SEKEN (Secure and Efficient Key Exchange for sensor Networks) [*]

Kamran Jamshaid
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: kamjam@acm.org

Loren Schwiebert
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: loren@cs.wayne.edu

Abstract

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, especially in scenarios where these sensors will be used for military and other mission-critical operations. This paper highlights some of the research challenges for extending secure communications over these resource-limited devices and points out why current protocols do not scale well in this unique application realm. A new key setup protocol (SEKEN) is proposed that neatly fits into the requirements of these device types. The performance of SEKEN is then analyzed against some other possible key setup mechanisms. Our initial results confirm that it performs better under most of the conditions anticipated for general wireless sensor networks.

1 Introduction

Wireless Sensor Networks (WSNs) have been the subject of extensive recent research with their use being advocated for a wide variety of applications. Researchers are visualizing their widespread deployment in challenging scenarios where many of the existing networking solutions would not scale efficiently. For example, military interest in a network of smart sensors is motivated by many problems that can be safely and effectively solved by smart sensors [14]. This network could be deployed in combat scenarios to track troop movements or detect the presence 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 being developed for our everyday lives. Biomedical sensors are being developed for a retinal prosthesis to aid the visually impaired [14]. Our current research forms the basis of another example in which a group of sensors distributed along the shoreline are used for pollution detection. During heavy rains, overloaded sewer systems may discharge a mixture of raw sewage, litter, and other wastes into local waterways, which can contaminate downstream beaches making them unsuitable for swimming and surfing [3]. 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.

Though the focus of recent research on WSNs has been on extending their lives using energy-conserving communication protocols [5] [8] [14], little effort has yet been extended in defining a framework for a secure communication model for these devices. Security is of paramount importance in these types of devices, especially where strategic decisions are expected to be based on information received from the sensor nodes.

In this paper, we propose a key setup protocol (SEKEN) that has been developed for optimizing energy utilization in a typical sensor network. We compare SEKEN against two other contemporary protocols and show that SEKEN provides considerable power savings without compromising security or system scalability.

2 Research Challenges

In this section we identify some unique characteristics of sensor nodes and networks that present interesting challenges for implementing security in these devices [1].

2.1 Physical Security of the Devices

Most of the sensor networks being envisaged for future deployment are small, inexpensive devices deployed in challenging scenarios where they can be an easy target for a number of attacks ranging from physical damage to alteration of device circuitry, which allows the adversary to send out data readings of its choice.

The challenge is the difficulty of differentiating trustworthy nodes from compromised ones. A compromised node is perhaps still capable of generating otherwise valid

[*] This material is based upon work supported by National Science Foundation under Grant No. 0086953.

routine information, preventing other nodes from taking punitive measures against their corrupt neighbor(s).

In some related work, tamper-resistant nodes have been identified as a possible solution to this problem [15]. Tamper-proofing (e.g., detecting a broken seal) is another possible solution, though it does not ensure that the compromised node will be detected early enough to prevent any damage. This mechanism also needs to ensure that appropriate actions will be initiated to maintain the secrecy of the previously agreed keys and the cached data, if any.

2.2 Scalability

Network scalability is another important factor that needs consideration while designing the security protocols for the wireless sensor devices. It is envisaged that sensor networks could have hundreds or even thousands of nodes spread over a wide area. Any security implementation should 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. These changes 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 most efficient in these networks 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 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). In order to ensure longer and more effective device operation, power-conserving methodologies will have to be adopted at all levels.

In terms of sheer power consumption, radio communication is much more expensive than local computation of data. Pottie *et al.* [12] have deduced that the energy cost of transmitting 1Kb over a 100 m distance is the same as the energy required by a general-purpose 100MIPS/W processor to execute 3 million instructions. Our protocol will have to minimize the exchange of security-related setup messages in order to enhance efficiency. Similarly, the choice of cryptographic ciphers employed for encryption should also reflect our overall strategy of saving 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. 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 controller ($e.g.$, a base station).

2.4 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 sensor nodes placed along a shoreline are tossed about their position by the tide at different times of the day, leading to loss of line of sight and resulting intermittent connectivity within the network [16]. Similarly, with the passage of time, some of the sensor nodes might drain their battery resources, develop a fault, or are detected to be compromised and hence should no longer be a part of the communication network.

These scenarios place a new challenging constraint on our network model, which assumes a changing topology where some of the nodes become unavailable for a period of time. Preventing network partitioning under these circumstances introduces a new set of problems. A sensor node with a compromised, faulty, or unavailable neighbor may need to discover nodes beyond its immediate neighbors in order to get its messages across. This means the node will have 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 security problem areas that have not been investigated before in the context of resource-starved sensor devices.

2.5 Device Constraints

Most sensor nodes will be small, low cost devices with limited computational and memory resources [14]. This places a stringent constraint on the cryptographic primitives employed for these devices. Storing and performing operations with long cryptographic keys (to ensure realistic security) will be resource draining, if not impossible. 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 [11]. This leaves the node with little memory for implementing many of the commonly available cryptographic routines and primitives.

Under these restraints, we rule out an extensive use of asymmetric cryptography, preferring to rely instead on symmteric cryptography, which uses a smaller key size and is orders of magnitude faster. [13].

3 Related Work

Carman *et al.* [1] analyze a number of key set up protocols for sensor networks including Kerberos, Otway-Rees, Key Hierarchy, and others. They compare these protocols based on the size of the exchanged messages as well as the computational resources required for key calculation on a number of different microcprocessors. In [4], the authors introduce a key management scheme which uses the pre-shared probabilistic deployment of a key ring in order to maintain connectivity among different members of the sensor network. This technique could have potential memory management isues on individual sensor nodes in a large scale deployment because of key ring sizes. Khalili *et al.* [7] propose a threshold, ID-based cryptosystem for ad hoc networks. The protocol relies heavily on concepts of public key encryption, and therefore does not adapt well to resource constrained sensor networks.

SPINS (Security Protocol for Sensor Networks) by Adrian Perrig, *et al.* [11] is a suite of security building blocks (SNEP and μTESLA) optimized for wireless communication in a resource-constrained environment. SNEP provides data confidentiality and two party data authentication along with data freshness while μTESLA provides an authenticated broadcast by introducing asymmetry through a delayed disclosure of symmetric keys. All the sensor nodes trust the base station and share a master key with it. The paper does not mention how the node is bootstrapped with this master key. Such a process could be carried out in advance over a secure medium [10].

Zhou and Haas [17] exploit the inherent redundancies in ad hoc networks to improve system security. Service availability is improved by distributing the trust over a set of servers and then allowing those servers to share and coordinate the key management responsibilities. The shares of each server is updated periodically to counteract mobile adversaries. Some additional security issues for wireless ad hoc networks have been identified in [15], though for key exchange they suggest a rudimentary method of exchanging shared secrets over secure electric contacts when the devices physically touch each other.

4 Environmental Sensor Networks

We consider an example of a sensor network deployed along a recreational shoreline. These sensors periodically sample water specimens and any significant deterioration in water quality is alerted to an on-shore processing station via radio communication. The 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 a secure node) [1]. Typically, 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.

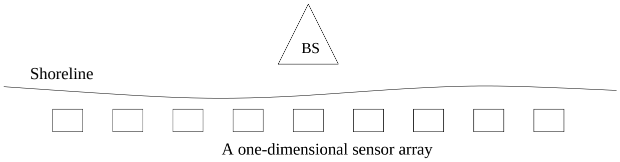

Figure 1: Topology for environmental sensor networks

The first order radio model [5] specifies that the energy consumed for transmission by a wireless device is proportional to both the message size (number of bits to be transmitted) as well as the square of the distance between the source and the destination. Hence in order to cut down on transmission energy, each device should transmit only over short distances. Maximum energy savings are obtained if the data is carried hop-by-hop by each sensor node. This technique also enables us to combine data aggregation with compression to achieve optimal energy conservation.

Placed in this topology, each sensor node will have two immediate neighbors. A node depends on these neighbors for relaying its messages towards the required destination; $i.e.$, each sensor node acts as a repeater for its neighbor, receiving data and helping it to propagate further in the network.

5 SEKEN

In this section, we describe SEKEN (Secure and Efficient Key Exchange for sensor Networks) which secures key exchange between two neighboring sensor nodes with minimal resource consumption.

5.1 Assumptions

Before describing the protocol, let us identify the assumptions underlying our model. We assume that the radio model is symmetric [8]: $i.e.$, 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 more resources than a regular sensor node. Specifically, since the base would be housed on-shore, it can run on utility electricity and use powerful computers with more memory and processing power [16]. The base station can keep a record of the keys it shares with each of the sensor nodes and can use these keys to send confidential messages to individual nodes. Because of readily available electric power, the base station can also make long range radio transmissions to reach a node anywhere within the sensor network. However, in order for messages to travel 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. First, we assume that the sensor nodes are created with a unique Device Identifier (DId), which is known only by that particular sensor node. The DId of all the nodes has to be manually programmed into the base station and each DId acts as an initial shared secret between that device and the base station. The DId is used only 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 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 obviates the need for a reliable, omnipresent Certification Authority (CA).

5.2 Notation

We will use the following notation to illustrate different primitives 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 DIds in the sense that the former is only a temporary tag assigned by the base station for a particular network topology, while the latter is a more permanent identifier for the device.
- N_1, N_A, $etc.$ are examples of a nonce (a random bit string), and TS is the current timestamp. These help provide protection against replay attacks.
- $MAC(K, C)$ is a message authentication code computed over a counter C using key K.

5.3 The Protocol

We define three basic message types used to initiate the SEKEN protocol. A node wishing to join the network sends a "join-network" message. After successfully authenticating with the base station, a node authenticates with its neighbors using an "authenticate-me" message. Finally, a node that 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 identity field in the message header, and this prompts a suitable action at the appropriate network devices.

We divide the protocol in two major steps, the key setup phase and the mutual authentication phase.

5.3.1 Key setup phase

The node closest to the base station initiates the key setup phase by issuing the "join-network" message. It retrieves its DId from memory, appends to it the current timestamp, TS, and encrypts the entire packet with the base station's public key. It waits a random amount of time before transmitting this packet to the base station.

$$A \rightarrow BS : E_{PUB}(DId_A, TS) \tag{1}$$

The node also calculates the local copy of the key, K_A, it will be sharing with the base station by computing K_A = MAC(DId, TS). The base station decrypts the received message with its corresponding private key and searches its database for a device with the same identifier. On confirming the validity of the device and realizing that this is the first node to request association, the base station computes its own copy of the proposed key K_A=MAC(DId,TS). It uses this key to send the following encrypted information to node A: the node ID, ID_A, and a counter, C_A, initialized to some random value. The node ID is a unique temporary device identification assigned to a device for the current 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 [6]. The counter value is used in a MAC to generate a session key between this node and its potential neighbor, and is incremented at both the base station as well as at the sensor node after each successful key authentication between the node and its neighbor.

$$BS \rightarrow A : E_{K_A}(C_A, ID_A) \tag{2}$$

The first node that manages to complete the key setup procedure with the base station acts as a gateway for all the other nodes in the network, helping them to communicate with the outside world. The next sensor node (assume $B1$) wishing to join the network performs the same sequence of steps. It starts by appending its current timestamp to its DId, encrypting the result with the public key of the base station, and computing its local copy of K_{B1}. The encrypted packet is then broadcast, and the closest neighbor in line toward the base station appends its own encrypted

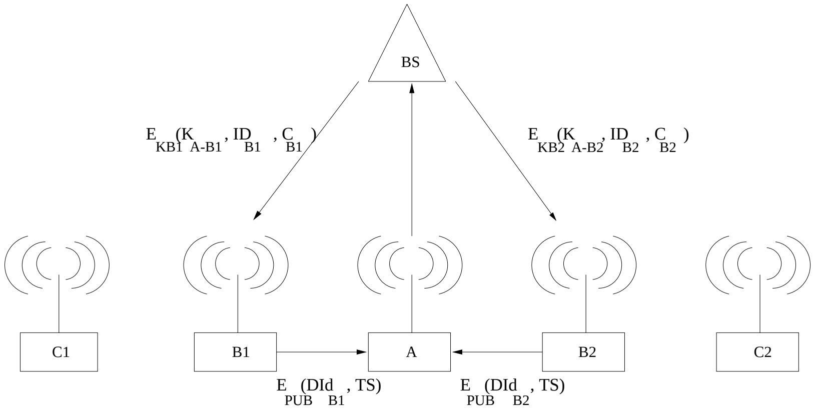

Figure 2: Message exchange for SEKEN protocol. Nodes B1 and B2 are setting up a secure key with the base station. The gateway node A has already received $E_{K_A}(C_A, ID_A)$ from the base station, BS.

ID to the message (to help the base station estimate the approximate location of the new node), and the message is finally transported to the base station.

$$B1 \rightarrow A : E_{PUB}(DId_{B1}, TS) \tag{3}$$

$$A \rightarrow BS : E_{PUB}(DId_{B1}, TS), E_{K_A}(ID_A) \tag{4}$$

The base station performs the routine validity checks on the node, computes the key proposed by the sensor node, and then sends the node the information it needs 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} = $MAC(K_A, C_A)$ it shares with its neighbor A.

$$BS \rightarrow B1 : E_{K_{B1}}(K_{A-B1}, ID_{B1}, C_{B1}) \tag{5}$$

5.3.2 Key Authentication

Once this information is available at node $B1$, it attempts to authenticate its neighbor A using a challenge-response scheme. Node $B1$ generates a nonce, N_1, encrypts it with the key K_{A-B1}, and transmits it to its neighboring node A. Node A, on receiving an "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 newly 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}.

$$B1 \rightarrow A : E_{K_{A-B1}}(N_A) \tag{6}$$

$$A \rightarrow B1 : E_{K_{A-B1}}(N_A, N_B) \tag{7}$$

$$B1 \rightarrow A : E_{K_{A-B1}}(N_B) \tag{8}$$

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 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 $B1$.

5.3.3 Node Addition and Removal

Suppose that a network node wants to attach itself to this chain of sensor nodes by appearing in between two existing nodes. For example, node $T1$ joins the network between nodes A and $B1$ in figure 3. It issues a "join-network" message to which node A appends its own ID and forwards it to the base station just like for any other node. The base station maintains the topological 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 this new node and its neighbors. After receiving this information, the new node authenticates itself to each of its two neighbors as explained in steps (6)–(8).

Now suppose that 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 an "update-neighbor" message and again the sequence of steps outlined in (3)–(4) are followed. 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

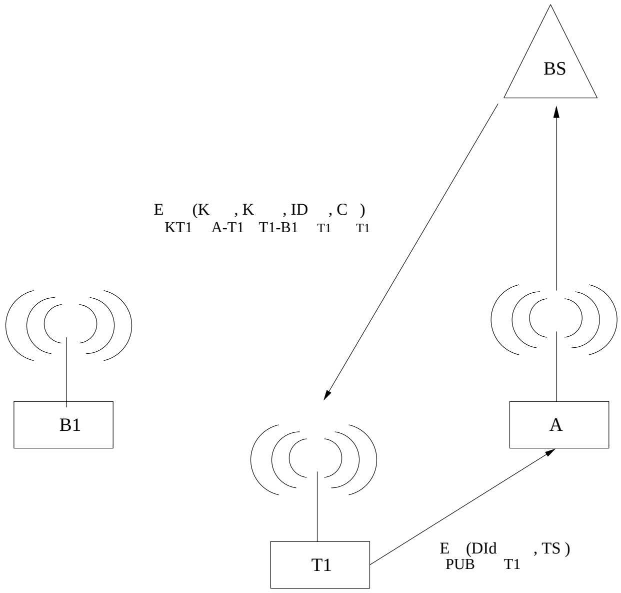

Figure 3: Member addition in the SEKEN protocol. Node T1 attempts to join the network comprising of nodes A, B1 and others. The usage of an incrementing counter during the key setup process enables a sensor node to share secret keys with a large set of neighbors.

to $B1$. Node $B1$ then authenticates itself to node A using the procedure outlined above.

6 Comparative Analysis and Results

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

One of the simplest key setup protocols is predeployment of keys before the sensor nodes are put into active operation [1]. Once deployed, the nodes already share the cryptographic keys, and therefore the protocol only requires node authentication using a challenge-response scheme. (Same as steps (6)–(8) stated for SEKEN above). Although this protocol has a minimum overhead, it raises scability and security concerns especially for changing mission configurations. $e.g.$,, 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. New secure methods would need to be developed to perform these operations for sensor nodes already deployed in the field.

We also compare the SEKEN protocol against a Kerberos key exchange set up between two parties. In [1], a number of key setup protocols were analyzed for the environment of sensor networks, and Kerberos was found to be the most energy-efficient a after pre-deployed key mechanism. In the Kerberos protocol, each node shares a long-term pairwise key with a trusted server *a priori* [9]. We assume that the base station 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.

$$B1 \rightarrow T : B1, A, N_B \tag{9}$$

$$T \rightarrow B1 : ticket_A, E_{K_{BT}}(K, N_B, A) \tag{10}$$

$$B1 \rightarrow A : ticket_A, authenticator \tag{11}$$

$$A \rightarrow B1 : E_K(T_B) \tag{12}$$

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 base station T and node A or node $B1$ respectively, K is the session-key chosen by T, and T_B is a timestamp from B's local clock.

We use the first-order radio model to compute the energy costs associated with transmission and reception of packets in the sensor network. To transmit k bits of data to another node d distance apart, the source 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

$$E_{Rx}(k) = E_{RX-elec}(k) = E_{elec} * k$$

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 used to amplify the transmitted signal.

We assume that all symmetric key sizes are 64 bits. Though our simulation results can easily be extended to larger key sizes, we believe that this key length is enough to provide sufficient protection against a brute-force attack over the lifetime 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. Cryptix Crypto 3.0 (a clean room implementation of Sun JCE) was used to implement the cryptographic primitives, and the security-related handshake messages were exchanged using socket communication. For the context of discussion in this paper, we compute and compare only the energy cost of communication among the different key setup schemes. The energy

Table 1: Energy consumption for larger sensor networks

Nodes	Pre-deployed Keys	SEKEN	Kerberos
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

Table 2: Energy consumption for member addition

Distance from gateway	SEKEN	Kerberos
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

consumption is the sum of the energy consumed by the sensor nodes involved in the key setup and subsequent authentication process ($i.e.$, transmission and reception costs for the base station are being ignored). The 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/bit while the cost of reception is 50 nJ/ bit under the radio model we have described above.

For pre-deployed key mechanism [1], each sensor node has only to authenticate its key with its neighbors. Application of SEKEN to such a sensor network has already been explained above. To maintain consistency in 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 simulation confirm that pre-deployment is indeed the most efficient method of authenticating two neighboring sensor nodes. The protocol, however, 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 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. Although there is not a huge difference between the energy consumption in SEKEN and Kerberos, a Kerberos requirement that the server shares a long-term explicit master key with every sensor node is a potential drawback, especially for large networks. No such assumption is made in the SEKEN protocol, in which all such keys are set up during the execution of the protocol itself. SEKEN just requires all potential network nodes to share a one-time secret with the base and to be 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).

As pointed out in earlier sections, scalability is another concern for large sensor networks. For pre-deployed key mechanism, imagine the hassle of correctly placing about a thousand sensor nodes, knowing that the network would not function correctly if a single one is misplaced! SEKEN offers cost-efficient scalability when compared to Kerberos (costs for both of which increase more than linearly because individual key setup requests have to travel from the farthest new node to the base station).

Table 2 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 constant 0.2816 mJ. 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 the way upto the base station). Once again, SEKEN shows superior performance characteristics over Kerberos.

Computational cost analysis forms the basis of our continuing work, yet our preliminary results confirm that despite performing one public-key encryption during the lifetime of the sensor node *(public key encryption, performed by the senor node once during its lifetime, requires about 20 times fewer operations when compared to the corresponding decryption with the private key performed by the base station)*, SEKEN maintains its overall advantage over other protocols because local computation of data is much more efficient compared to radio transmission due to the reasons outlined in section 2.4.

Tables 1–2 present the simulation results carried out in a "laboratory-safe" environment. In a real network we have to deal with a multitude of loss sources, many of which are random in nature. For our simulations, we assume that there are no message losses and hence no retransmissions. This might not be true, especially in wireless networks where the channel is highly susceptible to link losses and other sources of interference. However, we reason that in a real network both Kerberos and pre-deployed key mechanisms would be subject to the same network degradation and their performances will also suffer.

It has to be remembered that security is only an auxillary operation for sensor networks and therefore should not be a burden on system resources. 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 15.39 kJ [2]. A 200 node network will have a combined wattage of 3.08 MJ of which SEKEN consumes only 1.106 J for key setup. This is encouraging, and suggests that the final cost of implementing security can remain bounded through efficient protocols.

7 Conclusions

SEKEN provides scalable, power-efficient secure communication for a network of wireless sensors by reducing the explicit exchange of messages over the wireless medium and substituting enhanced local processing at the host node. The protocol allows self-configurable operations in an autonomous network with minimum user intervention, which is ideal for a high risk wireless sensor network with changing topology. The protocol enables each sensor node to share two types of keys:

1. A master key shared with the base station for confidential exchange of messages.

2. An explicit key between individual neighboring nodes, allowing secure information exchange.

Depending upon the level of assurance required, suitable mechanisms can also be put in place to periodically refresh these keys in order to safeguard against brute-force attacks.

8 Future Work

The development of a key management protocol is only the first step in developing a suite of protocols for securing the wireless sensor networks of tomorrow. Our future work will involve identifying cryptographic primitives that are most efficient in computational resource utilization. We earlier pointed out that it would be a waste of resources to blindly encrypt all data exchanged between the various communicating nodes. We intend to identify messages that will be critical for the functionality of the network. For instance, we would like to implement routing updates and a few other critical network management messages securely because they are one source of potential attack. The short length and only an occasional exchange of these messages prompt us to continue thinking along these lines. Finally we will build a working prototype of these protocols on the sensor nodes being developed for this project.

References

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

[2] Duracell. URL. http://www.duracell.com.

[3] EPA. Primary Sources of Pollution. URL: http://www.epa.gov/ost/beaches/2000/primary.html.

[4] L. Eschenauer and V.D. Gilgor. A Key-Management Scheme for Distributed Sensor Networks. In *Ninth ACM Conference on Computer and Communications Security*, November 2002.

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

[6] R. Jain, A. Puri, and R. Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks. In *IEEE Personal Communications*, February 2001.

[7] A. Khalili, J. Katz, and W.A. Arbaugh. Toward Secure Key Distribution in Truly Ad-Hoc Networks. In *IEEE Workshop on Security and Assurance in Ad-Hoc Networks*, 2003.

[8] S. Lindsey and C.S. Raghavendra. PEGASIS: Power Efficient Gathering in Sensor Information Systems. In *IEEE Aerospace Conference*, March 2002.

[9] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[10] A. Perrig. Private communication, July 2001.

[11] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.D. Tygar. SPINS: Security Protocols for Sensor Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*, 2001.

[12] G.J. Pottie and W.J. Kaiser. Wireless Integrated Network Sensors. In *Communication of ACM*, 2000.

[13] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

[14] L. Schwiebert, S.K.S.Gupta, and J. Weinmann. Research Challenges in Wireless Networks of Biomedical Sensors. In *ACM/IEEE International Conference on Mobile Computing and Networking*, 2001.

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

[16] D.C. Steere, A. Baptista, C. Pu, and J. Walpole. Research Challenges in Environmental Observation and Forecasting Systems. In *International Conference on Mobile Computing and Networking*, 2000.

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