A Low-Energy Key Management Protocol for Wireless Sensor Networks

Gaurav Jolly, Mustafa C. Kuşçu, Pallavi Kokate, and Mohamed Younis

Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, Maryland 21250
{jolly1, kuscu1, pallavi2, younis}@umbc.edu

[*]**Abstract.** *Sensor networks have recently gained popularity for a wide spectrum of applications. When performing monitoring tasks in hostile environments, security requirements become critically important. Moreover, limited energy, computing, and memory resources prohibit the use of complex security mechanisms within sensor networks. Therefore, well balancing between security level and the associated energy consumption overhead becomes a must. In this paper, we concentrate on the key management aspect of the security functionality. Key management is essential for any cryptographic security system. We present an energy-aware approach for managing the cryptographic keys in a clustered sensor network. Shared symmetric keys are pre-deployed into the sensors and gateways (the cluster heads), requiring each sensor node to store only two secret keys. Separate protocols handle network bootstrapping, sensor addition/revocation, and key renewals. The protocols do not trust individual sensors, and partially trust the gateways. Eviction of compromised sensors and gateways are supported, although yet we do not present an intrusion detection mechanism. Simulation shows that the energy consumption overhead for the key management is remarkably low and reports an order-of-magnitude energy saving.*

Keywords: Key management, wireless sensor networks, security protocols.

1 INTRODUCTION

Many compelling applications like distributed information gathering and distributed micro sensing in radiology, military, and manufacturing drive the research in sensor networks. Typically, sensor networks comprise of a large set of distributed low power sensors scattered over the area to be monitored. The sensors have the ability to gather data, and process and forward it to a central node for further processing.

Contemporary wireless sensors have limited battery, computation, and memory capacity. Such resource-constrained environment has motivated extensive research that address energy efficient and energy aware hardware and software design issues [1][2]. Most of the research effort has concentrated on energy efficient communication protocols [3][4][5]. The comparative progress in making these networks secure has been insignificant. This is despite the fact that in certain applications of sensor networks, like military applications, security is the most important criteria.

The energy-constrained nature of the sensor networks makes the task of incorporating security, a challenging problem. Most of the well-known security mechanisms introduce significant overhead and requires a lot of computation and communication resources. Since the design of security protocols for sensor networks should be geared towards resource conservation, the level of security versus the consumption of energy, computation, and memory resources constitute a design trade-off. Generally security protocols would provide the network with two-party encryption, authentication, and key management. In this paper we only focus on the design of a low energy key management scheme for a sensor network. However, we consider the security of key management and use encryption when distributing keys on radio link.

The organization of the rest of this paper is as follows. Section 2 describes the sensor network architecture that we are considering. In section 3, we present a hierarchical key management scheme that consists of protocols for key distribution, initialization, sensor addition, revocation (of sensor and gateway keys) and periodic key renewals for our sensor network, and also state the assumptions of our model. In section 4, we describe our simulation environment and experimental results. Related work on sensor network security and key management is summarized in section 5. Finally section 6 concludes the paper and outlines our future research plan.

2 SENSOR NETWORK ARCHITECTURE

We adopt the sensor network model proposed by Younis, et al. [3][6]. In this model, a sensor network consists of a large number of sensors distributed over an area of interest. There is at least one command node in charge of the network's mission. The model also introduces few super-nodes, called gateways, in addition to the sensor nodes. The gateways have considerably high energy resources compared to sensor nodes and are equipped with high performance processors and more memory. These gateways perform network management functions in a centralized

[*] Mustafa C. Kuşçu is the contact author. He can be reach by phone at (410) 455-3968, by FAX at (410) 455-3969 and by e-mail as kuscu1@umbc.edu. The authors would like Dr. Alan Sherman and William Byrd for their valuable comments.

fashion. As shown in the Figure 1, the gateways partition the sensors into distinct clusters, using some existing clustering algorithm, e.g. [7]. Each cluster is composed of a gateway and a separate set of sensor nodes. The operation within a cluster is independent from each other. Sensors of each cluster gather information and transmit it to the gateway of their cluster. The gateway fuses the data from the different sensors, processes the information pertaining to the required mission, and sends it to the command nodes via long-haul transmission.

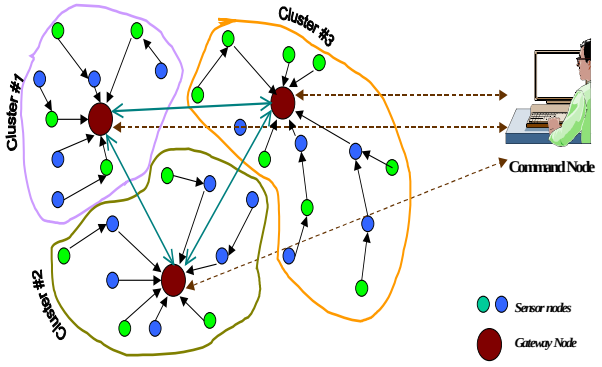

Figure 1. Multi-gateway, clustered sensor network.

Each *tier* of the network possesses different architectural capabilities. The command node can be assumed to pose no restrictions in terms of computation and storage. However, the communications channel between the command node and the network is an expensive resource to use, because its bandwidth is valuable and should not be unnecessarily flooded, and also substantial amount of energy is necessary to transmit over the long-haul channel. The gateways are assumed to have sufficient energy resources to carry out this transmission. Conversely, sensors are extremely constrained in energy. *MICA* sensor architecture is a good candidate for our network. MICA architecture is quite restrictive: uses TinyOS operating system with 128K flash memory, 4K RAM, and Atmel ATmega processor (8-bit, 4MHz), and runs with 2 AA batteries [9].

Energy-aware operation of the network is mainly based on switching unused sensors to sleep mode. The sensors have three distinct modes of operation: sleep, idle and active. Active and idle modes are almost identical and consume similar amount of energy. In the active mode the sensor is transmitting (receiving) data or sensing the environment. While in the idle mode the node is not doing anything but the circuitry is on. The computation part of the sensor is always on in active and idle modes. In the sleep mode the circuitry is switched off and hence the sensors consume considerably less energy. Therefore in order to conserve energy, the nodes make a transition to sleep mode, when idle. It is the responsibility of the gateways to schedule each sensor. Scheduling means that the gateway tells each sensor to either transmit (receive) or sense the environment. In our architecture the sensors employ Time Division Multiple Access (TDMA) based media access control (MAC) layer, wherein the gateway assigns distinct slots to the sensors. The gateway instructs the sensors when to make a transition from sleep to idle mode and vice versa. The sensors transmit in the slots assigned to them and thereafter, in order to conserve energy, make a transition to sleep mode.

Given the limited environment, providing security services for the sensor network is not a trivial task. Secure communications between a gateway and its sensors is a necessary part of the overall security of the network. In the next section, we focus on the key management aspects for a sensor network used in military reconnaissance setup.

3 KEY MANAGEMENT

A key management procedure is an essential constituent of network security. Symmetric key systems require keys to be protected. Insecure environments like those sensor networks will be used in make this even more important. Moreover, sensor networks have energy and computational constraints; therefore it is necessary to maintain a balanced security level with respect to those constraints. In this section we propose a key management scheme for sensor networks, with the objective of minimizing the computation, communications and storage overhead by the key management.

In military reconnaissance scenarios, sensors and gateways are expected to be deployed across the border in the enemy territory. The sensors will monitor the movements of the enemy and report the gathered information to the gateway that in turn will process that data and forward it to the command node. The command node will be deployed in the friendly environment on the other side of the border. Therefore we can assume that the command nodes are secure while the gateways and sensors can be compromised by the adversary.

3.1 Assumptions

We make the following assumptions regarding the security of the sensor networks.

- *Command Node*: The command node is assumed to be secure and is trusted by all the nodes in the sensor network. Moreover the command node is capable of detecting intrusion and will thereby instruct the other gateways to remove the compromised gateway from the cluster.

- *Gateways:* We assume that the gateways can be captured by the adversary. Each gateway can directly communicate with every other gateway in the network. Moreover, we assume that broadcast communication among gateways is available. We also assume that there is secure group communications among the gateways. The clustering algorithm, e.g. [7], can be easily modified to include the step of the secure communications establishment.

- *Sensors*: We have not made any trust assumptions on sensors. They can be captured, and compromised. Namely, the adversary can read all the information

from the sensor's memory, including the keys, and even program the sensor to inject malicious messages.

- *Group Key Establishment*: In order to establish secure clustering and communications among gateways, the gateways need to agree on a group key. A group key establishment protocol can be picked from the survey by Carman et al [10], for instance, A.GDH-3. This protocol requires a network root, which can be the command node in our scenario. At the end of the protocol, each node has its own group key K_{group}. We do not provide the details about the key establishment here, because we do not consider the complexities at the gateway level.

We have observed that end-to-end communications between sensors is not common. Alternatively, all traffic flows within a cluster is over the gateway of the cluster. Therefore, the high overhead of establishing secure sessions between individual sensors can be avoided. Moreover, we assume that each sensor uses direct communications with the gateway during key management operations.

3.2 Proposed Approach

In our approach the nodes use a symmetric key mechanism; therefore each node should store the keys it shares with its peers and the parent nodes at the higher layers of the network hierarchy. Since the sensors are memory constrained and are susceptible to attacks by the adversary, they should be assigned the minimum number of keys. Assigning minimum number of keys saves memory. In case a node is compromised it also helps to reduce the impact of the damage since less number of keys would be revealed to the adversary.

Gateways have sufficient memory resources and hence can be assigned large number of keys, however they cannot be trusted and therefore cannot be programmed with all the keys. Assigning all the keys to the gateway will compromise the entire network in case one gateway is captured. Command nodes have no restrictions as they are assumed to be secure, and have sufficient memory. Therefore the command node can store all the keys in the network.

3.2.1 Distribution of Keys

Keys are programmed into the memory of the sensors just before they are being deployed. Sensor and gateway keys are stored in the flash RAM and hence can be deleted when required. The command node stores all the keys it shares with the sensors and the gateways. Therefore the number of keys stored by the command node is equal to $G + S$, where G is the number of gateways and S is the number of sensors. The gateway stores the keys it shares with the sensors in its cluster and the key it shares with the command node. It also shares one key with exactly one other gateway in the network. The keys are pre-deployed therefore there is no key transmission overhead at the sensor side, during initialization (bootstrapping). Each sensor stores two pre-deployed secret keys; one that it shares with the command node, and one that it shares with the gateway.

We have assumed that the command node could detect the sensors and gateways that were compromised by the adversary. The key shared with command node key is used during revocation of a compromised gateway. If the same keys are used for an extended duration of time, the adversary may be able to deduce the keys by traffic analysis techniques. Therefore, the command node periodically renews the gateway-sensor keys and sends them to the gateways. The gateways forward this information to the sensors in their clusters.

The protocols in the following subsections use the terminology in Table 1.

Table 1: Terminology used in the key management protocols.

Notation	Description
CN	the command node
G_i	gateway i
G_{all}	all gateways
G_{head}	head gateway chosen by the command node, used for revocation
GID	gateway id
S	sensor node
NodeID	node identifier
Nonce	random nonce value
K_{SG}	key shared between a sensor and a gateway
K_{group}	group key agreed by gateways
K_{ij}	key shared between gateway i and j
K_{gcn}	key shared between a gateway and the command node
K_{scn}	key shared between a sensor and the command node
$E(K, \ldots)$	encryption function with key K
$\|$	concatenation operator

3.2.2 Initialization Phase

At the time of deployment, each gateway is assigned S/G keys where S is the number of sensors and G is the number of gateways. The key assignment is done randomly since we assume that sensors and the gateways are randomly distributed and are not aware of the topology prior to the deployment.

Once deployed, the gateways form clusters using some existing cluster formation algorithm and thereafter acquire keys of the sensors in their cluster from other gateways using our key exchange protocol. After exchanging the keys, each gateway deletes the keys of the sensors that are not in its cluster. This is essential since if a gateway is captured then only the keys of its cluster become available to the adversary.

Sensors also store the ID of the gateway that contains the key for that sensor prior to deployment. The sensor includes this ID in the "hello" message it broadcasts after deployment. If this ID is not included in the message, each gateway will have to send the ID's of all sensors to all the gateways, in order to acquire the keys for that sensor. Further each gateway will have to search the list of sensors to find if the requested sensor ID is stored in its memory. This requires lot of computation and including the gateway's ID in the message from the sensor to the gateway reduces this overhead. The sensor nodes include some information regarding the location and the energy level of the sensor encrypted by the key it shares with the gateway. This information is used by the gateway during the route formation process. The protocol for the initialization phase is described as follows:

$S \rightarrow G$: $GID \parallel NodeID \parallel E(K_{sg}, Nonce \parallel Data)$ (i1)

...clustering... (i2)

$G_i \rightarrow G_{all}$: $GID \parallel E(K_{group}, Nonce \parallel NodeID_1 \parallel \ldots NodeID_i)$ (i3)

$G_i \leftarrow G_j$: $E(K_{ij}, Nonce \parallel (K_{SG,1}, NodeID_1) \parallel \ldots \parallel (K_{SG,i}, NodeID_i))$ (i4)

$S \leftarrow G$: $GID \parallel E(K_{SG}, Nonce \parallel GID)$ (i5)

3.2.3 Addition Phase

If a new sensor is added to the network, it will contain two keys: one is shared with the command node and the other will be shared with the gateway of its cluster. But the gateways have already been deployed and are not aware of the new sensor IDs as well as the keys of these sensors. Since the sensors are deployed at random, they cannot be pre-assigned to a cluster.

The command node partitions the list of newly deployed sensors into distinct sets of equal size. These lists contain the sensor IDs and the corresponding keys they are supposed to share with the gateway. Then the command node transmits messages containing one of these lists to some gateway at random. Once deployed each new sensor node broadcasts a 'hello' message to the gateways. On receiving the messages the gateways arbitrate amongst themselves and assign this sensor to one of the gateways using the clustering algorithm. Rest of the exchanged messages is same as those required for the communication during the initialization phase.

This approach minimizes the involvement of the command node in the addition phase. This is important since the channel between the command node and the gateways may not always be available and the communication between the gateway and the command node is expensive. The sensor addition protocol is descried as follows:

$CN \rightarrow G$: $E(K_{gcn}, (K_{SG,1}, NodeID_1) \parallel \ldots \parallel (K_{SG,i}, NodeID_i))$ (a1)

$S \rightarrow G$: $GID \parallel NodeID \parallel E(K_{SG}, Nonce \parallel Data)$ (a2)

...clustering... (a3)

$G_i \rightarrow G_{all}$: $GID \parallel E(K_{group}, Nonce \parallel NodeID_1 \parallel \ldots \parallel NodeID_i)$ (a4)

$G_i \leftarrow G_j$: $E(K_{ij}, Nonce \parallel (K_{SG,1}, NodeID_1) \parallel \ldots \parallel (K_{SG,i}, NodeID_i))$ (a5)

$S \leftarrow G$: $GID \parallel E(K_{SG}, Nonce \parallel GID)$ (a6)

3.2.4 Revocation Phase

Revocation procedures are involved after detecting compromised nodes. Presently, we assume that there is an intrusion detection mechanism that informs the command node about the compromised nodes. From the key management point of view, supporting revocation operations is sufficient.[1] If a sensor node is compromised, it will be evicted from the cluster by the gateway. The gateway will not honor further messages from the evicted sensor. Also other sensors in the cluster will ignore that sensor because the centralized medium arbitration will ignore it.[2]

In the case of a gateway compromise, the command node will elect an uncompromised gateway as a *head*, and instruct it to remove the compromised gateway from the gateway group and distribute its sensors among the remaining gateways. The group key management protocol, which we had assumed to exist at the gateway level, will make sure that the evicted gateway is not honored. It should be noted that the gateways distribute the sensors among themselves using the clustering algorithm. Thereafter the head gateway distributes the keys of the sensors to their respective gateways.

Each gateway then independently contacts the sensors assigned to its cluster. It transmits the message containing its own ID and also forwards the message from the

[1] Design of a lightweight intrusion detection mechanism is part of our future work.

[2] In particular, this is an issue related to the routing function and the TDMA-based medium access control protocol, which re-schedules transmission and reception time slots at each period, i.e., a TDMA frame. We prefer centralized methods here and in other protocols as we try to minimize the workload on sensors.

command node to the sensor encrypted with the key shared between the sensor and the command node.

On successfully decrypting the message the sensor knows that the message originated from the command node and hence discards messages from the evicted gateway and tunes its receiver to listen to the newly assigned gateway. It also discards its previous key and uses the new key for future communication.

$CN \rightarrow G_{head}$: $E(K_{G,CN}, (K_{SG,1}, NodeID_1) \| \ldots \| (K_{SG,I}, NodeID_i) \| data)$ (r1)

$E(K_{1,KSCN}, Nonce \| G_i \| K_{SG}) \| .. \| E(K_{N,KSCN}, Nonce \| G_i \| K_{SG}))$ (r2)

...clustering... (r3)

$G_{head} \rightarrow G_i$: $E(K_{head,i}, Nonce \| NodeID_1 \| \ldots NodeID_i \| E(K_{S,CN}, data)$ (r4)

$G_i \rightarrow S$: $E(K_{KSCN}, Nonce \| G_i \| K_{SG})$ (r5)

3.2.5 Key Renewals

A single key encrypts significant amount of data. If an adversary has information about the sent data and its format, it is likely that he can conduct a known plaintext attack. A remedy to this threat is to renew the encryption keys periodically [11]. SPINS [8] does not provide a mechanism against this source of threat, nevertheless, the sensor battery drains quickly in that system because of small battery capacity and high power consumption by the always-on radio. This requires the gateways to re-generate keys at some period, depending on the amount of traffic, the strength of underlying cryptographic primitives, and the overhead incurred at the gateways. The new keys will be transmitted in a similar fashion to the key update phase of the gateway revocation protocol defined in Section 3.2.4.

4 SIMULATION RESULTS

We have extended a Visual C++ based sensor network simulator by Younis et al. [3][6] to measure energy consumption overhead due to the key management protocols. Our key management approach does not call for any sensor to generate keys, or to perform any extensive computation associated with key management. There exist a few encryption/decryption operations required during key management, nevertheless their complexity depends on the algorithm, and we believe there exist very low energy consuming algorithms that provide sufficient level of security [8]. Therefore, major energy consumption is due to bit transmissions and receptions during communications.

In the simulation, our primary objective is to observe the behavior of the network, which is difficult to predict analytically. In this context, our metrics are the energy consumption per node, within overall network, and the overhead of security in terms of energy consumption.

The simulation results can be seen in figures 2-5. The average energy consumption overhead is very low, in the order of micro Joules. This overhead consists of communications costs at the sensor due to transmission of authentication information during bootstrapping of the network. For instance, each sensor in a 1000-node network with 7 gateways consumes 60 μJ on the average (see Figure 2), which is a very feasible cost when considering the energy in batteries can be high. The density of gateways can be increased to further reduce the energy consumption by sensors. This is seen in Figure 5.

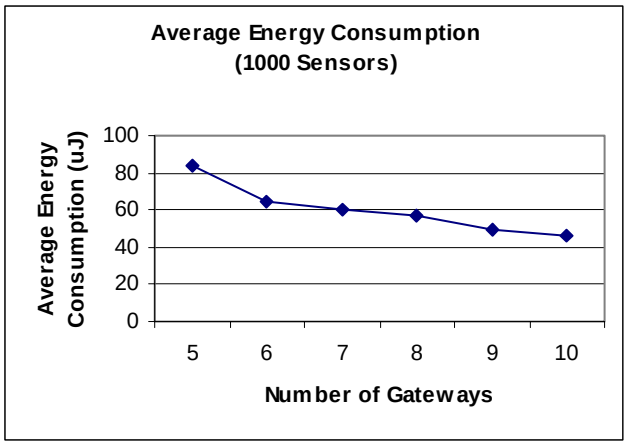

Figure 2. Average consumed energy by sensors versus number of gateways. The number of sensors is fixed.

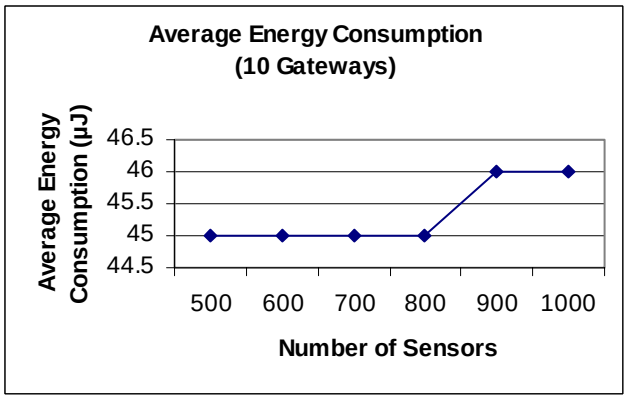

Figure 3. The effect of increasing the number of sensors to average energy consumed by sensors.

Revocations that have been defined in Section 3.2.4 are result of compromise detection. After revocation of a gateway, all nodes of the revoked gateway are assigned new gateways and sent new keys. Therefore the energy consumption of revocation step is high. This high amount of energy is only consumed when a gateway is compromised, and thus the frequency of gateway compromises affects the lifetime of the network.

Key renewals require significant amount of energy. This is similar to the case after revocation of every gateway, in terms of sensor energy consumption. That means, all sensors in the network are assigned their new keys. Key renewals can be done periodically. It would be a good choice to develop a key renewal strategy to keep the number of renewals at a minimum. An example strategy could be to keep track of the amount of traffic per sensor,

and renew the keys of those sensors whose traffic has passed a threshold. This will eliminate key renewals, i.e. provides extra energy savings, for unused sensors.

The next section surveys related research and compares them with our approach.

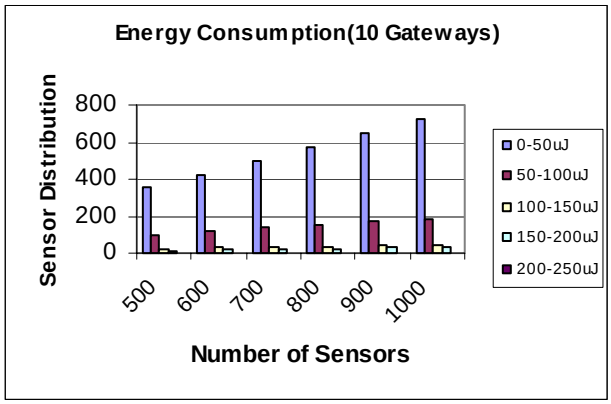

Figure 4. Distribution of sensor energy consumption, varying the number of total sensors.

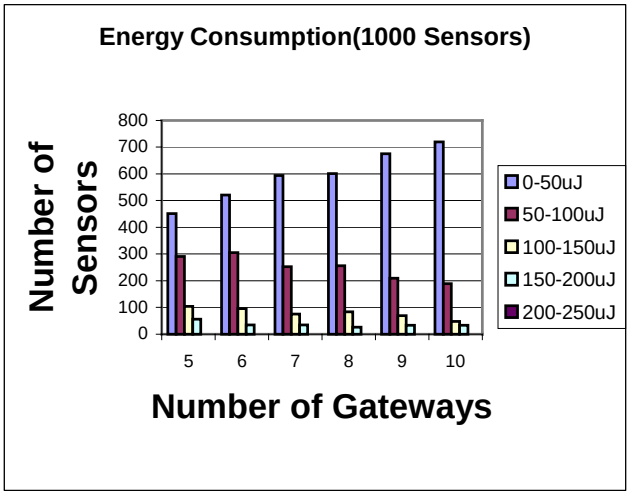

Figure 5. Distribution of sensor energy consumption, varying the number of gateways.

5 RELATED WORK

During the past few years, sensor networks have attracted wide attention of researchers around the world [3][6]. The architecture and design of sensor networks have progressed significantly [12]. Numerous strategies aim to reduce energy consumption at the sensor nodes [3]. Communication hardware has also received attention [13][4][5].

Recently, security has become a topic of interest in sensor networks research. Network Associates Laboratories (NAI Labs) has conducted a detailed survey on security mechanisms for sensor networks [10]. Their work evaluates application of existing security technologies on sensor networks, in terms of energy consumption, and memory requirements. They observed that *pre-deployed keying*, which requires deployment of keys to the sensors prior to their use, is very energy efficient; however, it poses inflexibility to configuration changes and requires storage of high number of keys. One way to do pre-deployed keying is to use a *pre-deployed group keying* approach. NAI researchers noted that this approach had significant problems. First, compromise of a single node results in the compromise of the entire network, because all nodes use the same key for encryption and authentication functions. This is a significant security problem. Second, pre-deployment reduces the flexibility to dynamicity of sensor nodes. *Node-specific pre-deployment* is another pre-deployment method for keys. In this method, each node receives the set of keys that it is going to use. For large networks, the number of required keys per node and the storage requirements at each node can be too high. Therefore this method cannot be scalable in the number of sensor nodes. Alternatively, the shared authentication key can be pre-deployed network-wide, whereas encryption keys are deployed in a node-specific fashion, which requires less storage than the node-specific pre-deployment, and provides more security than the network-wide pre-deployment.

Instead of pre-deployment, a trusted third party like a *Key Distribution Center* (KDC) can be used. An example adaptation of Kerberos, a KDC-based protocol family, was provided in the NAI Labs report. It is suitable for sensor environments with available global clock. However its energy consumption is huge for a sensor node when compared to pre-deployed keying. Because for every secure session, a sequence of request and response message transmissions are required. Therefore, using a traditional KDC scheme would incur a significant communications cost, and it remains beyond the feasibility limits. Figure 6 is a good example for this. Given the same network and the same communications assumptions as in pre-deployed keying, when Kerberos is used for key distribution instead of key pre-deployment, the majority of the sensors dissipate energy in the 1.5 - 3.0mJ range. This is mainly due to the long messages involved per every established secure session. The energy range has been 0-50μJ for pre-deployed keying that we had used.

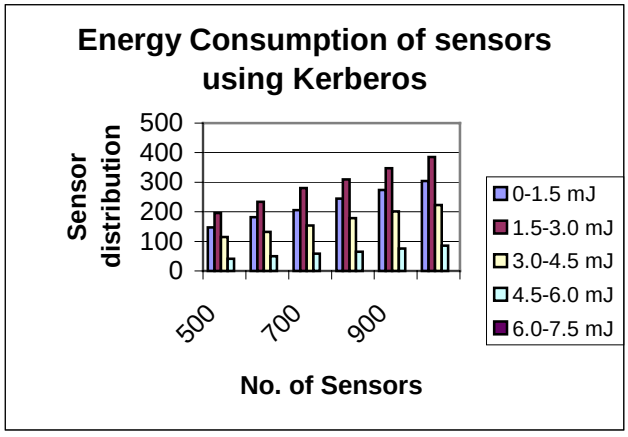

Figure 6. Distribution of sensor energy consumption using Kerberos-based key distribution

A new protocol proposed in the report is the Identity-Based Symmetric Keying (IBSK), which is very similar to our approach. This protocol requires the shared keys to be pre-deployed in sensor nodes. The node-specific keys are different for each node. Therefore, IBSK provides less energy consumption than Kerberos protocols and more compromise protection than group keying protocols. However, the storage requirements for a sensor node are high, because the original IBSK protocol assumes that any individual sensor can communicate with another sensor in an end-to-end fashion, which requires one shared key per sensor pair.

IBSK and other work by NAI Labs in [10] assume that node-to-node communication can take place. In terms of energy conservation, we believe this assumption has a negative effect. For a wide range of applications, we assume that sensor-to-gateway traffic will be the dominant traffic pattern. Many applications can be designed in a way that makes direct sensor-to-sensor communications unnecessary. This is partially due to the fact that data gathering applications that utilize sensor networks eventually have an inherent centralized architecture. This observation was employed by Perrig et al. [8] for significantly reducing the key management overhead. Under the assumption, IBSK—without any protocol modifications—energy consumption would be order-of-magnitude less than its nominal values when node-to-node communication is allowed. Identity based symmetric keys are used in other authentication protocols, as well [14].

The key management scheme proposed by Eschenauer et al [16] relies on probabilistic key sharing among the nodes of a random graph and uses a simple shared-key discovery protocol for key distribution, revocation and node re-keying. In this scheme a key ring is provided to each sensor node prior to network deployment. Each key ring consists of randomly chosen k keys from a large pool of P keys, which is generated on-line. Since the choice of keys on key rings is random, a shared key may not exist between some pairs of nodes. Although a pair of nodes may not share a key, if a path of nodes sharing keys exists between the two nodes, the pair of nodes can use that path to exchange a key that establishes a direct link.

Elected Simple Key Distribution Center (ESKDC) protocol is a lightweight protocol and is efficient for small groups. Clique Group Diffie-Hellman Protocols, like Authenticated Group Diffie-Hellman protocol (A.GDH-3), can be more suited for secure group communications key establishment among gateways. We are using group key management at the gateway level (among the gateways and the command node), and pairwise keying among sensors. Providing hybrid keying abilities to sensors has been shown to be conserving energy [10] although it incurs additional complexity in the sensors. In our case, pairwise-only keying does not become a problem in terms of energy consumption, because the sensor network utilizes centralized gateways and no sensor-to-sensor communication is present. Moreover, the route determination and medium arbitration are performed by the centralized gateways, unlike by a purely distributed routing algorithm, where a high volume of peer traffic among sensors would be necessary. Therefore, a pairwise-only approach, which does not have the overhead of group keying, performs very well.

6 CONCLUSIONS AND FUTURE WORK

We have presented an energy conserving method to provide key management for sensor networks. The method uses pre-deployed symmetric keying. A critical observation is that sensor-to-sensor secure channel establishment is not necessary for many monitoring applications. Therefore, pre-deployed keying has become sufficient, cost-effective approach to provide a keying infrastructure for security protocols that use those keys. The cost of introducing key management to the sensor network architecture has been shown to be in the order of μ Joules per sensor. The overhead per sensor appears feasible, and none of the sensors are required to store large numbers of keys, exceeding their storage limit. We believe this is a significant improvement, when compared to the traditional key distribution protocols. Moreover, our approach supports key revocation and renewal mechanisms, as well.

Our future work will be to provide lightweight and energy-aware authentication protocols. An open problem is to integrate a lightweight intrusion detection mechanism to detect compromised nodes.

Reference

[1] J. Rabaey, et al., "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design," in Proceedings of the International Solid-State Circuits Conference, (San Francisco, CA), February 2002.

[2] S. Hollar, "COTS Dust," Master's thesis, Electrical Engineering and Computer Science Department, UC Berkeley, 2000.

[3] M. Younis, M. Youssef, and K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks," in Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), (Forth Worth, TX), October 2002.

[4] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy Aware Wireless Microsensor Networks," IEEE Signal Processing Magazine, March 2002.

[5] E. Shih, B. Calhoun, S.-H. Cho, and A. Chandrakasan, "Energy-Efficient Link Layer for Wireless Microsensor Networks," in Proceedings of the Workshop on VLSI 2001 (WVLSI '01), (Orlando, Florida), April 2001.

[6] K. Arisha, M. Youssef, and M. Younis, "Energy-Aware TDMA-Based MAC for Sensor Networks," in Proceedings of the IEEE Workshop on Integrated

Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), May 2002.

[7] G. Gupta, M. Younis, "Performance Evaluation of Load-Balanced Clustering of Wireless Sensor Networks," in the Proceedings of the 10th International Conference on Telecommunications (ICT'2003), Tahiti, Papeete – French Polynesia, February 2003 (to appear).

[8] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," Wireless Networks, vol. 8, no. 5, pp. 521-534, 2002.

[9] M. Horton, et al., "Mica: The commercialization of microsensor motes," Sensors Online Magazine, April 2002. http://www.sensorsmag.com/articles/0402/40/main.shtml.

[10] D. Carman, P. Kruus, and B. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. 00-010, NAI Labs, September 2000. http://download.nai.com/products/media/nai/zip/nailabs-report-00-010-final.zip.

[11] W. Fumy and P. Landrock, "Principles of key management," IEEE Journal of Selected Areas in Communications, vol. 11, pp. 785-793, June 1993.

[12] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Algorithm and Protocol Design for Energy-Efficient Wireless Sensor Networks," in the 7th ACM Annual International Conference on Mobile Computing and Networking 2001), July 2001.

[13] S.-H. Cho and A. Chandrakasan, "Energy Efficient Protocols for Low Duty Cycle Wireless Microsensor Networks," in International Conference on Acoustics, Speech, and Signal Processing 2001, May 2001.

[14] M. Tatebayashi, N. Matsuzaki, and D. Newman, "Key Distribution Protocol for Digital Mobile Communications Systems," in Advances in Cryptology: Proceedings of CRYPTO'89 (G. Brassard, ed.), vol. LNCS 435, pp. 324-334, Springer-Verlag, 1989.

[15] A. Perrig, et al.,, "SPINS: Security Protocols for Sensor Networks," in Seventh Annual International Conference on Mobile Computing and Networks (MobiCOM 2001), (Rome, Italy), 2001.

[16] L. Eschenauer and V. D. Gligor, "A Key Management Scheme for Distributed Sensor Networks," in 9th ACM Conference on Computer and Communications Security—CCS 2002, November 2002.