaign, Technical Report*.

Zhang, Y. and W. Lee (2000, Aug). Intrusion Detection in Wireless Ad-Hoc Networks. In *Sixth Annual International Conference on Mobile Computing and Networking*.

Zhou, L. and Z. J. Haas (1999, Nov). Securing Ad Hoc Networks. In *IEEE Network Magazine 13(6)*.

Abstract

A FRAMEWORK FOR IMPLEMENTING SECURITY
IN WIRELESS SENSOR NETWORKS

by

KAMRAN JAMSHAID

May 2002

Advisor: Dr. Loren Schwiebert

Major: Computer Science

Degree: Master of Science

Wireless Sensor Networks are edging closer to widespread feasibility with recent research showing promising results in developing and adapting new mechanisms to suit their environment. Secure Communication between these distributed wireless devices is a desired characteristic, specially in scenarios where these sensors are being used for military and other mission-critical operations.

This work highlights some of the research challenges for extending secure communication over these resource-constrained wireless devices and points out why the current protocols do not scale well in this novel application realm. A new key set-up protocol (SEKEN) is proposed which neatly fits into most of the requirements for these device types. Our simulation results confirm the energy efficiency our protocol enjoys over some other current implementations. Our work also analyzes mechanisms to secure the underlying routing protocol in a typical hierarchical sensor network.

Autobiographical Statement

Kamran Jamshaid received his MS degree in Computer Science from Wayne State University, Detriot, Michigan and his BS degree in Electronic Engineering from GIK Institute, Topi, Pakistan. At present he is working as a Graduate Research Assistant in the Networking and Wireless Sensors Laboratory (NeWS Lab) in the Department of Computer Science at Wayne State University. He areas of reserach interests include security issues in wireless and mobile computing, internet protocols and applications and large-scale distributed systems.

Kamran is a member of ACM. He can be reached at kamran@cs.wayne.edu